

CentraSite User's Guide

Version 10.5

October 2019

This document applies to CentraSite 10.5 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2005-2021 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <https://softwareag.com/licenses/>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <https://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <https://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

Document ID: CS-UG-105-20210330

Table of Contents

About this Guide	7
Document Conventions.....	8
Online Information and Support.....	8
Data Protection.....	9
1 Introduction	11
CentraSite's Role in Today's SOA.....	12
Use of CentraSite by Other Products.....	15
Topology and Architecture.....	16
CentraSite Editions.....	18
CentraSite Features.....	20
CentraSite Interfaces.....	29
Starting the Graphical User Interface.....	32
Logging On to CentraSite.....	32
Using the CentraSite Business User Interface.....	33
Using the CentraSite Control User Interface.....	62
2 Organization Management	77
Introduction to Organizations.....	78
Managing Organizations through CentraSite Business UI.....	84
Managing Organizations through CentraSite Control.....	94
Deleting Organizations through Command Line Interface.....	98
3 User Management	101
Introduction to Users.....	102
Managing Users through CentraSite Business UI.....	104
Managing Users through CentraSite Control.....	115
Managing Users through Command Line Interface.....	128
Selecting Users or Groups from Repository.....	136
4 Group Management	139
Introduction to Groups.....	140
Managing Groups through CentraSite Business UI.....	142
Managing Groups through CentraSite Control.....	147
Managing Groups through Command Line Interface.....	155
Selecting Users or Groups from Repository.....	158
5 Role Management	161
Introduction to Permissions and Roles.....	162
Managing Roles through CentraSite Business UI.....	169
Managing Roles through CentraSite Control.....	176

6 Type Management	181
Introduction to Types.....	182
Basic Components of Type.....	182
Classification of Types.....	200
Composite Asset Types.....	208
Managing Types through CentraSite Business UI.....	241
Managing Types through CentraSite Control.....	253
Managing Types through Command Line Interface.....	274
7 Taxonomy Management	281
Introduction to Taxonomies.....	282
Managing Taxonomies through CentraSite Business UI.....	286
Managing Taxonomies through CentraSite Control.....	292
8 Lifecycle Management	303
Introduction to Lifecycle Management.....	304
Lifecycle Model for Lifecycle Models (LCM for LCMs).....	307
Predefined Lifecycle Models.....	308
Customizing Lifecycle Models.....	308
Updating Assets that are Under Lifecycle Management.....	314
Reverting an Asset That is Under Lifecycle Management to a Previous State.....	315
Managing Lifecycle Models through CentraSite Business UI.....	316
Managing Lifecycle Models through CentraSite Control.....	335
9 Asset Management	353
Introduction to Asset Catalog.....	354
Customizing Your Asset Catalog.....	355
Executable Design/Change-Time Actions on Your Asset Catalog.....	366
Managing Assets through CentraSite Business UI.....	379
Managing Assets through CentraSite Control.....	549
Managing Assets through Command Line Interface.....	616
RAML to CentraSite REST API Mappings.....	634
Swagger to CentraSite REST API Mappings.....	641
Asset Navigator.....	648
10 Policy Management	667
Introduction to Design and Change-Time Policies.....	668
Managing Design Time Policies through CentraSite Business UI.....	681
Managing Design and Change-Time Policies through CentraSite Control.....	711
Managing Design-Time and Change-Time Policies through Command Line Interface.....	758
Predefined Policies.....	776
Built-In Design/Change-Time Actions Reference.....	782
Configuring Email Notifications.....	842
11 Report Management	853
Introduction to Reports.....	854

Predefined Reports.....	855
Configuring JDBC Support for BIRT Reports.....	867
Reserved Identifiers.....	869
Managing Reports and Report Templates through CentraSite Business UI.....	869
Managing Reports and Report Templates through CentraSite Control.....	881
Managing Reports and Report Templates through Command Line Interface.....	888
12 Portlet Management.....	909
Introduction to Portlets.....	910
Types of Portlets.....	910
Tailor Your Portlets.....	913
Adding a Portlet to Your Welcome Page.....	913
Viewing Your Portlets.....	922
Configuring a Portlet.....	922
Collapsing or Expanding Portlets.....	924
Rearranging Portlets.....	924
Removing Portlets.....	925
Built-in Design/Change-Time Portlets.....	926
Built-in Run-Time Portlets.....	937
13 Runtime Governance.....	943
Introduction to Runtime Governance.....	944
Virtual Service Asset Management.....	959
Run-Time Policy Management.....	1176
Gateway Management.....	1364
Consumer Management.....	1397
Access Token Management.....	1430
Run-Time Alias Management.....	1463
Endpoint Management.....	1472
Runtime Events and Key Performance Indicator (KPI) Metrics.....	1479
Monitoring Logs.....	1511
14 Exporting and Importing Registry Objects.....	1517
Introduction to Export and Import of Registry Objects.....	1518
Exporting and Importing Registry Objects through CentraSite Business UI.....	1527
Exporting and Importing Registry Objects through CentraSite Control.....	1531
Exporting and Importing Registry Objects through Command Line.....	1536
Best Practices: Exporting and Importing Assets.....	1540
15 Suite Usage Aspects.....	1543
Introduction to Suite Usage Aspects.....	1544
Versioning Assets.....	1544
Modifying or Deleting Assets.....	1544
Publishing Assets.....	1544
CentraSite Communication with Designer UI.....	1545
User Accounts.....	1545
UDDI Clients.....	1545
Using CentraSite with ARIS.....	1546

Using CentraSite with webMethods API Portal.....	1558
16 CentraSite and API Gateway Integration.....	1565
CentraSite and API Gateway Integration.....	1566
API Gateway Asset Mapping Details.....	1566
Virtual Service Mapping Details.....	1569
Alias Mapping Details.....	1572
Runtime Policy Mapping Details.....	1581
Consumer Application Mapping Details.....	1643
Modifications to Error Codes and Responses for Runtime Policies.....	1647

About this Guide

- Document Conventions 8
- Online Information and Support 8
- Data Protection 9

This guide describes how to use CentraSite once it is installed and configured in your environment. It describes various CentraSite features and how these features can be used.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at <http://documentation.softwareag.com>. The site requires credentials for Software AG's Product Support site Empower. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at <https://empower.softwareag.com/>.

You can find product information on the Software AG Empower Product Support website at <https://empower.softwareag.com>.

To submit feature/enhancement requests, get information about product availability, and download products, go to [Products](#).

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the [Knowledge Center](#).

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at https://empower.softwareag.com/public_directory.asp and give us a call.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at <http://techcommunity.softwareag.com>. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 Introduction

■ CentraSite's Role in Today's SOA	12
■ Use of CentraSite by Other Products	15
■ Topology and Architecture	16
■ CentraSite Editions	18
■ CentraSite Features	20
■ CentraSite Interfaces	29
■ Starting the Graphical User Interface	32
■ Logging On to CentraSite	32
■ Using the CentraSite Business User Interface	33
■ Using the CentraSite Control User Interface	62

CentraSite's Role in Today's SOA

Today's enterprises are quickly adopting Service Oriented Architecture (SOA) as a strategy for delivering business applications that can be developed and extended quickly.

SOA is an approach to building business systems in which IT organizations deploy computing capabilities as coarse-grained, reusable blocks of functionality known as *services*. Typically, a service models a single task or repeatable process within the enterprise. Business analysts, enterprise architects, and developers assemble services into higher level constructs such as business processes, composite applications, and complex services.

IT organizations usually host services on various back-end systems within (or possibly outside) the enterprise and expose them to consumers in loosely coupled fashion through an enterprise service bus (ESB) or other mediator. Deploying services into a mediation layer provides the services with location transparency and implementation independence, allowing an IT organization to interchange and evolve back-end service implementations without disrupting the consumer applications that use them.

Despite its advantages, SOA requires a development and governance infrastructure that is radically different from traditional computing applications. To design, develop, and deploy SOA-based applications, architects and administrators must consider how to:

- Manage a computing environment comprised of hundreds (or potentially thousands) of services and supporting artifacts.
- Ensure that the many computing artifacts (for example, services, schemas, business processes) supplied by autonomous development organizations meet enterprise policies and standards.
- Define processes to ensure that services and other artifacts are accepted into the SOA in a controlled and well defined way.
- Provide a development environment in which developers and business analysts are encouraged to assemble applications from existing assets in the SOA rather than building them from scratch.
- Evaluate the consequences of a proposed change on a complex array of interdependent artifacts.
- Provide a development environment that accommodates the unique processes and requirements of individual development organizations.

CentraSite is a standards-based SOA registry and repository. It serves as the central system of record for the web services and other computing assets of an organization and provides the tools and infrastructure necessary to implement and manage SOA-based applications successfully.

CentraSite supports the entire development lifecycle of an SOA-based application, from its design and implementation to its deployment and ongoing operation in the runtime environment.

Support for the SOA Design-Time Environment

CentraSite supports the development of SOA-based applications by enabling developers, architects, and business analysts to:

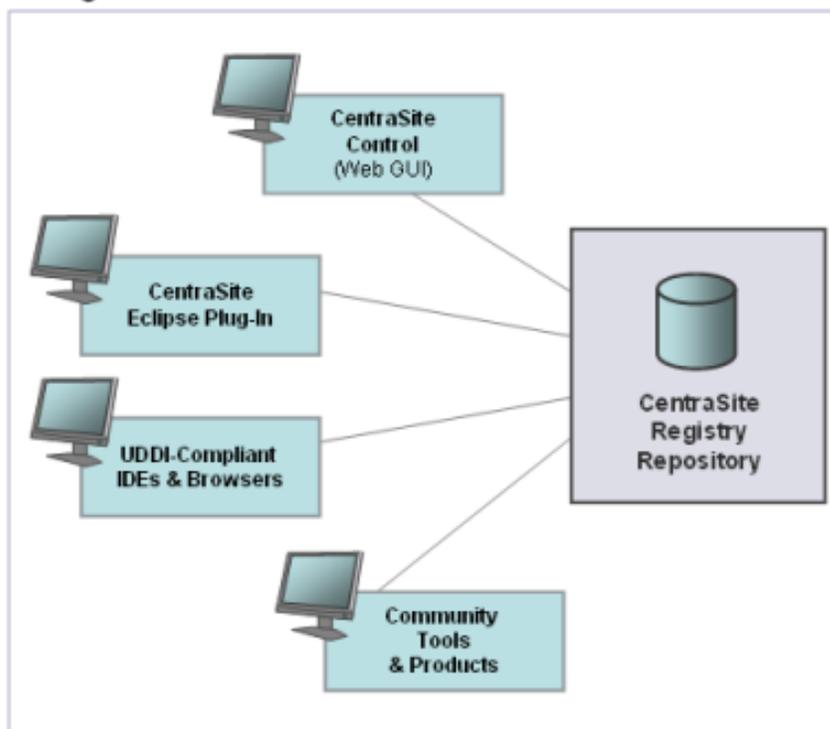
- Publish web services and other reusable assets into their organization's central registry.

- Discover web services and use them to assemble consumer applications.
- Obtain detailed information about a web service, including the list of its consumers, its technical support contacts, its disposition in the development lifecycle, usage tips, and performance data.
- Examine the relationships that a web service has with other artifacts in the SOA in order to understand how a change to that service will impact the service's sub-components and dependents.

CentraSite supports an array of design-time tools that enable developers, architects, and business analysts to discover, publish, and re-use SOA assets. These tools include:

- CentraSite Control, a browser-based user interface provided with CentraSite.
- CentraSite Business UI, a browser-based user interface provided with CentraSite.
- The CentraSite plug-in for Eclipse, also provided with CentraSite.
- UDDI V3.0-compatible registry browsers and IDEs.
- Third-party design-time tools available from members of the CentraSite Community. The CentraSite Community is a group of independent software vendors and system integrators who develop products that integrate with CentraSite.

Design-Time Tools Available for CentraSite



SOA Governance and API Management

Managing the content of the registry is critical to the success of an SOA environment. To support this effort, CentraSite provides governance capabilities and tools that enable administrators and architects to:

- Control access to CentraSite and to the metadata for individual assets listed in the registry.
- Model the specific entities that make up an organization's SOA environment as well as the dependencies and interrelationships of those entities.
- Enable reuse of computing assets by providing easy access to in-depth information about an artifact's technical properties, semantics, and relationships to other artifacts in the SOA.
- Define classification systems (taxonomies) that enable web services and other assets to be easily discovered and managed.
- Impose mandatory testing, approval processes, and quality checks to ensure that assets accepted into the SOA adhere to organizational standards and policies.
- Model the lifecycle process associated with each asset type and specify the events that are to be triggered when an asset transitions from one lifecycle state to another.

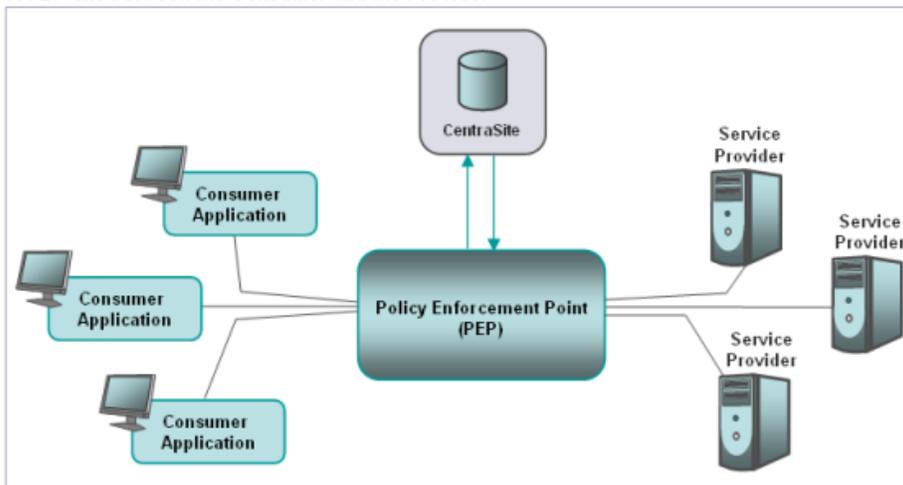
Administrators and architects use CentraSite Control and Business UI, the browser-based user interfaces provided with CentraSite, to perform these types of governance-related tasks. Systems and tools from the CentraSite Community can also provide this kind of functionality. Some basic administrative tasks can also be performed using the CentraSite Eclipse plug-in tool.

Support for the SOA Run-Time Environment

CentraSite provides tools that support the management and monitoring of services in the run-time environment. Using CentraSite, administrators can define policies that execute on policy enforcement points (PEPs) that reside between the consumer and the service endpoint. These policies typically perform security-related activities (such as authentication and message encryption or decryption), auditing or logging tasks, and performance reporting functions.

When webMethods Mediator is used as a policy enforcement point, administrators can define and deploy virtual services into the run-time environment. Virtual services operate as consumer-facing proxies for the endpoints where Web services are actually hosted. Besides performing security, logging, and monitoring activities, a virtual service can also execute advanced mediation steps such as message routing, load-balancing, failover handling, and message transformation.

A PEP sits between the Consumer and the Provider



CentraSite supports the run-time environment by enabling administrators and analysts to:

- Define and manage standard run-time policies.
- Attach run-time policies to web services and deploy the policies to specified PEPs in the run-time environment.
- Define and deploy virtual services, to perform mediation steps such as routing, load-balancing, failover, and message transformation.
- Monitor the run-time performance of services and identify services that fail to meet specified thresholds.

Out of the box, CentraSite provides support for the following:

- webMethods Mediator, which is a PEP that provides policy enforcement, service mediation, and monitoring capabilities. webMethods Mediator enforces run-time policies that you create in CentraSite.

Use of CentraSite by Other Products

CentraSite's governance capabilities are used by many Software AG products.

- Developers who use Software AG Designer can publish service metadata to CentraSite and browse CentraSite's registry from the Designer IDE.
- Analysts who use the webMethods Business Process Management System (BPMS) can browse CentraSite for services that they can use to implement the process steps in their business processes. Analysts can also publish process models into CentraSite and use CentraSite's Asset Navigator feature to understand the model's dependencies on other components in their environment.
- Analysts who use ARIS Architect can publish their business processes into CentraSite and integrate them with processes from webMethods BPMS (and vice versa).

- Developers who use the web service stack to expose functionality in Software AG's EntireX or ApplinX products can register their web services in CentraSite for others to find and reuse.
- Developers who build applications using the Software AG Natural product can use the NaturalONE user interface to publish business services to, and retrieve business services from, CentraSite's registry. In addition, the NaturalONE Lifecycle Manager adds its own asset types and related objects to CentraSite.

Topology and Architecture

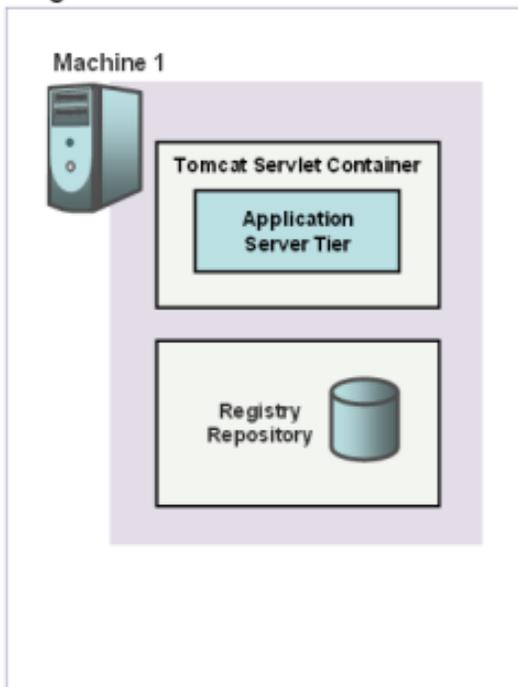
Basic Topology

CentraSite consists of two basic components:

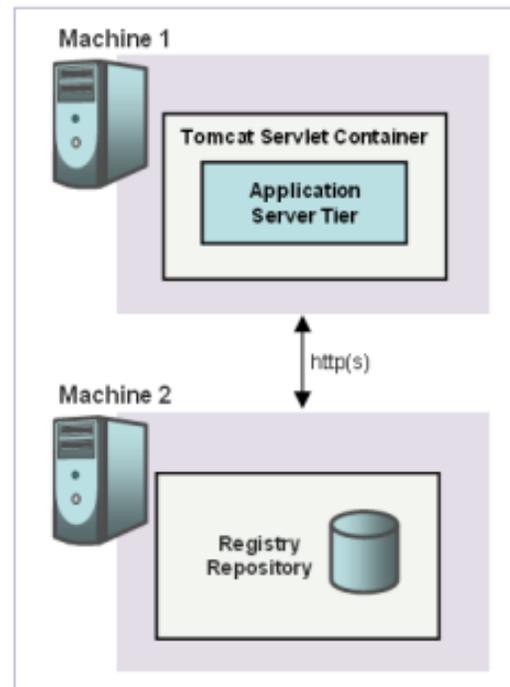
- **CentraSite Application Server Tier.** The CentraSite Application Server Tier is a web application that runs on the Software AG Runtime. This component hosts the CentraSite graphical user interfaces and also supplies the UDDI V3.0 interface to the registry.
- **CentraSite Registry Repository.** The CentraSite Registry Repository is the portion of CentraSite that hosts the registry and the repository.

Typically, both components reside on the same machine. However, if conditions at your site require it, you can install these components on separate machines.

Single Node Installation

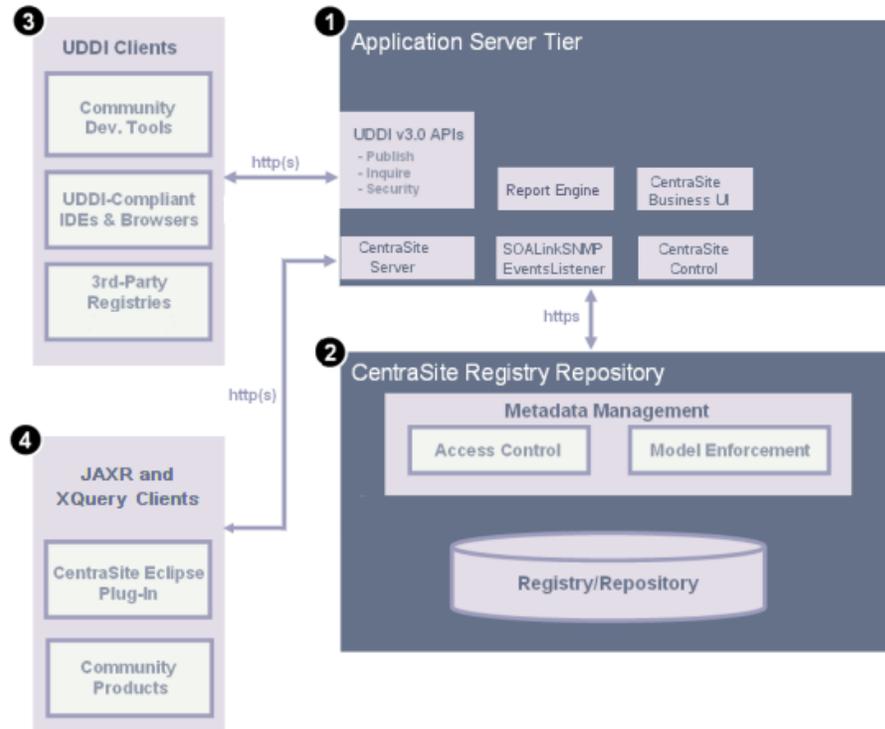


Dual Node Installation



Architecture

The following diagram describes the major sub-systems on each of the two major CentraSite components and describes the types of clients that they serve:



#	Description
---	-------------

1	The CentraSite Application Server Tier (CAST) hosts CentraSite components that are provided as web application. The components of the CentraSite Application Server Tier function as clients of the CentraSite Registry Repository component.
---	---

CentraSite Control is CentraSite's browser-based user interface, providing general purpose features for administrators, designers, and end users.

CentraSite Business UI is a lightweight alternative to CentraSite Control, offering high-level features for business users.

The CentraSite Server authenticates client applications to the CentraSite Registry Repository.

The CentraSite Application Server Tier hosts the UDDI services that client programs use to interact with CentraSite using the UDDI V3 API.

The CentraSite Control, CentraSite Business UI, and UDDI services interact with the Registry and Repository component using the CentraSite API for JAXR (registry) and HTTPS (repository).

#	Description
2	<p>The CentraSite Registry Repository component manages the content of the registry and the repository. Besides housing the data that makes up the registry and repository, this component controls access to CentraSite and ensures that the data objects in the registry conform to the CentraSite information model.</p> <p>Client programs interact with the registry portion of this component using the CentraSite API for JAXR and the CentraSite API for XQuery. Client programs interact with the repository portion of this component using HTTP.</p> <p>Client programs do not interact directly with the CentraSite Registry Repository. Instead, the communication from the client is always routed through the CentraSite Server component of the Application Server Tier.</p>
3	<p>UDDI clients interact with CentraSite using the UDDI V3 services that reside on the Application Server Tier. UDDI clients include developer tools supplied by Community partners, third-party UDDI browsers, UDDI-compliant IDEs and UDDI registries that are integrated with CentraSite.</p>
4	<p>XQuery and JAXR-based clients interact with the CentraSite Server component of the Application Server Tier. These clients include the CentraSite plug-in for Eclipse, third-party applications developed by CentraSite Community and (potentially) applications developed by your own organization.</p> <p>Authentication of the client communication is performed by CentraSite Server. Therefore, the clients do not send their requests directly to the Registry Repository but instead to the CentraSite Server. The CentraSite Server forwards authenticated client requests to the Registry Repository.</p>

CentraSite Editions

In addition to the standard, full-feature CentraSite edition, Software AG provides the CentraSite Community Edition. The Community Edition is a free-of-charge version of CentraSite that is available for download from <http://www.centrasite.com/>. This edition provides basic registry functionality and supports the installed set of asset types. The Community Edition enables you to explore CentraSite's basic capabilities before advancing to the full-feature edition.

The following table describes the features that are available in each CentraSite edition:

Feature (available in full-feature CentraSite edition) Availability in Community Edition

Registry	✓
Repository	✓
Catalog Features	✓

Feature (available in full-feature CentraSite edition) Availability in Community Edition

- Browse Assets	✓
- Create Assets	✓
- Keyword Search	✓
- Advanced Search	✓
- Version Asset	✓
- Impact Analysis	✓
Supporting Document Library	✓
My Favorites	✓
Notifications	✓
Design/Change-Time Policies	
Approvals	
Interoperability with Policy Enforcement Points	
Run-Time Policies	
Virtual Services	
Web Service Quality-of-Service Monitoring	
Lifecycle Management	
Federation	
Logging	
Organization Management	✓
Users, Groups, Roles Management	✓
Taxonomy Management (Custom Taxonomies)	✓
Asset Type Management (Custom Asset Types)	
Reporting	✓

Feature (available in full-feature CentraSite Availability in Community Edition edition)

Eclipse-based GUI	✓
Browser-based GUI (CentraSite Control & CentraSite Business UI)	✓
XQuery and JAXR-based APIs	✓
UDDI API	✓

CentraSite Features

The Registry

The *registry* refers to the part of CentraSite that manages the set of objects that represent the artifacts in your SOA environment (for example, web services, XML schemas, BPEL processes). The registry also contains supporting objects such as Organizations, Users, Policies, and Taxonomies that CentraSite itself uses to manage and organize the SOA artifacts that are contained in the registry.

It is important to understand that the registry functions as a directory, not a library. That is, it describes the properties of an artifact (for example, name, description, location, contact information, technical specifications, lifecycle disposition), but it does not hold the artifact itself. For example, the registry entry for an XML schema contains information about the XML schema. However, the schema itself resides in a different data store known as the *repository*.

The CentraSite registry supports the Java API for XML Registries (JAXR). JAXR is a standard API for working with XML registries. The JAXR information model provides a standard way to describe registry content. Its API includes powerful capabilities for describing, classifying, relating, and querying the content of a registry.

The Catalog

In CentraSite, the term *catalog* refers collectively to the sub-set of the objects in the registry that are *assets*. Generally speaking, an asset is an object that represents an artifact in your SOA environment, such as a web service, an XML schema, or a BPEL process.

When initially installed, the CentraSite catalog supports the following types of assets:

- Services
- Virtual Services
- REST Services
- Virtual REST Services
- OData Services

- Virtual OData Services
- XML Schemas
- BPEL Processes
- Application
- Application Servers

It also includes support for the types of assets that are published and consumed by products in the webMethods product suite (for example, assets such as CAF Task Types, TN Document Types, and so forth).

However, CentraSite's catalog is completely extensible and can be configured to hold any type of artifact that your organization cares to model. For example, you might want to customize your catalog to include metadata for artifacts such as Java libraries, portlets, and XSLT documents.

Any custom object type that you add to CentraSite is, by default, treated as an asset that is part of the catalog.

Note:

The terms *object* and *asset* have very specific meanings within CentraSite. An *object* is any data object that CentraSite maintains in its registry. An *asset* is an object that is treated as a member of the catalog. Therefore, all assets are objects, but not all objects are assets.

The Repository

The repository is the data store in which CentraSite maintains documents and other file-like resources. For example, when you publish an XML schema to CentraSite, CentraSite generates an entry in the registry that describes the schema and then stores the schema document itself in the repository (the registry entry includes a link to this document). By providing both registry and repository capabilities, CentraSite enables you to centrally manage the metadata for an asset as well as the asset itself.

Note:

When an asset in the catalog represents an entry in the repository, the item in the repository is often referred to as the *asset file*.

CentraSite also uses the repository as the data store for items that it produces and consumes, such as report templates, policy descriptions, and lifecycle models.

The JAXR Information Model

The CentraSite registry supports the JAXR information model. Physically, the registry resides in a single XML database that can be accessed using XQuery, UDDI, or JAXR-based client APIs.

A JAXR-based registry is built around a generic, but extensible object called a RegistryObject. The RegistryObject class defines a minimal set of metadata and provides methods that enable an object to be classified and associated with other objects in the registry.

Although the metadata that RegistryObject specifies is very minimal (Name, Description, and Key), it can be dynamically extended to incorporate additional metadata. By extending the RegistryObject, one can model and catalog virtually any type of artifact in an SOA environment.

Objects in the Information Model

In general, CentraSite's information model consists of *system-related objects* that represent the artifacts in your SOA and support the administration and management of the registry.

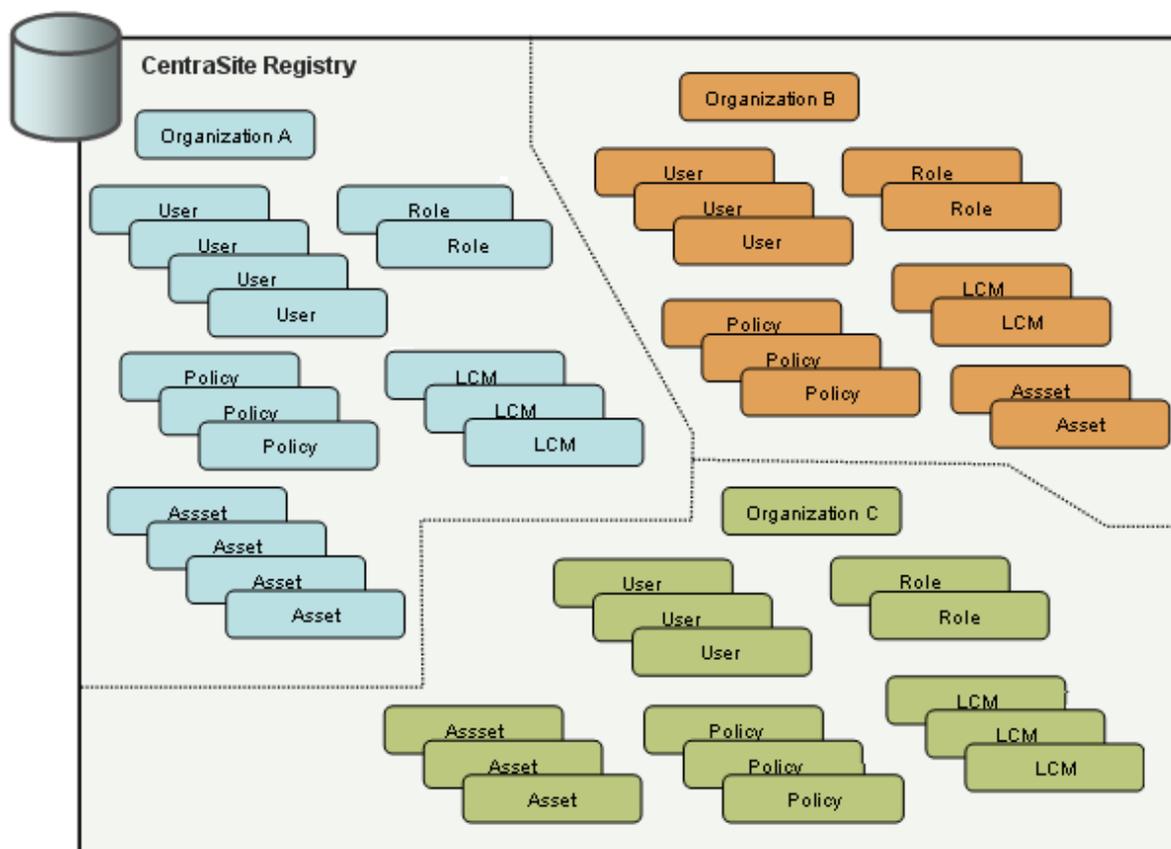
- System-related objects include objects such as organizations, users, groups, roles, taxonomies, policies, and lifecycle models. These type of objects do not appear in the catalog, however, they play a key role in managing its content.

The organization object in particular plays a major role within the registry. Under the registry's information model, any object that is not an organization must be associated with an organization object.

CentraSite is installed with one predefined organization. When the administrator of this organization creates users (which are represented by user objects in the registry), CentraSite automatically associates those user objects with the administrator's organization. Similarly, when those users subsequently create objects in the registry, CentraSite associates those objects with the user's organization.

When users work with CentraSite, they see only the registry objects that belong to their organization. Because organizations restrict users to the portion of the registry that belongs to their organization, they provide a way to, in effect, partition CentraSite into multiple, logical registries.

Organizations enable you to partition the registry



Note:

When necessary, it is possible to share objects between organizations. Using permissions, one can give a group of users in another organization permission to access a specific data object. Additionally, there are certain types of objects (such as Policies) that can be made globally available to all organizations. In both cases, however, the organization in which the objects were originally created maintains ownership of the shared objects.

- **Assets** refer to registry objects that represent the artifacts in your SOA. As installed, the CentraSite registry supports a set of asset types (Services, Virtual Services, REST Services, Virtual REST Services, OData Services, Virtual OData Services, XML schemas, BPEL processes, and Application Servers). Because these asset types are based on the JAXR extensible RegistryObject, you can customize the amount and type of metadata that CentraSite maintains for each type. You can also create additional asset types as necessary.

Relationships Between Objects in the Registry

In JAXR, relationships are modeled using Association objects. Internally, CentraSite uses these objects extensively to establish relationships between many system-related objects in the registry. For example, Associations are used to relate an organization to its set of registry objects. CentraSite automatically generates and maintains these types of underlying associations when you create or modify system-related objects using CentraSite GUIs or APIs.

Besides the implicit relationships that CentraSite maintains for system-related objects, CentraSite also allows you to explicitly establish relationships between registry objects through the use of *relationship attributes*. A relationship attribute is part of an object's metadata.

An object can have many relationship attributes reflecting the many types of relationships it has with other objects. For example, a Service asset might include a relationship attribute called `Uses`, which relates the service to its constituent artifacts (that is, assets that it uses or otherwise depends upon). The same asset might also include a relationship attribute called `Used By` which relates the service to its dependents.

Relationship attributes can be *predefined* or *ad hoc*. Predefined relationship attributes are ones that are part of an asset's type definition. When a relationship attribute is predefined, the attribute is present in all assets of that type. Ad hoc relationships are ones you can create as necessary for an individual instance of an asset.

Organizations

An organization is a high-level container for the assets and other objects that make up a CentraSite registry. Any object that is not an organization must belong to an organization. You use organizations to partition your registry into autonomous collections of assets that you can administer independently. You can use organizations to arrange your registry around functional lines of business, regional subsidiaries, branches, legal entities, departments, and B2B partners (for example, suppliers, and customers.) You can define parent-child associations between organizations to model the hierarchical structure of entities in your enterprise.

Users, Groups, Roles, and Permissions

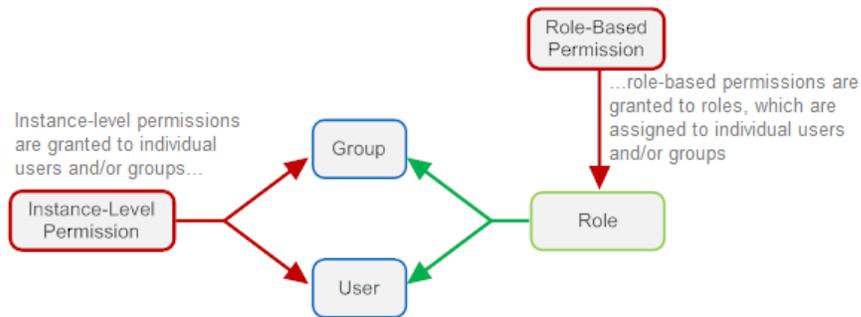
Users are individuals that are known to CentraSite. You assign roles and permissions to users to specify which operations they can perform and which registry objects they can access. You can create groups of users and assign roles and permissions to those groups.

Permissions determine the operations users can perform and the set of objects users can access.

- *Instance-level permissions* grant users or groups access to a specific instance of an object in the registry, a specific folder in the repository, or a specific file in the repository. For example, you would use instance-level permissions to give a user or group the ability to modify a specific asset in the catalog.
- *Role-based permissions* grant roles access to an entire class of objects or give the ability to perform certain operations. For example, you would use role-based permissions to give the role the ability to edit all of an organization's assets.

The following diagram illustrates the relationships between instance-level permissions, role-based permissions, users, groups, and roles:

Assignment of instance-level vs role-based permissions



Asset Types

CentraSite's flexible and extensible registry structure enables you to model any kind of asset that you might want to include in your asset catalog. It supports a rich set of attribute types for defining the different properties and qualities of your assets. These types include attributes that you can use to classify an asset according to a predefined or custom taxonomy and attributes that you can use to associate the asset with other objects in the registry.

Taxonomies

A taxonomy is a hierarchical classification scheme. In CentraSite, you use taxonomies to classify objects in the registry. Taxonomies enable you to filter, group, and sort the contents of the registry.

Lifecycle Models

CentraSite enables you to associate lifecycle models with assets and certain other object types in the registry. A lifecycle model defines a set of states that make up the lifecycle of a particular object type. For example, the lifecycle of a web service in your organization might consist of the Development, Test, Production, Revision, and Retirement states. In addition to defining the states that a particular object type can assume, a lifecycle model specifies who is permitted to transition an object from one state to another. You can also define policies that will execute when specified transitions occur.

CentraSite's lifecycle management feature provides added visibility into your SOA environment by enabling you to capture and report the disposition of the assets in the SOA. Moreover, it provides you with a single point of control from which you can centrally manage the lifecycle process of your computing assets.

The lifecycle-management user interface is not available in CentraSite Community Edition.

Design/Change-Time Policies

Design/change-time policies enable you to define and enforce organizational standards within your registry.

A design/change-time policy defines a series of actions that you associate with a registry event such as the addition, deletion, or modification of an object. When the specified event occurs, CentraSite executes the actions prescribed in the policy.

Among other things, you can use design/change-time policies to:

- Initiate review and approval processes at specified points during the lifecycle of a registry object.
- Validate metadata that users submit to the registry to ensure that it conforms to organizational standards and conventions.
- Perform automated testing and quality checks.
- Issue notifications to specified groups or individuals.
- Trigger updates or other types of procedures on external systems.

For example, you might define a policy that performs a series of automated tests when a provider submits a web service to your catalog. The policy would accept the service into the catalog only if the series of tests execute successfully.

Note:

Design/change-time policies are not available in CentraSite Community Edition.

Run-Time Policies

Run-time policies define actions that are to be carried out by a policy-enforcement point (PEP) when a consumer requests a particular service through the PEP. The actions in a run-time policy perform activities such as identifying/authenticating consumers, validating digital signatures, logging run-time events, and capturing performance measurements.

You use CentraSite Business UI to define run-time policies, associate them with services, and deploy them on specified PEPs, say Mediator gateways in the run-time environment. You also use CentraSite Control to monitor quality-of-service and other performance metrics for the services to which you have attached run-time policies.

Note:

Run-time policies are not available in CentraSite Community Edition.

Virtual Services

A virtual service functions as public-facing proxy for a web service, REST service, or OData service endpoint. You deploy virtual services on a specific type of policy enforcement point called the webMethods Mediator. Consumers who wish to use a particular web service, for instance, submit their requests to the virtual service on the webMethods Mediator, not to the endpoint where the service is actually hosted. The virtual service receives requests from consumers and routes them to the appropriate service endpoint.

You define and deploy virtual services using CentraSite Business UI. You can attach a policy to a virtual service just as you would a regular web service. Additionally, you can include processing steps in a virtual service to perform activities such as content-based or context-based message routing, load-balancing, failover handling, and message transformation.

Note:

Virtual services are not available in CentraSite Community Edition.

Versions

CentraSite's versioning capabilities enables you to maintain multiple versions of the same object.

Versioning an object creates a clone of the object. When you version an object, CentraSite copies the source object with all of its attributes and then increments the version number in the cloned target object. Additionally, CentraSite establishes a supersedes relationship between the new version of the object and the old one. Among other things, this relationship enables you to view and manage all versions of an asset.

Consumers of Virtual Services

In CentraSite there are three concepts of consumers.

- The first refers to developers who discover assets in the catalog that they want to reuse. Such developers can register to become *registered consumers* of those assets. You might give registered consumers access to more of the asset's metadata (that is, enable them to view additional profiles) and/or develop processes that notify them when changes occur to an asset that they consume.
- The second concept refers specifically to any arbitrary asset that consumes any other asset in the CentraSite registry. This specific type of consumer is for the design-time usage.
- The third concept refers specifically to a computer application that consumes (invokes) virtual services at run time. This specific type of consumer is represented in the registry by instances of the *Application* asset type. Application assets are used by webMethods Mediator to determine from which computer application a request for a virtual service originated.

A consumer application is represented in CentraSite by an *application asset*. An application asset is an instance of the Application asset type, which is one of the predefined types installed with CentraSite. An application asset defines the precise characteristics by which Mediator can identify messages from a specific consumer application at run time.

Reporting

CentraSite provides reporting capabilities based on the Business Intelligence Reporting Tools (BIRT) open source reporting system. A standard set of reports is installed with CentraSite. You can define additional reports using the BIRT report designer in Eclipse.

Using the reporting features in CentraSite, you can obtain reports about any object type (or multiple types) in the registry. For example, you might want to create a report that list of all the inactive users in your organization. You might also want a report that provides the change history for a specified service in the catalog. Reports can also include information about files and documents in the repository.

Gateways

To use an instance of CentraSite with webMethods Mediator, API Portal, or Insight you must define a Mediator gateway, an API Portal gateway, or an Insight Server gateway that identifies the specific instance of Mediator, API Portal, Insight or other policy enforcement point that you want to use. The gateway instance specifies the address of the deployment endpoint, which is the endpoint that CentraSite uses to interact with Mediator or API Portal to deploy virtual services or virtualized APIs.

Impact Analysis

Note:

The *Impact Analysis* feature that was used to easily navigate and visualize the associations between the catalog assets and registry objects, and hence identify the impact when updating or deleting assets in the catalog is deprecated and will be removed in future releases. Instead, you can use the Asset Navigator interface of CentraSite Business UI that helps you to easily navigate and visualize the dependencies between assets for various use cases.

The *Asset Navigator* provides a graphical representation of relationships between assets based on most common use cases. Several use cases for navigation are provided out of the box to allow for analysis of the runtime deployment landscape, organization structure, or dependencies to or from particular assets. The asset navigator can be extended through saved searches. This feature helps you to:

- Understand dependencies between assets
- Visualize your runtime landscape and deployment information
- Understand your versioning and consumption history of services
- Get an organization chart of your organization structure in CentraSite

You can view the information based on the use case requirements. For example, you can visualize the information based on the Global use cases or the Asset Specific use cases. Global use cases includes use cases such as Organization Structure and Runtime Landscape view. The Asset Specific use cases includes use cases such as Asset Dependency, Asset Usage, and so on.

Events and Metrics

webMethods Mediator collects performance data (for example, average response time, total request count, fault count) for the virtual services that it hosts. It publishes this data to CentraSite at regular intervals. When you install and configure the Mediator, you must specify whether you want it to collect performance data and, if so, how often you want it to publish the data to CentraSite.

In addition to performance metrics, webMethods Mediator can also log *event data*. Event data supplies information about activities or conditions that occur on Mediator.

Mediator logs two basic kinds of events: data relating to the operation of Mediator itself and data relating to the execution of virtual services.

Security and Auditing

Access to CentraSite is restricted to authorized users who are authenticated through an external directory system such as an Active Directory Server (ADS). Access to objects in the CentraSite registry is controlled by both coarse-grained permissions (through the use of roles) and fine-grain view, edit, or delete permissions on individual object instances. CentraSite also maintains a complete audit trail of the operations that users perform on the individual objects in the registry.

Role-Based Access

Role-based access provides coarse-grained access control to features and objects in CentraSite. A role is a set of system-level permissions that you associate with a user account or a group of users. The permissions within a role determine which types of objects (for example, Organizations, Policies, Lifecycle Models) users in that role can create. They also specify whether a user is allowed to perform certain restricted actions (for example, View System Audit Log).

The role(s) to which a user belongs determines which screens and controls that user receives in the CentraSite user interface. With respect to API access (based on JAXR or UDDI), the role(s) associated with the client program's user account determine which methods or operations the program is allowed to perform.

CentraSite is installed with a number of predefined roles. For example, users that belong to the Policy Admin role are permitted to create and manage design/change-time policies. However, you can also create custom roles if you require a specific combination of permissions that is not supplied by the predefined roles.

CentraSite Interfaces

CentraSite has the following interfaces:

- Graphical User Interfaces (GUIs)
- Application Programming Interfaces (APIs)

To use the CentraSite Graphical User interfaces (GUIs) or APIs, you must have a user account on the instance of CentraSite to be used. Properties associated with your user account determine the CentraSite features and the set of registry objects that you can use.

During installation, CentraSite automatically creates a user account for the user who performs the installation. This user account is assigned to the CentraSite Administrator role that has super user permissions. If you are this user, you can log on to CentraSite using your regular operating system or domain credentials. If you are not this user, contact the administrator of your CentraSite installation to have a user account created for you.

Graphical User Interfaces to CentraSite

You can use the following graphical user interfaces (GUIs) to interact with CentraSite:

CentraSite Business UI

CentraSite Business UI is a browser-based interface that offers a business-level view of the CentraSite registry. Business users use this interface to browse the asset catalog, publish assets to SOA, and generate reports.

You can start CentraSite Business UI from the Windows Start menu on the machine where Software AG Runtime is installed, or you can open a browser and enter this URL:

```
http://Software AG Runtime_server:Software AG Runtime_port/BusinessUI
```

The default user ID and password are `Administrator` and `manage`, respectively. `Administrator` has the CentraSite Administrator role.

You can access CentraSite Business UI as follows:

- Log on as the internal user `Administrator`. Provide the default user ID `Administrator` and password `manage` in the logon screen and then click **Log In**.
- Log on as a user who has been registered as a CentraSite user. Provide your user ID and password in the logon screen and then click **Log In**.
- Browse CentraSite Business UI as a guest user who can only see assets whose view permissions include the Everyone group. On the logon screen, click **Access as Guest** without supplying a user name or password.

To log off CentraSite Business UI, click the **Log Out** link at the top of the screen. If you do not explicitly log out, your session automatically times out after 30 minutes of inactivity. Software AG recommends explicitly logging off to make sure the cookies from your session are cleared from your machine and not reused if you log back on to CentraSite Business UI before your earlier session's timeout period elapses.

CentraSite Control

CentraSite Control is a browser-based user interface that supplies access to all of CentraSite's features for various types of users, as follows:

Type of User	Tasks You Can Perform
Administrator	Manage user accounts, user groups, and roles.
Enterprise architect or an analyst	Manage policies, generate reports, and define new asset types.
Developer	Search the catalog for assets that you want to use and to publish assets that you create.

You can start CentraSite Control from the Windows Start menu on the machine where Software AG Runtime is installed, or you can open a browser and enter this URL:

```
http://Software AG Runtime_server:Software AG Runtime_port/PluggableUI
```

The default user ID and password are `Administrator` and `manage`, respectively. `Administrator` has the `CentraSite Administrator` role.

You can access `CentraSite Control` as follows:

- Log on as the internal user `Administrator`. Provide the default user ID `Administrator` and password `manage` in the logon screen and then click **Log On**.
- Log on as a user who has been registered as a `CentraSite` user. Provide your user ID and password in the logon screen and then click **Log On**.
- Browse `CentraSite Control` as a guest user who can only see assets whose view permissions include the `Everyone` group. On the logon screen, click **Browse as Guest** without supplying a user name or password.

To log off `CentraSite Control`, click the **Log Off** link at the top of the screen. If you do not explicitly log off, your session automatically times out after 60 minutes of inactivity. Software AG recommends explicitly logging off to make sure the cookies from your session are cleared from your machine and not reused if you log back on to `CentraSite Control` before your earlier session's timeout period elapses.

CentraSite Eclipse UI

The `CentraSite Eclipse` user interface is a plug-in for the `Eclipse Integrated Development Environment (IDE)`. This interface enables developers to easily discover and publish assets directly from their `Eclipse` development environment. It also permits developers to use drag-and-drop techniques to incorporate assets from the `CentraSite` registry into development projects in `Eclipse`. (Certain administrative operations can also be performed using the `Eclipse` plug-in.)

Application Program Interfaces to CentraSite

You can use the following application programming interfaces (APIs) to interact with `CentraSite`.

For information about using these APIs, see the *CentraSite Developer's Guide*.

CentraSite API for JAXR

The `CentraSite API for JAXR (Java API for XML Registries)` supports the `Java EE` interface and enables you to develop `Java` programs that interact with the `CentraSite` registry.

UDDI API

The `UDDI API` enables you to interact with `CentraSite` using the `UDDI` services (for example, the `Publish`, `Inquiry`, and `Security` services).

API for JMX

The monitoring and management functions are exposed for the `Java Management eXtensions API (JMX)`. Using this interface enables you to monitor and manage `CentraSite` through `JMX`-based management tools.

Starting the Graphical User Interface

This section describes how to access the browser-based user interface for managing the CentraSite Registry Repository.

Note:

Because CentraSite's browser-based user interface uses JavaScript, you must ensure that your web browser is set up to allow JavaScript to execute. For more information about how to verify that your web browser settings to allow JavaScript, see the help for your web browser.

Using URL to Start the Graphical User Interface

CentraSite Business UI: The following URL has been predefined to allow you to start the CentraSite Business UI directly in a browser:

`http://server:port/BusinessUI/`

This URL starts the CentraSite Business User Interface.

Important:

The use of the CentraSite Business UI is not supported if you are using a CentraSite Community Edition license.

CentraSite Control: The following URL has been predefined to allow you to start the CentraSite Control directly in a browser:

`http://server:port/PluggableUI/`

This URL starts the CentraSite Control User Interface.

In the above URLs, *server* is the machine on which the CentraSite is running, and *port* is the number of the CentraSite port. So, for example, `http://localhost:53307/BusinessUI/` would start the CentraSite Business UI on a local installation using the default CentraSite port number.

Alternative Procedures on Windows

The Windows **Start** menu item **Programs > Software AG > Tools** contain an entry for starting CentraSite.

This Start menu entry is available on the machine where the Software AG Runtime is installed.

If you have just installed CentraSite, clear your web browser's cache, otherwise JavaScript errors can occur when you start some browser-based components of CentraSite.

Logging On to CentraSite

When you log on for the first time after the product installation, you must use the default credentials - username `Administrator` and password `manage`. This logs you on as the internal user `Administrator` and this user has the CentraSite Administrator role. After you have logged on with this username

and password, you can perform all administration tasks, such as customizing the portlets, viewing your inbox, and so on.

For security reasons, it is recommended that you must change the default credentials to new credentials.

You can alternatively log on as a guest user but this user cannot perform any administration tasks.

When you log on as a registered user, you supply a username and password. CentraSite user interface validates the username and password against your machine's user repository (for example, operating system or LDAP).

When you access CentraSite user interface as a guest, you are permitted to view only the assets that have been made available for general viewing by their owners. By default, guest users cannot create, modify, or delete any asset data. However, if the guest users have been granted permissions explicitly they can then access the user interface functions appropriately.

Using the CentraSite Business User Interface

On logging on to CentraSite Business UI, you see the Welcome page.

The CentraSite Business User Interface (UI) allows you to quickly and simply access CentraSite's features geared towards occasional users and non technical user roles.

CentraSite is a registry for creating, searching, and publishing assets in a Service Oriented Architecture (SOA) environment. SOA is an approach to building business systems using reusable blocks of functionality known as services. Business analysts, enterprise architects, and developers assemble services into higher level constructs such as, business processes, composite applications and complex services.

CentraSite serves as the central catalog for the services, APIs and other computing assets of an organization, and provides the tools and infrastructure necessary to implement and manage SOA-based applications, from their design and implementation to their deployment and ongoing operation in the runtime environment.

The CentraSite Business UI offers business-level features such as:

- Search or browse for assets in CentraSite.
- Create assets in CentraSite.
- Export assets from one CentraSite registry to another.
- Monitor the lifecycle state of an asset.
- Create reports about asset usage.
- Navigate to frequently-used functions from your Welcome page.
- Receive automatic notifications about changed assets.
- View or respond to your most important notifications from the Inbox.

Logging on to CentraSite Business UI as Registered User

> To log on to CentraSite Business UI as a registered user

1. Open a web browser and navigate to CentraSite Business UI.
2. In the Login screen, type the user name for your CentraSite account in the **Username** field.
3. Type the password for your CentraSite account in the **Password** field.

You may opt to select the **Remember me** check box to store the specified login credentials as a cookie in your computer.

This cookie contains information that Software AG Runtime can use to authenticate your user credentials automatically the next time you visit the CentraSite Business UI.

The display of the **Remember me** check box in the Login screen is controlled by the `RememberMe visibility` property in the `centrasite.xml` configuration file.

```
<RememberMe visibility="true"/>
```

4. Click **Log In**.

Logging on to CentraSite Business UI as Guest User

> To log on to CentraSite Business UI as a Guest user

1. Open a web browser and navigate to CentraSite Business UI.
2. In the Login screen, click the **Access as Guest** link.

Access the CentraSite Business UI without providing a username or password.

Taking a Tour

CentraSite helps govern SOA and manage APIs. It governs the lifecycle of services, APIs, and related metadata such as policies. This allows an organization to offer a robust API for developers and partners. It also helps increase of re-use software assets and improve their alignment with business needs. Before you use CentraSite, you can take a tour to learn more about the features of CentraSite.

> To take a tour

1. Open a web browser and navigate to the CentraSite Business UI.

There is no need to logon or register at this point. The tour can be taken by both the registered users and guest users.

2. Click the **Take a Tour** button.

This opens the URL configured in `centrasite.xml` file in a web browser.

Creating a New Account in CentraSite

If you do not have an existing CentraSite account and wish to open one, use the **Request an Account** link to create a new account.

➤ To create an account

1. Open a web browser and navigate to CentraSite Business UI.
2. In the Login screen, click the **Request an Account** link.
3. In the **Create an Account** screen, provide the following information:
 - a. **First Name:** Type a name that contains letters, numbers, or a combination of both. You can also use special characters: . (dot), _ (underscore), and @ (at sign). Other special characters and spaces are not allowed. The user name is case sensitive.
 - b. **Last Name:** Type a name that contains letters, numbers, or a combination of both. You can also use the following special characters: . (dot), _ (underscore), and @ (at sign). Other special characters and spaces are not allowed. The user name is case sensitive.
 - c. **Password:** Type a password that contains letters, numbers, special characters, or a combination. Spaces are not allowed. The password is case sensitive. Retype your password to confirm.
 - d. **Email:** Type a valid email address that you can access. All emails from the system are sent to this address. The email address is not made public and is only used if you wish to receive a new password or wish to receive certain notifications by email.

Note:

You cannot specify an email address that is already associated with a CentraSite account.

- e. **Organization:** Type the name of the organization.
- f. **Reason for Request:** Provide a reason as to why you need this account in CentraSite.
- g. **Show my password:** Select this check box to display the password in plain text as you type in the **Password** text box.

- h. Do one of the following:
 - Click **Register**.
 - Click the **Back to Login page** link to return to the Login screen.
4. CentraSite Business UI displays a message informing that your request for the new account is submitted to a user in the Organization Administrator role.

CentraSite internally executes an user registration workflow and submits the request for new account to the administrator or a designated group of approvers.

Important:

This workflow helps you to create a new account in an organization of interest within the CentraSite registry or repository.

CentraSite does not execute the user's request for new account operation until it obtains the necessary approvals. If an approver rejects the request, CentraSite sends a notification at the specified email address and immediately exits the workflow. On the other hand, if all the designated approvers accept the request, CentraSite sends an email notification with your account details.

5. You can log on and use CentraSite Business UI.

Logging off from CentraSite Business UI

To log off from the CentraSite Business UI, click the **Log Out** link at the top right hand corner of the screen.

If you do not log off after using CentraSite (that is, if you simply close the browser window without logging out), by default, your session automatically times out after 30 minutes of inactivity. You can set user sessions to automatically time out after a period of inactivity. You can define the length of the period of inactivity that results in an automatic logout.

If configured Single-Sign-On (SSO) authentication, you specify a URL to redirect to when you log off. When you specify a logout URL, and when you click **Log Out** or your session expires, you are redirected to that page. If you don't specify a logout URL, you are redirected to the general CentraSite login page.

You can specify any customized page to open when the logout event occurs. This is controlled by the following property statement in the `centrasite.xml` configuration file:

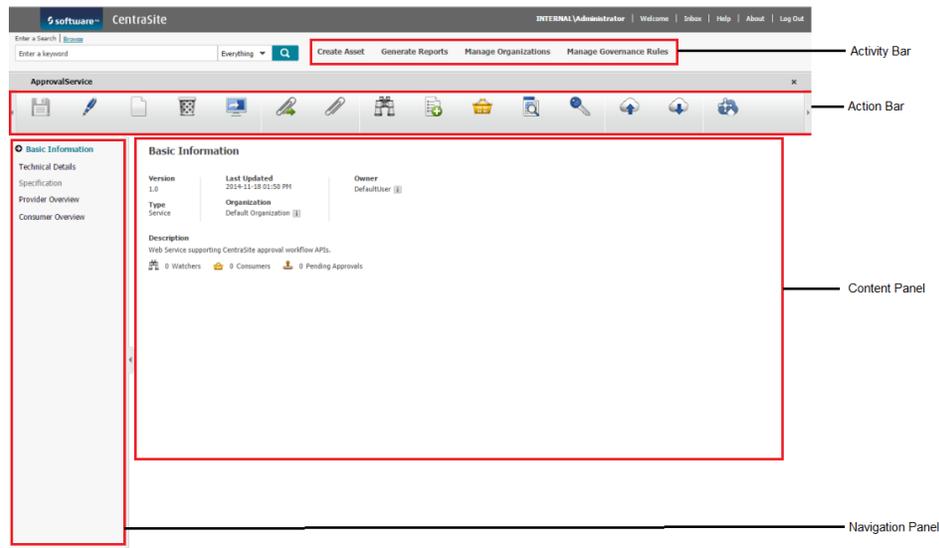
```
<SSOLogoffLandingURL>http://www.softwareag.com</SSOLogoffLandingURL>
```

Note:

Software AG recommends logging off after using the user interface. Logging off ensures that the cookies from your session are cleared from your machine. If you close your browser without logging out, these cookies might not be cleared (depending on which browser you use) and could be reused if you were to log on to CentraSite Business UI before your earlier session's timeout period had elapsed.

Navigation Controls in CentraSite Business UI

The main navigation controls in CentraSite Business UI are:



Following is an example of the asset details page.

■ Activity bar

You use the activity bar to perform various registry functions in CentraSite. The activities appearing in this space depends upon your **centrasite.xml** configuration.

■ Action bar

You use the icons on the action bar to select the task that you want to perform on the individual registry object. The actions appearing in this space depends on the role(s) assigned to user account.

■ Navigation panel

The navigation panel in the details page of an object displays the profiles that are defined in the object's type definition.

■ Content Panel

This is the area in which CentraSite Business UI displays the detailed information of each individual profile in the object. The actual content that you will see for an object depends on the role-based and instance-based permissions assigned to your user account.

The CentraSite Welcome page is your entry point to the CentraSite Business UI.

The Welcome page is configurable and you can set up views of your most frequently used functions and search queries, such as, My Favorite Assets and Recently changed assets.

From the Welcome page you can:

- Perform the CentraSite business functions from the Activities menu.

- Search or Browse for assets in CentraSite registry.
- Navigate to frequently-used Saved Searches.
- Set User Preferences.
- View or respond to your most important notifications from the Inbox.
- Access Help Center for information on the CentraSite business functions.

Access CentraSite Business Functions

Use the **Activities** menu to access the core business functions of CentraSite - **Create Asset, Reports, Organizations, Governance Rules, Taxonomies, Asset Types, and Asset Navigator.**

Search or Browse for Assets

Type a search string or browse through to find assets, taxonomies, lifecycle models, and attributes that are currently defined in the CentraSite registry.

Navigate to Frequently-Used Functions

Customize the Welcome page through Portlets to view the frequently-used functions (for example, My Favorite Assets, My Saved Searches, and so on). CentraSite Business UI organizes these functions as widgets and renders them based on your preferences.

Set Preferences

At the upper-right corner of the header area, click your display name to set personal options. Options include account information, the name of your organization, contact information, language, time zone, notification preference, saved searches, display settings, and so on.

Manage the Inbox

The Inbox displays items that involve your user account, for example, notification requests.

Help Center

On the Help Center, you can find the CentraSite help information you want.

Additionally, click on a topic of interest to display information on the top-level topics.

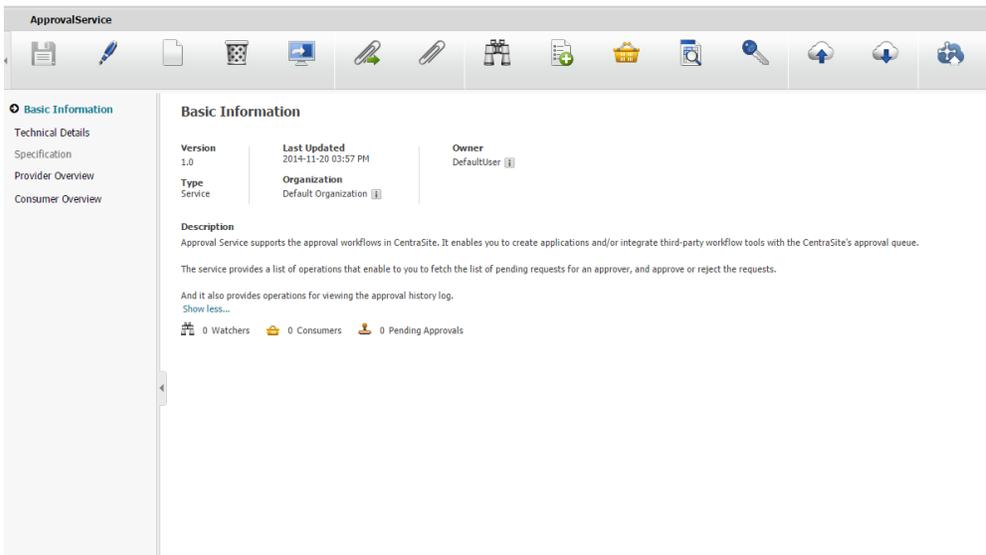
Enhanced Profile Layout For Registry Objects

CentraSite Business UI offers a new enhanced profile layout that provides easier access to the details of the individual profiles of registry objects.

Note:

The legacy profile layout is disabled from version 9.8 and later. However, you can enable this layout through the `centrasite.xml` file. You can configure CentraSite to indicate whether you want to use the enhanced or legacy profile layout.

The enhanced profile layout provides a split view representation of the profile details for an individual object and includes a navigation panel and a content panel. The layout maintains a list of all of the profile names on the navigation panel and displays the details for the selected profile on the content panel.



Keep the following information in mind when viewing the enhanced profile layout:

- CentraSite renders the **Basic Information** profile by default.
- Profile selected for rendering displays highlighted in the navigation panel with an active  icon. Also, if the profile has one or more required attributes with an empty value, then CentraSite displays an invalid  icon in edit mode.
- Profiles that have no data are displayed in gray in the navigation panel when you are in view mode. These profiles become active in edit mode.
- Attributes that only have a value are displayed in the content panel when you are in view mode. CentraSite maintains all of the attributes associated with a profile in edit mode, irrespective of their values.
- When displaying the list of profiles in the navigation panel, if the character length for any single profile name exceeds the limit set by the `navigationPane\MaxCharLength` value, then that profile name gets truncated.
- If you have a profile name which displays truncated in the navigation panel, move the cursor over the profile to view its fully qualified name as defined in the type definition.

CentraSite uses the enhanced layout of the profile details for displaying information of the following registry objects:

- Organizations
- Users
- Groups
- Roles
- Assets (instances)
- Gateways

Configuring the Enhanced Profile Layout

When you configure the enhanced profile layout, you indicate whether or not CentraSite displays the asset profile information in the enhanced or legacy profile layout and the maximum size of the navigation panel in the CentraSite configuration file.

Beginning with version 9.8, the default rendering is the enhanced profile layout. The only time you have to configure the enhanced profile layout is if the system administrator has explicitly configured for legacy profile layout.

> To configure the enhanced profile layout

1. Open the customization file, `centrasite.xml`, in a rich text editor.

You can find the **centrasite.xml** file on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\custom\conf`.

2. Locate the `AssetDetailSettings` section in the file.

```
<GUIConfiguration>
  ..
  <UIProperties>
    ...
    <AssetDetailSettings newRenderingMode="true"
      navigationPanelMaxCharLength="25" />
  </UIProperties>
</GUIConfiguration>
```

3. Set the value of the property `newRenderingMode` to `true`.

Possible values:

Value...	Indicates...
<code>true</code> (default value)	Usage of the enhanced profile layout.
<code>false</code>	Usage of the legacy profile layout.

4. Set the value of the property `navigationPanelMaxCharLength` to specify the number of characters in the profile name.

This property defines the default size of the profile name in navigation panel when displaying the asset details page in the enhanced profile layout, .

If the character length for any single profile name exceeds the number of characters you specify, CentraSite truncates the name to the specified length. The default number of characters is 25.

5. Save and close the file.
6. Restart Software AG Runtime.

Note:

If at a later stage you want to use the legacy profile layout, you must set the property `newRenderingMode` to `false`, and then restart Software AG Runtime.

User Profile Management

CentraSite allows you to view and manage your profile. In CentraSite Business UI, your profile is a collection of useful data about you. The User Preferences page in CentraSite Business UI enables you to view and manage your own profile.

By default, the User Preferences page presents the following information:

- Your contact information, such as, display name, first name and last name, photo, email address, organization, phone number, and so on.
- Your notification preference.
- Your display language and locale settings, date and time format, and the time zone in the defined locale.
- A list of your portlets.
- A list of your API keys or OAuth2 client tokens.
- A list of your saved searches, favorites, and so on.

Note:

Some settings might not take effect until you log off and then log back in to your CentraSite application.

Viewing Your User Profile

> To examine your user profile information

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your display name in the upper-right corner of the header area.

This opens the User Preferences page.

Modifying Your User Profile Details

Your system administrator determines how much of your personal information you may revise. When you place your cursor over information that is editable, a highlighted box appears.

Note:

Even if your system administrator has enabled the editing of your profile section, it is possible that some fields in that section are read-only. For example, you should generally not be able to change the value for your organization.

➤ To modify your user profile information

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your display name in the upper-right corner of the header area.

This opens the User Preferences page.
3. Expand the **Account Details** section.

This opens your user profile in Edit mode.
4. Modify your personal information, such as your display name, first name and last name, email address, and phone number as necessary.
5. Click **Save**.

Changing Your Profile Picture

In the **Account Details** section, upload, modify, or remove your profile picture.

➤ To change your profile picture

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your display name in the upper-right corner of the header area.

This opens the User Preferences page.
3. To change your profile picture in **Avatar**:
 - a. Click **Choose**.
 - b. Click **Browse** to find and select a picture.

- c. Click **OK** to complete the upload the selected picture.
4. To remove your profile picture, click **Delete**.

Note:

You do not need to click **Save** on the User Preferences page to update your changes for the profile picture.

User Preferences Management

You can control how you interact with CentraSite Business UI by specifying user preferences that you can set from the User Preferences page. The values that you specify in the User Preferences page override the default values set by your administrator.

You can manage user preferences in CentraSite using the following methods:

- CentraSite Business UI
- CentraSite Command Line Interface

Managing User Preferences through CentraSite Business UI

This section describes operations you can perform to manage preferences through CentraSite Business UI.

Setting Notification Options

The **Notification Options** allows you to enable or disable the notification options on specific events.

➤ To set your notification options

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your display name in the upper-right corner of the header area.

This opens the User Preferences page.

3. Locate the section **Notification Options**.
4. Select a notification preference in **Notification Options**.

The options include:

- **CentraSite**: Sends notifications to your **Inbox**.
- **Email**: Sends notifications to your email address.

5. Click **Save**.

Important:

When you attempt to disable both the notification options, note that CentraSite:

- Disables the **Watch** action in the asset details page.
- Disables the **Inbox** in the header.

Setting Language and Timestamp

You can configure your display settings in CentraSite Business UI. The display settings specify the language in which CentraSite Business UI displays the user interface (assuming the appropriate language pack is installed on the Software AG Runtime) and the time zone in which timestamped events are rendered when you view the activity logs and other dated information in CentraSite Business UI.

➤ To change your display settings

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your display name in the upper-right corner of the header area.

This opens the User Preferences page.

3. Locate the section **Language and Time Settings**.
4. Modify the values for the fields as necessary.

Field	Specify...
Display Language	The language in which you want CentraSite Business UI to be displayed.
Operating Language (Locale)	The language in which you want data in the CentraSite Business UI to be displayed.
Date Format	The date format to use for your current CentraSite session. If you do not specify a date format, then the date format defaults to YYYY-MM-DD.
Time Format	The time format to use for your current CentraSite session. If you do not specify a time format, then the time format defaults to HH:MM AM/PM.
Time Zone	The time zone in which you want time stamped log information rendered when it is displayed to your user account.

5. Click **Save**.

Important:

The changes to the **Display Language** and **Operating Language (Locale)** settings take effect at the next logon. All other changes you make are immediately active.

Viewing API Keys and OAuth2 Tokens

The API key or the OAuth2 token both acts as a unique identifier and secret token for authentication and has a set of access rights on the API associated with it. You can view the list of API keys and OAuth2 tokens that are available to you.

> To view a list of your API Keys and OAuth2 tokens

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your display name in the upper-right corner of the header area.

This opens the User Preferences page.

3. Locate the section **My Access Keys**.
4. Click the chevron to expand **My Access Keys**.

This displays a list of API keys and OAuth2 tokens that are available for your usage.

Renewing API Keys

The API key acts as a unique identifier and secret token for authentication and has a set of access rights on the API associated with it. You can renew the API keys that are available to you.

> To renew API keys

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your display name in the upper-right corner of the header area.

This opens the User Preferences page.

3. Locate the section **My Access Keys**.
4. Click the chevron to expand **My Access Keys**.

This displays a list of API keys and OAuth2 tokens that are available for your usage.

5. Hover over an API key.

This displays icons for one or more actions that you can perform on the API key.

6. Click **Renew** to renew the API key.

Important:

If an API key has an unlimited expiration period, then you will not see the **Renew** icon for that particular API key.

Revoking API Keys and OAuth2 Tokens

You can revoke the expired or unused API keys and OAuth2 tokens from the CentraSite registry.

> To revoke API keys and OAuth2 tokens

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your display name in the upper-right corner of the header area.

This opens the User Preferences page.

3. Locate the section **My Access Keys**.
4. Click the chevron to expand **My Access Keys**.

This displays a list of API keys and OAuth2 tokens that are available for your usage.

5. Hover over an API key or OAuth2 token you want to revoke.

This displays icons for one or more actions that you can perform on the API key.

6. Click **Delete** to revoke the API key or OAuth2 token.

A confirmation message appears that the particular API key or OAuth2 token will be revoked.

7. Click **Yes** in the confirmation dialog box.

The API key or OAuth2 token is revoked from the CentraSite registry.

Viewing Saved Searches

You can view the list of saved searches that are available to you.

> To view a list of saved searches

1. Open a web browser and navigate to CentraSite Business UI.

2. In CentraSite Business UI, click your user name that is located in the upper-right corner of header area.

This opens the User Preferences page.

3. Locate the section **Saved Searches**.
4. Click the chevron to expand **Saved Searches**.

This displays a list of saved searches that are available to you.

Adding a Search to Saved Searches List

You can create a custom user-defined search and add this search to your list of saved searches.

> To add a search to the list of saved searches

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, define a custom advanced search using the **Browse** functionality.
3. Specify a name for the user-defined search in the **Save Your Search** text box.
4. Select **Save**.

Note that you can save your search without first executing it.

Important:

If a saved search with the given name already exists in the CentraSite, you will be asked whether you wish to replace the existing search with your current search criteria.

Modifying a Search in Saved Searches

You can modify a user-defined search that is available in your list of saved searches.

> To modify a saved search

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your user name that is located in the upper-right corner of header area.

This opens the User Preferences page.

3. Locate the section **Saved Searches**.
4. Click the chevron to expand **Saved Searches**.

This displays a list of saved searches that are available to you.

5. Click the name of the saved search whose details you want to examine or modify.

CentraSite executes the saved search. The Search Results page displays the user-defined search criteria and the appropriate results.

6. Examine and modify the user-defined search criteria as necessary.
7. Click **Save**.

Renaming a Search in Saved Searches List

You can rename a user-defined search that is available in your list of saved searches.

» To rename a saved search

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your display name in the upper-right corner of the header area.

This opens the User Preferences page.

3. Locate the section **Saved Searches**.
4. Click the chevron to expand **Saved Searches**.

This displays a list of saved searches that are available to you.

5. Hover over the saved search you want to rename.

This displays icons for one or more actions that you can perform on the saved search.

6. Click **Edit** to modify the name of the saved search.
7. Type a new name for the saved search.
8. Click **Save**.

Important:

If a saved search with the given name already exists in the CentraSite, you will be asked whether you wish to replace the existing search with your current search criteria.

Removing a Search from Saved Searches List

You can remove a saved search from your list of saved searches.

> To remove a saved search

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your display name in the upper-right corner of the header area.

This opens the User Preferences page.

3. Locate the section **Saved Searches**.
4. Click the chevron to expand **Saved Searches**.

This displays a list of saved searches that are available to you.

5. Hover over the saved search you want to remove.

This displays icons for one or more actions that you can perform on the saved search.

6. Click **Delete** to remove the saved search.

A confirmation message appears that the particular saved search will be removed.

7. Click **Yes** in the confirmation dialog box.

The saved search is removed from the list of saved searches that are available to you.

Viewing My Favorites

The My Favorites feature enables you to create lists and shortcuts to items that you use routinely or otherwise want to keep close at hand. You can view the list of your favorites in CentraSite.

> To view a list of your favorites

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your display name in the upper-right corner of the header area.

This opens the User Preferences page.

3. Locate the section **My Favorites**.
4. Click the chevron to expand **My Favorites**.

This displays a list of your favorites in CentraSite.

Adding Assets to My Favorites List

Using My Favorites, you can build a collection of assets that you can display as a group. When the asset names are displayed as a group, you can click on the name of any given asset and select the actions that are appropriate for that asset.

> To add an asset to the list of favorites

1. Open a web browser and navigate to CentraSite Business UI.
2. Access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

A list of defined assets in CentraSite for which you have the View permission is displayed in the Search Results page.
3. Select the check box for a single asset, or select the check boxes for multiple assets you want to add to the **My Favorites** list.
4. On the actions bar of the Search Results page, click **Add to List**.
5. In the **Add to List** dialog box, do one of the following:
 - Select the list from the **Select an Existing List** drop-down box to add the assets to an existing list.
 - Type a name for the new list in the text box.
6. Click **Add**.

Removing Assets from the My Favorites List

You can remove assets from your list of favorites.

> To remove assets from your favorites

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your user name that is located in the upper-right corner of header area.

This opens the User Preferences page.
3. Locate the section **My Favorites**.

4. Click the chevron to expand **My Favorites**.

This displays the lists of your favorites.

5. Select the list that contains the asset you want to remove.

This displays the assets that are included in the favorites list.

6. Select one or more assets you want to remove from the favorites list.

7. Click **Remove from List**.

Renaming a Favorites List

You can rename a favorites list at any time.

> To rename a favorites list

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your display name in the upper-right corner of the header area.

This opens the User Preferences page.

3. Locate the section **My Favorites**.

4. Click the chevron to expand **My Favorites**.

This displays the lists of your favorites.

5. Hover over a favorites list.

This displays icons for one or more actions that you can perform on the favorites list.

6. Click **Edit** to rename the favorites list.

7. Type a new name for the favorites list.

8. Click **Save**.

Important:

If a favorites list with the given name already exists in the CentraSite, you will be asked whether you wish to replace the existing list with your current list.

Deleting a Favorites List

You can remove a favorites list at any time.

Note that when you delete a favorites list, any underlying assets that are referred by the list are not affected.

➤ **To delete a favorites list**

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your user name that is located in the upper-right corner of header area.

This opens the User Preferences page.

3. Locate the section **My Favorites**.
4. Click the chevron to expand **My Favorites**.

This displays the lists of your favorites.

5. Hover over a favorites list.

This displays icons for one or more actions that you can perform on the favorites list.

6. Click **Delete** to delete the favorites list.

A confirmation message appears that the particular list will be deleted.

7. Click **Yes** in the confirmation dialog box.

The favorites list is deleted from the CentraSite registry.

Viewing Portlets

➤ **To view the list of portlets**

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your user name that is located in the upper-right corner of header area.

This opens the User Preferences page.

3. Locate the section **My Portlets**.
4. Click the chevron to expand **My Portlets**.

This displays a list of portlets that are available for your usage.

Renaming a Portlet

The **My Portlets** section includes a list of portlets that represent the result set of a search query, any external HTML page or a graphical image. You can rename a portlet at any time.

> To rename a portlet

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click your display name in the upper-right corner of the header area.

This opens the User Preferences page.

3. Locate the section **My Portlets**.
4. Click the chevron to expand **My Portlets**.

This displays the lists of portlets that are available to you.

5. Hover over a portlet.

This displays icons for one or more actions that you can perform on the portlet.

6. Click **Edit** to rename the portlet.
7. Type a new name for the portlet.
8. Click **Save**.

Important:

If a portlet with the given name already exists in the CentraSite, you will be asked whether you wish to replace the existing portlet with your current portlet.

Note:

As an alternative, you can rename a portlet using its **Configure** option.

Removing a Portlet

You can remove a portlet at any time. Note that when you remove a portlet, any underlying objects that are referred by the portlet are not affected.

> To remove a portlet

1. Open a web browser and navigate to CentraSite Business UI.

2. In CentraSite Business UI, click your display name in the upper-right corner of the header area.

This opens the User Preferences page.

3. Locate the section **My Portlets**.

4. Click the chevron to expand **My Portlets**.

This displays the lists of portlets that are available to you.

5. Hover over a portlet.

This displays icons for one or more actions that you can perform on the portlet.

6. Click **Delete** to remove the portlet.

A confirmation message appears that the particular portlet will be removed.

7. Click **Yes** in the confirmation dialog box.

The portlet is removed from the CentraSite registry.

Managing User Preferences through Command Line Interface

This section describes operations you can perform to manage preferences through CentraSite Command Line Interface.

It is possible that the user preference settings, specified using a GUIConfiguration file for an user in the target registry is different from the user preference settings for the same user in the source registry. This could happen, if, for example, the user preference settings in the target registry specifies a revised GUIConfiguration file with different settings.

For example, if the user has a custom portlet that is required in the target registry but not in the source registry, such a mismatch will occur.

You can use the command line interface to reset your user preference settings in CentraSite. You can use this tool to perform the following tasks:

- Update your existing user preferences
- View your current user preferences
- Reconstruct your user preferences

Modifying User Preference Settings

Pre-requisites:

To update your user preference settings through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `set preferences` for this purpose.

➤ To modify user preference settings

- Run the command `set preferences`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set preferences -user <USER-ID> -password <PASSWORD> [-url <CENTRASITE-URL>] -targetUser <TARGET-USER> -file <CONFIG-FILE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
TARGET-USER	The user ID of a registered CentraSite user whose preference settings you want to update.
CONFIG-FILE	Name of the configuration file which contains the default user preference settings.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set preferences -user
Administrator -password manage -url http://localhost:53307/CentraSite/CentraSite
-targetUser LDAPDomain\Claire -file c:\temp\UserProfileConfig.xml
```

The response to this command could be:

```
Executing the command : set preferences
Successfully executed the command : set preferences
```

Displaying User Preference Settings

Pre-requisites:

To view your user preference settings through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `get preferences` for this purpose.

➤ To modify user preference settings

- Run the command `get preferences`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get preferences -user <USER-ID> -password <PASSWORD> [-url <CENTRASITE-URL>] -targetUser <TARGET-USER> -file <CONFIG-FILE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
TARGET-USER	The user ID of a registered CentraSite user whose preference settings you want to view.
CONFIG-FILE	Name of the configuration file which contains the user profile properties.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get preferences -user Administrator -password manage -url http://localhost:53307/CentraSite/CentraSite -targetUser LDAPDomain\Claire -file c:\temp\UserProfileConfig.xml
```

The response to this command could be:

```
Executing the command : get preferences
Successfully executed the command : get preferences
```

Resetting User Preference Settings

Pre-requisites:

To reset your user preference settings in CentraSite to default settings through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `reset preferences` for this purpose.

> To reset user preference settings

- Run the command `reset preferences`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd reset preferences -user <USER-ID> -password <PASSWORD> [-url <CENTRASITE-URL>] -targetUser <TARGET-USER>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
TARGET-USER	The user ID of a registered CentraSite user whose preference settings you want to reconstruct.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd reset preferences -user
Administrator -password manage -url http://localhost:53307/CentraSite/CentraSite
-targetUser LDAPDomain\Claire
```

The response to this command could be:

```
Executing the command : reset preferences
Successfully executed the command : reset preferences
```

Inbox Management

Your Inbox is where you receive notifications, and send and receive messages.

If you have requested to receive notifications whenever certain assets are modified, CentraSite Business UI's inbox shows the list of such notifications.

Important:

Remember that CentraSite does not display **Inbox** in the navigation bar unless you chose to receive notifications through the CentraSite's **Inbox**.

Using the Inbox, you can do the following:

- Every notification you receive appears as an item in your inbox. You may choose to receive other notifications in your inbox as well.
- Explicitly clear an entry from the list using the **Delete Notification** action.
- When you click on the **Inbox** link, the new message notification icon disappears.

Accessing Your Inbox

➤ To access your inbox

1. Open a web browser and navigate to CentraSite Business UI.

2. In CentraSite Business UI, click the **Inbox** link in the upper-right corner of the header area.

This opens the Welcome to Your Inbox page.

Creating a Notification Request

You should create a notification request if you want CentraSite to notify you when a specified asset is modified.

➤ To create notification request for assets

1. Open a web browser and navigate to CentraSite Business UI.
2. Access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

A list of defined assets in CentraSite for which you have the View permission is displayed in the Search Results page.

3. Select the check box for a single asset, or select the check boxes for multiple assets for which you want to receive notifications.
4. In the actions bar of the Search Results page, click **Watch**.

Your notification request is reflected using the attribute, **Watchers**, in the **Basic Information** profile of the selected assets. A click on the **Watchers** link displays the list of users who are currently registered to receive notifications for that particular asset.

5. Verify that the notification request has been added to your **Inbox**.

Viewing Notification Requests

If you have requested notification for one or more assets, CentraSite Business UI displays an envelope (✉) beside the **Inbox** in navigation bar indicating objects on your notification list have been updated since the last time you viewed your notifications.

➤ To view notifications you have received in Inbox

1. Open a web browser and navigate to CentraSite Business UI.
2. In CentraSite Business UI, click the **Inbox** link in the upper-right corner of the header area.

This opens the Welcome to Your Inbox page.

3. Click the asset whose notification details you want to examine.

You will see the list of users who are notified whenever this asset is updated by using the **Watchers** attribute in the **Basic Information** profile of the Asset Details page.

Deleting a Notification Request

If at any time you do not want to receive notifications for an asset, you can delete the notification request for that particular asset.

> To delete notification request for assets

1. Open a web browser and navigate to CentraSite Business UI.

2. Access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

A list of defined assets in CentraSite for which you have the View permission is displayed in the Search Results page.

3. Select the check box for a single asset, or select the check boxes for multiple assets for which you want to stop receiving notifications.

4. In the actions bar of the Search Results page, click **Unwatch**.

CentraSite automatically resets the count of the attribute, **Watchers**, in the **Basic Information** profile of the selected assets.

5. Verify that the notification request has been removed from your **Inbox**.

Removing Notification Requests from Inbox

You can also remove a notification request directly in the **Inbox**.

Note that when you remove a notification request, any underlying assets that are referred by that request are not affected.

> To remove notifications from Inbox

1. Open a web browser and navigate to CentraSite Business UI.

2. In CentraSite Business UI, click the **Inbox** link in the upper-right corner of the header area.

This opens the Welcome to Your Inbox page.

3. Select the check box for a single asset, or select the check boxes for multiple assets you want to remove from the **Inbox**.

4. On the actions bar of the **Inbox**, click **Delete Notifications**.

Using Refiners in Your Inbox

Refiners enable you to drill down into the **Inbox** based on attributes that are associated with the notifications, for example, name, description, event type, last modified date, last modified user, and so on.

Refiners are displayed in the **View** menu that is located just above the result view area.

Using the Help Center

CentraSite's **Help Center** gives an overview of the functionality of CentraSite Business UI.

Following are the various help topics in CentraSite Business UI:

Introduction

Begin with our Introduction section to know about the features that are offered by the CentraSite Business UI.

All about Assets

Chances are, you're here to check out the possible actions on an asset. For quick access to various modifications on the asset, start with this page that lists All about Assets.

Using Keyword Searching

Need some help with Using Keyword Searching? The CentraSite Business UI provides a powerful search facility. You can search for assets across organizations, classifications and types on the basis of several search criteria using ALL/ANY combinations. We'll also tell you how to define a simple search and give you plenty of points that help you define a simple search.

Browsing the Catalog

You'll undoubtedly need to view the complete list of assets at some point. You can also view a list of assets that belong to a particular asset type, organization, and user by Browsing the Catalog.

Managing the Catalog

Need some help with Managing the Catalog? CentraSite has dozens of assets that you'll want to get familiar with. We'll show you how to create or import an asset of a certain type and give you plenty of information if you want to go further on asset's definition.

Collaborating on Assets

We can provide you with all the details about approving requests for assets. The Collaborating on Assets section of the Help Center is all about approval management, so start here if you need info about the approval workflows.

Working with Notifications

You can start **Working with Notifications** feature to request CentraSite to alert you when specified assets are modified. The notification can be sent to you through an Email and/or Inbox, depending on how notification is configured in your user preferences.

Customizing your Welcome Page

The Welcome page of the CentraSite Business UI is configurable and you can set up views of your most frequently used functions and reports as *portlets*. You can walk through the contents of the Welcome page and the supported portlet types in our Customizing Your Welcome Page section.

Updating Your User Profile

Need some help with Updating Your User Profile? You can change various aspects of the CentraSite Business UI displays to suit your personal preferences.

Further Resources

For more information about CentraSite Business UI and its functions, see the Further Resources section.

Accessing CentraSite's Help Center

➤ To access the CentraSite Help Center

1. Open a web browser and navigate to the CentraSite Business UI.
2. In CentraSite Business UI, click the **Help** link that is located in the upper-right corner of the header area.

This opens the Welcome to the Help Center page.

Also, you will find the **Help Center** links on several pages of the CentraSite Business UI.

3. In the Help Center page, browse topics in the view area.
4. Click on a topic to have the information displayed.

Using the CentraSite Control User Interface

After you have logged on to CentraSite Control, the Welcome page appears. In addition to the standard navigation bar, this page offers you links to frequently used CentraSite Control features and also to external links associated with CentraSite, such as CentraSite community pages.

Logging on to CentraSite Control as Registered User

➤ **To log on to CentraSite Control as a registered user**

1. Open a web browser and navigate to the CentraSite Control.
2. In the Login screen, type the username for your CentraSite account in the **Name** field.
3. Type the password for your CentraSite account in the **Password** field.
4. Click **Log On**.

Logging on to CentraSite Control as Guest User

➤ **To log on to CentraSite Control as a Guest user**

1. Open a web browser and navigate to the CentraSite Control.
2. In the Login screen, click the **Browse as Guest** link.

Access the CentraSite Control without supplying a username or password.

Logging off from CentraSite Control

To log off from the CentraSite Control, click the **Log Off** link at the top right hand corner of the screen.

If you do not log off after using CentraSite (that is, if you simply close the browser window without logging out), your session automatically times out after 60 minutes of inactivity.

Note:

Software AG recommends you to explicitly log off when you have finished using the user interface. Logging off ensures that the cookies from your session are cleared from your machine. If you close your browser without logging out, these cookies might not be cleared (depending on which browser you use) and could be reused if you were to log on to CentraSite Control before your earlier session's timeout period had elapsed.

Navigation Controls in CentraSite Control

The main navigation controls in CentraSite Control are:

■ Navigation bar

You use the tabs and menus on the navigation bar to select the task that you want to perform. The tabs and submenus that appear on this bar depend upon the edition of CentraSite that you are using and the role(s) assigned to your user account.

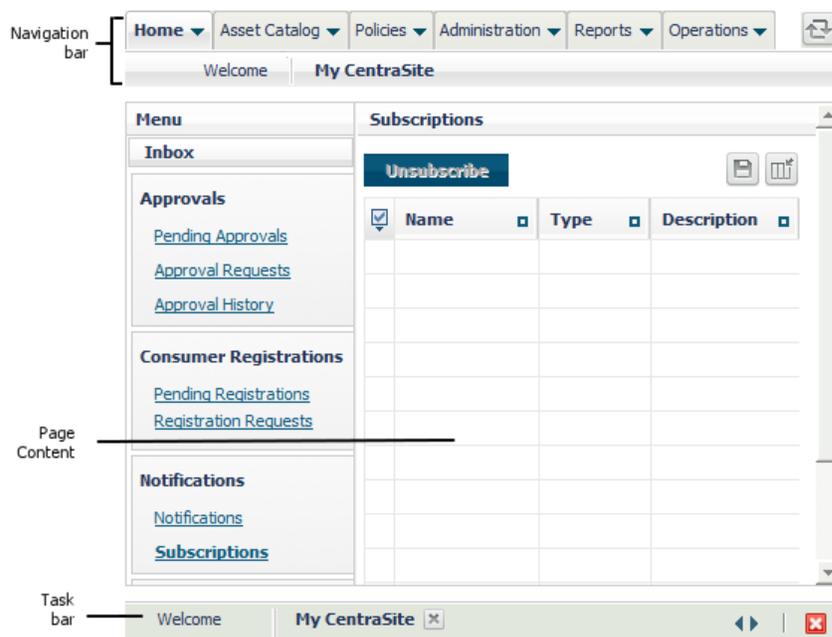
■ Page content

This is the area in which CentraSite Control displays the details associated with the task selected. The actual content that you will see for a task depends upon the edition of CentraSite that you are using as well as the role-based and instance-based permissions associated with your user account.

■ Task bar.

The task bar maintains the list of all open tasks. You can use the task bar to switch from one open task to another. You can also use the task manager to close an individual task or to close all open tasks in a single step.

Following is an example of the **My CentraSite** page that you access through the **Home > My CentraSite** menu.

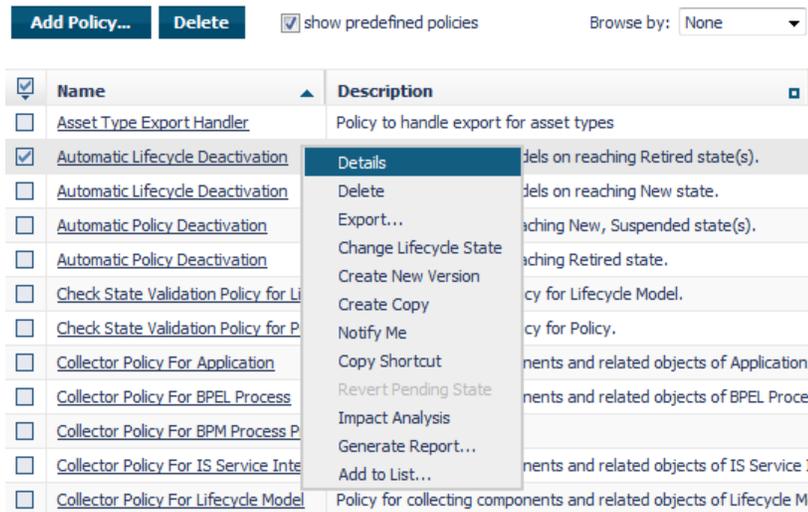


Note: CentraSite Control provides various controls that enable you to navigate from one page to another in the user interface. To ensure actions that are carried out correctly, always use the navigation controls provided within the CentraSite Control user interface and not the general navigation controls (page forward, page backward, and so on) available in your browser.

Context Menu

A context menu enables you to perform an action on a specific object in a list.

To display the object's context menu, right-click the object upon which you want to act. The set of actions in the context menu depends upon the type of object that you have selected.



Actions Menu

Many screens offer the **Actions** menu in addition to the context menu. The **Actions** menu is a drop-down menu that contains actions that you can perform on one or more objects in the currently displayed list of objects.

The set of actions offered by the **Actions** menu on a particular screen is usually a subset of the actions offered by the context menu. This is because the actions in the **Actions** menu can be applied to multiple selected objects. The set of actions is restricted to just the actions that can be performed on all of the screen's listed objects. In particular, if the screen contains objects of several object types, actions that are relevant for one object type might not be relevant for another object type. Therefore, such actions are not offered in the menu.

To use this menu, mark the checkbox for the required objects in the currently displayed list of objects, then open the **Actions** menu and select the required action. This action is applied to the selected objects.

Setting Display Options in CentraSite Control

The display settings specify the language in which CentraSite Control displays the user interface (assuming the appropriate language pack is installed on the Software AG Runtime) and the time zone in which time-stamped events are rendered when you view the activity logs and other dated information in CentraSite Control.

➤ **To set display preferences**

1. In CentraSite Control, click **My Account**.
2. On the My Account page, specify the following:

In this field...	Specify...
Language	The language in which you want CentraSite Control to be displayed.
Locale	The language of the operating system locale being used on your machine. If you are the CentraSite Administrator you can restrict the locale availability.
Time Zone	The time zone in which you want timestamped log information rendered when it is displayed to your user account.
Date Format	The format in which you want CentraSite Control to display dates to you.
Time Format	The format in which you want CentraSite Control to display times to you.
Render numbers with 1000 separator	Whether you want CentraSite Control to display numeric values with a thousands' separator character (for example, 10,000).
Start Page	Your preferred start page. You can specify whether the start page should be the Welcome page (the default start page) or the Inbox page (containing your Inbox and the other elements of the My CentraSite page).

3. Click **Save**.

Important:

Modifications made to the **Language**, **Locale**, and **Start Page** settings take effect at the next log on. All other modifications are immediately active.

Restricting the Locale

This property allows the CentraSite Administrator to restrict the locale CentraSite Control user interface to users. This setting is available in the **Locale** field in My Account page.

This setting does not allow users to change their locale settings in the My Account page. Thus, logged on users are restricted to the locales allowed by the administrator.

If you are the CentraSite Administrator, you can define the locales in the **Allowed Locale** field. Only these defined locales will be available in the **Locale** field for every other user who logs in to the CentraSite Control.

When User is Restricted with Multiple Locale(s)

If the administrator had restricted to more than one locale, then when a user logs into the CentraSite Control one of the following happens:

- If you are an existing user and your locale matches with the allowed locale(s) in CentraSite Control, then you are permitted to directly logon to the CentraSite Control.
- If you are an existing user and your locale does not match with the allowed locale(s) in CentraSite Control, then you have to select the allowed locale(s) from the **Locale Settings** pop-up.
- If you are a new user, then you have to select the allowed locale(s) from the **Locale Settings**.

When User is Restricted with a Single Locale

If the administrator had restricted to just one locale, then when a user logs on to the CentraSite Control one of the following happens:

- If you are an existing user and your locale matches with the allowed locale in CentraSite Control, then you are permitted to directly log on to the CentraSite Control.
- If you are an existing user and your locale does not match with the allowed locale in CentraSite Control, then CentraSite Control automatically sets the locale and displays that the current locale is modified to match with the allowed locale.
- If you are a new user, then CentraSite Control by default sets the allowed locale.

> To restrict the locale

1. In CentraSite Control, click **My Account**.
2. In the **My Account** page, select the locale(s) from the **Allowed Locales** field in System-wide Settings panel.

If you want to specify multiple locales, use the plus button to add additional rows.

3. Select **Save** to confirm the change.

The changes take effect immediately and you are redirected to the logon page. If you do not specify a locale and cancel the setting, you are still redirected to the logon page.

Viewing or Editing Information About Your User Account

> To view information about your user account

1. In CentraSite Control, click **My Account**.
2. In the **Users** list, select the user name that represents your account.
 - To modify general information about your user account (name, email address, postal address), edit the contents of the **User Information** box and the **Additional Information**

tab as necessary. (Depending on the role or roles associated with your user account, you might not be permitted to edit all fields.)

- Use the **Groups** and **Roles** tabs to view the user groups and roles with which your account is associated.
- Use the **Assets** tab to display the list of assets that you currently own.
- You may want to add some metadata information that you feel is relevant to your user account. To do this, you can select the **Object Specific Properties** tab and define a set of one or more properties, each consisting of one keyword and one or more values associated with the keyword. The keywords you provide and their values can be selected.

3. Click **Save**.

My Favorites

The My Favorites feature enables you to create lists and shortcuts to items that you use frequently. Using My Favorites, you can:

- Display the list of assets that you own (or that you consume) with a single click.
- Create quick links to selected assets in the catalog.
- Create a quick link to a saved search or XQuery.
- Create a quick link to a list of assets of a particular category or type.

The My Favorites list is available on the **Home > My CentraSite** page. By default, it contains the following entries:

Choose this entry...	To...
 Assets I Consume	Display the list of assets for which you are registered as a consumer.
 Assets I Provide	Display the list of assets that you own.

Additionally, you can add any of the following entries to My Favorites:

Entry	Description
 Individual asset	Displays the details for a specified asset. This type of entry acts as a shortcut to the selected object.
 Individual report template	Displays the details for a specified report template. This type of entry acts as a shortcut to the selected object.
 List	Serves as a folder for a collection of objects. Each such list can contain a combination of objects that have different object types, such as assets, taxonomies, organizations and users.

Entry	Description
 Saved search	Executes a specified search and displays the resulting list of assets.
 Saved XQuery	Executes a specified query and displays the resulting list of assets.
 List of Type	Displays a list of objects of a specified type.
 List of Category	Displays a list of assets of a specified taxonomy category.

Smart Lists

The following types of entries are called *smart lists*:

- Saved search
- List of Type
- List of Category

A smart list produces a dynamic result that is based on a query. For example, when you select a List of Type or a List of Category entry, CentraSite Control executes the appropriate query against the registry and returns a list of objects that matches the requested object type or taxonomy category.

Viewing My Favorites

> To access My Favorites

1. In CentraSite Control, select **Home > My CentraSite**.

The My CentraSite page is displayed.

2. Select **My Favorites** from the **Menu** panel.

Adding Shortcuts to Individual Objects

Using My Favorites, you can create individual shortcuts to the following types of objects:

- Assets (any type)
- Report templates

Adding a shortcut to My Favorites enables you to jump directly to the detail page for the specified asset or report template. Additionally, you can perform additional actions on the asset or report template using the shortcut's context menu. For example, if the shortcut is to a report template, you can execute the report directly from the shortcut's context menu.

When you add a shortcut to My Favorites, be aware that:

- If you add a shortcut to an object that already exists in your My Favorites list, CentraSite Control ignores your request. It will not create a duplicate shortcut.
- If you add a shortcut to My Favorites and the underlying asset or report template is subsequently deleted from the registry, the shortcut is automatically removed from My Favorites.

➤ To add a shortcut to an individual object to My Favorites

1. Use the **Asset Catalog** tab or the **Reports** tab to display a list containing the asset or the report template that you want to add to My Favorites.
2. Locate the asset or report template that you want to add to My Favorites and from its context menu select **Add to Favorites**.
3. Provide a name for the shortcut, then click **OK**.

Adding Shortcuts to Multiple Assets in a Single Operation

➤ To add shortcuts to multiple assets in a single operation

1. In CentraSite Control, use either the Browse or the Search feature in the asset catalog to select a list of the assets you want to add.
2. Mark the checkbox of each asset for which you wish to create a shortcut.
3. In the **Actions** menu, click **Add to Favorites**.

The shortcuts for the selected assets are then created in the My Favorites list. The name of each shortcut is the name of the asset to which the shortcut points. Note that if two or more of the assets have the same name, their shortcuts also have the same name.

Tip:

The **Add to Favorites** command is also available from the **Actions** menu on the details page of each asset and report template.

Adding an Object to My Favorites

In CentraSite Control, you can combine objects of different types into so-called *lists*. These are user-defined logical collections that allow you to treat the whole list as a single entity. You can, for example, create a list containing assets, organizations and users, then perform an Export operation on the list, the resulting export set contains all of the objects contained in the list.

When you add objects to a list, CentraSite does not physically copy the objects into the list. Instead, the list contains pointers to the objects. This means, for example, that if you delete a list, you only delete the pointers to the physical objects but you do not delete the physical objects themselves.

The lists that you create are shown in CentraSite Control under **My CentraSite > My Favorites**.

➤ **To add an object to a list in My Favorites**

1. In CentraSite Control, display the objects that you wish to add to the list.

For example, if you wish to add an asset to a list, select **Asset Catalog > Browse** and locate the asset that you want to add to My Favorites.
2. Open the object's context menu and select **Add To List**.
3. To add the object to an existing list, select the list from the drop-down list in the dialog box. Otherwise type a name in the text box to create a new list.
4. Select **OK**.

Adding Multiple Objects to a List in a Single Operation

➤ **To add multiple assets to a list in My Favorites in a single operation**

1. In CentraSite Control, display that shows the objects that you wish to add to the list.
2. Mark the checkbox of each object you want to add.
3. In the **Actions** menu, click **Add To List**.

Removing an Object from a List

When you remove an object from a list, you just remove the name of the object from the list. The physical object that the list entry points to is not affected in any way.

If you add an object to a list and later delete the original object that the list entry points to, the entry in the list is deleted automatically.

➤ **To remove an object from a list in My Favorites**

1. Display My Favorites and select the list that contains the object that you want to remove.

This displays the names of the objects contained in the list.
2. Select the object you want to remove from the list.
3. Select **Delete**.

Adding a Saved Search to My Favorites

Adding a saved search to My Favorites enables you to execute the search with a single click. When you select a saved search in My Favorites, CentraSite Control executes the search and displays the results.

➤ **To save a search to My Favorites**

1. In CentraSite Control, select **Asset Catalog > Search**.
2. Use the **Keyword** tab or the **Advanced** tab to define your search criteria.
3. Select **Save**.

You can save your search without first executing it.

4. Specify a name for the saved search and click **OK**.

If the name you specify for the saved search already exists, you are asked to provide a different name.

Viewing or Editing a Saved Search

➤ **To view or edit a saved search**

1. Display My Favorites and select the saved search that you want to view or edit.
2. In the results pane, click **Refine**.
3. Examine and redefine your search criteria as necessary.
4. If you made modifications to the search criteria, click **Save** to save the changes.

The refined search is considered to be a new saved search rather than an update to an existing saved search, so CentraSite Control asks you to provide a name for the new saved search.

Adding an XQuery to My Favorites

Adding an XQuery to My Favorites enables you create a shortcut to an XQuery expression. When you select a saved XQuery in My Favorites, CentraSite Control opens the expression in the XQuery editor.

➤ **To save an XQuery to My Favorites**

1. In CentraSite Control, select **Asset Catalog > Search**.
2. Use the **XQuery** tab to define your XQuery expression.

3. Choose **Save**.

You can save your XQuery expression without first executing it.

4. Specify a name for the saved XQuery expression and click **OK**.

Adding a Category List to My Favorites

You use a category list to display the list of assets that belong to a specified taxonomy category. For example, if your site uses a taxonomy that classifies assets by project, you might want to create shortcuts that display the assets associated with your projects.

➤ To save a Category List to My Favorites

1. In CentraSite Control, select **Asset Catalog > Browse** to open the asset catalog.
2. In the **Browse By** list, select the taxonomy that contains the category that you want to add to My Favorites.
3. In the taxonomy tree, locate the category that you want to add to My Favorites and from its context menu and click **Add to Favorites**.
4. Specify a name for the category list and click **OK**.

Note:

If you have permission to access the **Administration > Taxonomies** task from the navigation bar, you can also add a category list to My Favorites from the context menu for a taxonomy listed on the Taxonomies page.

Adding a Type List to My Favorites

You use a type list to create a shortcut to a list of assets of a specified asset type (for example, Application Servers, Web Services, or XML Schemas).

➤ To save a Type List to My Favorites

1. In CentraSite Control, select **Asset Catalog > Browse** to open the asset catalog.
2. In the **Asset Types** list (in the upper left corner of the page), locate the asset type that you want to add to My Favorites.
3. Open the context menu for the asset type and click **Add to Favorites**.
4. Specify a name for the type list and click **OK**.

Note:

If you have permission to access the **Administration > Types** task from the navigation bar, you can also add a type list to My Favorites from the context menu for an object type listed on the Type Management page.

Removing Entries from My Favorites

Note:

When you remove an entry, any underlying assets to which the entry refers are not affected.

> To remove an entry from My Favorites

1. Display **My Favorites** and locate the entry that you want to remove.
2. From the context menu for that entry, click **Remove from Favorites**.

Renaming an Entry in My Favorites

Use the following procedure to rename a user-defined entry in My Favorites.

Note:

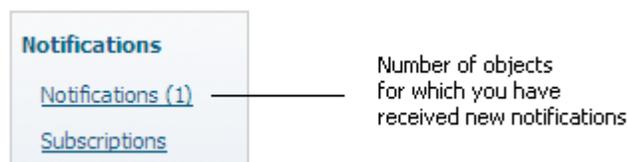
Some of the entries in My Favorites are predefined and cannot be renamed. If an entry cannot be renamed, the **Rename** command does not appear on the context menu.

> To rename an entry in My Favorites

1. Display **My Favorites** and locate the entry that you want to rename.
2. From the context menu for that entry, click **Rename**.
3. Specify a new name for the entry and click **OK**.

Notifications

Using the notification feature, you can request CentraSite to alert you when specified assets or policies are modified. If you have requested a notification for one or more assets or policies, CentraSite Control displays a number in the **Notifications** Inbox indicating how many objects on your notification list have been updated since the last time you viewed your notifications.



When you select the **Notifications** link in your inbox, CentraSite Control opens a two-pane screen. The upper pane displays the list of objects for which you have asked to receive notifications. The

lower pane displays the *change log* of the object currently selected in the upper pane. The change log is a record of the changes that have been made to the object since the object was initially created. The log indicates when a change was made and who made the change.

The upper pane lists the objects for which you are registered to receive notifications ...

Name	Description	Last Change
<input checked="" type="checkbox"/> BusinessForecast	This service implements the bus	2012-02-10 04:52 PM

... and the lower pane displays the change log for the object you selected in the upper pane.

Event Type	Date/Time	User	Comment
updated	2012-02-10 04:52 PM	INTERNAL\Administrator	Updated from Control
created	2012-02-10 04:50 PM	INTERNAL\Administrator	Updated from Control

Note:

When you create a notification request for an asset, your request is reflected on the asset's **Subscription** profile. The **Subscription** profile displays the list of users who are currently registered to receive notifications for the asset.

Creating a Notification Request

Use the following procedure to have CentraSite notify you when a specified asset or policy is modified.

> To create a notification request

1. In CentraSite Control, display the list that contains the asset or policy on which you want to receive notifications.
2. From the context menu for the asset or policy, click **Notify me**.

A message appears, informing you that the notification has added to the **Notifications** list.

Note:

If you see the **Remove From My Notifications** command on the context menu instead of the **Notify me** command, that indicates that you are already registered to receive notifications for the selected asset or policy.

3. Go to the **Notifications** list on the **My CentraSite** page to verify that the notification request has been added to your **Notifications** list.

Creating Notification Requests for Multiple Assets in a Single Operation

> To create notification requests for multiple assets in a single operation

1. In CentraSite Control, use either the Browse or the Search feature in the asset catalog to select a list of the assets for which you wish to create notification requests.
2. Mark the checkbox of each asset for which you wish to create a notification request.
3. In the **Actions** menu, click **Notify Me**.

Viewing Notifications You Have Received

> To view notifications you have received

1. Go to **Home > My CentraSite**.
2. Display the **Inbox** and click **Notifications** to open the **Notifications** screen.

Rows with bold text in the notification list in the upper pane indicate objects that have been updated since you last viewed this list.

3. To view the change list for an object, select the object in the notification list in the upper pane. When you select the object, *click anywhere in its row except on the object name*. If you click the object's name, CentraSite Control displays the details page for that object instead of the opening the object's change list.

Deleting a Notification Request

If you do not want to receive notifications for an asset or a policy you can delete the notification request.

> To delete a notification request

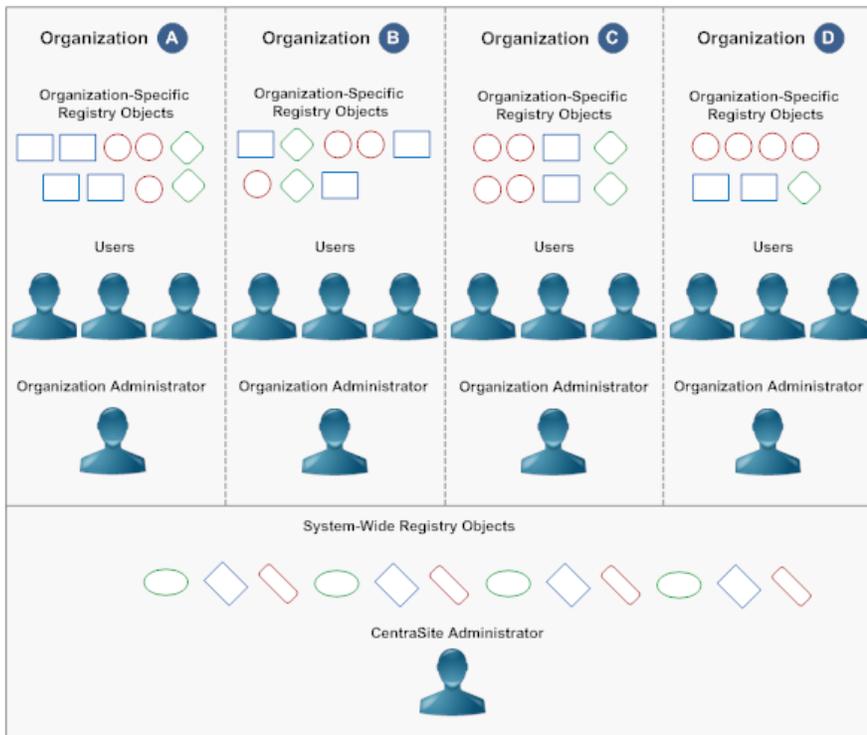
1. In CentraSite Control, display the list that contains the asset or policy for which you want to delete a notification request.
2. From the context menu for the asset or policy, click **Remove from My Notifications**.

2 Organization Management

- Introduction to Organizations 78
- Managing Organizations through CentraSite Business UI 84
- Managing Organizations through CentraSite Control 94
- Deleting Organizations through Command Line Interface 98

Introduction to Organizations

An *organization* represents an entity that owns a collection of assets. Organizations enable you to partition a registry into autonomous collections of objects that can be administered independently. You define organizations that represent actual entities within your enterprise, such as functional lines of business, regional subsidiaries or branches, legal entities, B2B partners (for example, suppliers and customers), projects, or departments. You can define parent-child associations between organizations to model the hierarchical structure of entities in your enterprise.



By default, only users within an organization have permission to view the organization's assets. If other users require access to the organization's assets, they must obtain explicit permission. Organizations enable you to restrict the visibility of a collection of assets to a specified group of users.

Organizations also function as a scoping mechanism for the following registry functions:

- Role-based permissions
- Lifecycle models
- Design and change-time policies
- Run-time policies
- Supporting documents

For example, organizations enable administrators to create lifecycle models, and design and change-time policies that apply only to the assets that belong to their organization.

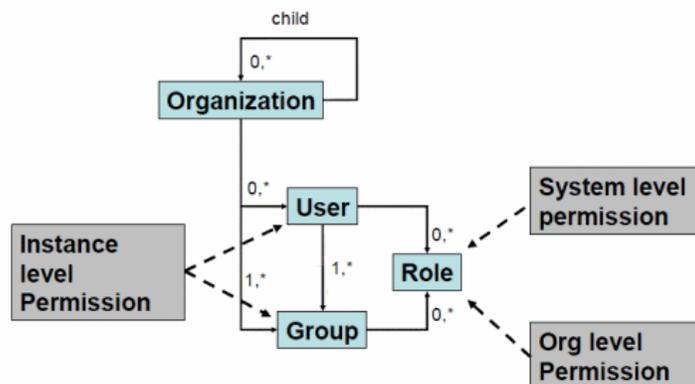
Each organization that you add is stored in the CentraSite registry as an object of the type Organization.

The information that you store for an organization includes:

- General information about the organization, such as the organization's name, the description of the organization, the contact person for the organization, and the address of the organization's web site.
- The CentraSite users who belong to the organization, as well as any group or role assignments.
- Details of any child organizations of the organization.

Basic Organization Structure

An organization is composed of users, groups, roles, and permissions. The structure you select determines how the assets in your registry are organized. It also plays a key role in controlling who can access various assets of the registry. The users that belong to an organization are permitted to access all assets of the organization. If other users require access to the organization's assets, they must obtain explicit permissions to do so.



An organization can have zero or more child organizations. Each child organization is a separate organization in its own right and has its own set of users, groups, roles, and permissions.

An organization can have one or more users. *A user can belong to only one organization.*

An organization has one or more groups. A group represents a set of users. Groups enable you to collectively apply permissions and other capabilities to a set of users. All organizations include the following predefined groups:

Group	Description
Users	All users belonging to the organization. The API requires all organizations to have this group.
Members	All users belonging to the organization or any of its descendants (that is, children, children's children, and so forth).

An organization has one or more roles that can be assigned to users or groups. By default, each organization includes the following set of roles: Organization Administrator, Policy Administrator, Asset Administrator, Asset Provider, and Asset Consumer. A role is a collection of system-level permissions and organization-level permissions. These permissions enable users to work with specific types of objects or perform certain tasks.

Instance-level permissions are used to give specific users or groups access to individual assets or registry objects. They enable you to apply fine-grain access controls to the assets in your organization.

Default Organization

CentraSite is installed with one pre-defined organization called Default Organization. The Default Organization owns the system-defined registry objects that CentraSite uses. You cannot delete the Default Organization or rename it.

As a best practice, avoid using the Default Organization as an ordinary organization. Instead, treat it as the home for system-wide objects such as asset types, taxonomies, gateways and system-wide policies, and restrict membership in this organization to a small number of administrative users.

To create and manage a top-level organization (that is, an organization that is a sibling of the Default Organization), you must belong to a role that has the Manage Organizations system-level permission. By default, users with the CentraSite Administrator role have this permission, and can assign this permission to other roles. The Manage Organizations permission enables you to manage all organizations (including the Default Organization). This permission allows you to create, view, edit, and delete any object within any organization.

Child Organization

You can use parent-child associations to represent hierarchical relationships among organizations. Parent-child relationships enable you to collectively administer certain aspects of the associated organizations. For example, after you establish a parent-child relationship between organizations, administrators can collectively grant roles and permission to users that belong to a parent organization or any of its children. The administrator of a parent organization also has administrative control over the organization's descendants, and can manage users, assets, policies, and lifecycles within the parent organization and all of its descendants.

To create a parent-child relationship, you must create the child organization from within the parent organization. A child organization does not inherit any characteristics from its parent. It has its own administrator, set of users, policies, lifecycle models, and assets. The administrator of the parent organization also has administrative privileges in the child organization. A child organization can have additional child organizations of its own.

To create child organizations from an organization, you must belong to a role that has either the Manage Organizations permission for the organization in which you want to create the child organization or that permission for one of that organization's antecedents.

When defining your organizational structure, keep the following points in mind with respect to parent-child relationships:

- A child organization can belong to only one parent.

- You must create the parent organization first and then create the child from the parent. In other words, you cannot create two unrelated organizations and then, at a later time, establish a parent-child relationship between them.
- Once you associate organizations in a parent-child relationship, you cannot disassociate the organizations.
- You cannot move a child organization to a different parent.
- You cannot promote a child organization to become top-level parent organization.
- A child does not inherit properties or objects from its parent and organization administrators in the parent organization are automatically allowed to administer the child organization. The parent's organization specific policies and lifecycles do not apply to its children. A child organization creates and maintains its own policies and lifecycles independent of its parent. In this respect, it is like any other organization. To impose policies and lifecycles on all organizations in the registry, you can create system-wide policies and lifecycles.

Consumer Organization

CentraSite gives every user in an organization the ability to act both as an Asset Consumer, where the user can view assets in the registry and an Asset Provider, where the user can publish assets into the registry. CentraSite does this by assigning both the Asset Provider role and the Asset Consumer role to the organization's Users group, which is the group comprising all users in the organization.

If you have groups of users who consume only assets, consider creating a separate organization for those users. Remove the Asset Provider role from the organization's Users group so that the users in the organization have just the Asset Consumer role. Any organization that wants to extend assets to these consumers can give the consumer organization's Users group, permission to view the assets.

Organization Administrators and Primary Contacts

When you create an organization, you must specify a user to serve as the Organization Administrator and a user to serve as the Primary Contact.

- The *Organization Administrator* is a user that has the Organization Administrator role for the organization. An organization must have at least one user in this role. A user in one organization can serve as an Organization Administrator for another organization; however, this role is generally given to someone within the organization. An organization administrator performs administrative tasks for the organization, such as:
 - Adding users to the organization
 - Defining groups and roles
 - Defining custom lifecycle models for the organization
 - Creating child organizations

An organization administrator can also view, edit, and delete any asset, policy or lifecycle model that belongs to the organization or any of its descendants.

- The *Primary Contact* is a user who acts as the point-of-contact for an organization. An organization has just one primary contact. The user who is designated as the primary contact does not receive any additional roles or permissions. The same user can serve as the organization's administrator and its primary contact. You can assign a different user to each position. The primary contact is not required to be a user within the organization, but usually is in most of the cases.

Modeling your Organizations

Instead of organizing the content of your registry around the low-level departments on an organization chart, group it by higher-level concepts such as functional units, lines of business, process owners, legal entities (for example, subsidiaries and affiliates) or regional divisions. Think in terms of who owns the assets that resides in the registry or has primary responsibility for developing and maintaining them. Create organizations to represent those areas.

If you intend to give external partners access to CentraSite, create separate organizations for each partner. By partners we mean any external entities with which your enterprise interacts, such as suppliers and other vendors, dealers and distributors, customers, government agencies, trade organizations, and so on.

The figure shows the organizational structure that is defined in the starter kit example. This example has a level of granularity that is appropriate for defining organizations:

Example Organization Structure

<input checked="" type="checkbox"/>	Name	Parent Organization
<input type="checkbox"/>	Default Organization	
<input type="checkbox"/>	Shared IT Services	
<input type="checkbox"/>	Customer Care	
<input type="checkbox"/>	E-Commerce	
<input type="checkbox"/>	Trading Partners	
<input type="checkbox"/>	Acme Inc	Trading Partners
<input type="checkbox"/>	Flowers.com	Trading Partners

Organization	Description
Default Organization	This organization serves as the home for administrative and system-wide artifacts.
Shared IT Services	This organization represents Information Technology (IT) departments that operate across (that is, are shared by) other lines of business. This organization would include the Quality Assurance (QA) department, the SOA Competency Center and individuals who serve as enterprise-wide IT architects.
Customer Care	This organization serves as the home for administrative and system-wide artifacts.

Organization	Description
E-Commerce	This organization represents the department responsible for providing external-facing services including customer-facing portal sites and business-to-business services.
Trading Partners	This organization is the parent organization for all external partner organizations. It is a pseudo organization that is used to hold organizations that represent external parties.
Acme Inc.	This organization represents an external partner that provides or consumes assets. <i>This organization is a child of the Trading Partners organization.</i>
Flowers.com	This organization represents an external partner that provides or consumes assets. <i>This organization is a child of the Trading Partners organization.</i>

If you are using a multi-stage deployment, you might replicate the same organizational structure across all registries or you might adopt a different structure in each registry. For example, on the *creation* CentraSite you might use the organizational structure like the one described above. However, on the *consumption* CentraSite, you can define just two organizations: an operations organization, which owns and manages all of the assets in the registry, and a consumer organization, which contains users who can browse the consumption registry. The organizational structure that you adopt for other stages depends on your requirements.

Selecting an Organizational Strategy

Apart from selecting a deployment strategy, defining your organizational structure is one of the most critical deployment decisions. It is important to select a structure that is stable and endures over time. After establishing your registry's structure and governance processes in place, it is difficult to make fundamental changes to the way in which the registry is organized. Such a change would not only require you to transfer assets to different organizations, but may also require you to redefine the lifecycle models, policies, and permissions that support your governance environment.

When planning your organizational strategy, take the following points into consideration:

- In general, create organizations around the concept of asset visibility, ownership or responsibility. In the pre-production stages, use organizations to represent the major stakeholders involved in the pre-production aspects of the asset's development lifecycle. For example, service owners, developers, and the SOA Competency Center. In the production stage, use organizations to represent the groups of users who represent assets owners, consumers of assets, and the operators of the production services.
- If a particular group of users requires the use of custom lifecycle models or design-time policies, create a separate organization for those users.
- If you have groups of users whose needs are consumer only, create separate consumer organizations for those users.
- Keep in mind that a user can belong to only one organization within a CentraSite registry. If you have a user who creates assets for multiple organizations, add the user to the organization

in which he or she works primarily. Then assign roles to the user that enable the user to create assets in other organizations.

- Keep in mind that an asset also belongs to only one organization. To make an asset accessible to multiple organizations, you must give the users in those organizations permission to access the asset.
- Avoid creating an organizational structure that is too fine-grained. Keeping the structure of the registry synchronized with your low-level development teams and work-units requires a significant amount of work. Moreover, a fine-grained structure is generally not needed in order to govern your assets effectively.

Managing Organizations through CentraSite Business UI

This section describes operations you can perform to manage the organizations through CentraSite Business UI.

Adding Organization

Pre-requisites:

To create a new organization, you must have the Manage Organizations permission in CentraSite.

- To create a top-level organization, you must have the Manage Organizations permission at the system level.
- To create a child organization, you must have the Manage Organizations permission on the organization's parent (or other antecedent).

By default, CentraSite contains the Default Organization. You use the **Organizations** activity to create other top-level organizations (that is, organizations that are siblings of the Default Organization).

When you add a new organization to CentraSite, CentraSite creates the new organization in the registry and populates the organization with a set of default objects (roles, users, policies, a folder in the supporting document library, and so on).

➤ To add an organization

1. In the CentraSite Business UI activity bar, click **Organizations**.

This displays a list of defined organizations. Also, the actions bar displays a set of actions that are available for working with organizations.

2. Click **Add Organization**.
3. Provide the required information for each of the displayed data fields. To access a data field, click on the link for the field.

In this field...	Do the following...
Name	Type a name for the new organization. An organization name can contain any character (including spaces).
	<p>Note: An organization name does not need to be unique within the CentraSite registry. However, to reduce ambiguity, you should avoid giving the same name to multiple organizations.</p>
Description	<i>Optional.</i> Type a description for the new organization. This description appears when a user displays the list of organizations on the CentraSite Business UI.
Administrator(s)	<p>Assign an administrator for this organization.</p> <p>Each organization requires at least one administrator. The administrator receives all permissions required to perform any administrator-level operation on the organization. As you type characters in this field, the dialog displays all known users whose user ID begins with the characters you have provided. If your user repository is based on LDAP, the search additionally looks for any user that has an LDAP attribute value that matches the characters you type.</p> <p>You can use wildcards in this field. For example *abc finds users with user IDs such as org1abc or department52abc, or users with LDAP attributes that have these values. The wildcard * represents any number of characters; the wildcard % represents any single character.</p> <p>The list of user IDs returned displays users who are already registered in the organization, as well as users in the user repository who are not yet registered as users in CentraSite. If you select one of the unregistered users, this user becomes a registered user in the organization and is assigned to be an administrator of the organization.</p> <p>You can define more than one administrator for the organization. Click + beside the field labeled Administrator(s) to add additional administrators.</p>
Web Page	<i>Optional.</i> Type the website URL of the organization.
Address	<i>Optional.</i> Provide the address information for the primary location of this organization.
Contact Information	<p>Provide the contact information for the primary contact of this organization.</p> <p>a. Enable the Select Administrator as Primary Contact option if you want the individual specified in the Administrator field to serve as the organization's primary contact.</p> <p>—OR—</p>

In this field...**Do the following...**

Click **Pick Existing** or **Create New** to select a user from CentraSite's user database or from an external directory, respectively.

- b. Provide the phone and fax numbers for the primary contact. You can provide multiple phone and fax numbers.

4. Click **Save**.

Viewing the Organization List

In CentraSite Business UI, you can display the list of organizations in one of the following ways:

- Using the Typeahead **Search**.
- Using the **Browse** functionality.
- Using the **Organizations** activity.

If you have the Manage Organizations permission, the **Organizations** activity allows you to view a list of the organizations that you are allowed to access. The list includes all organizations for which you are the owner. If you have the Manage Organizations system-level permission, the list includes all the organizations defined in the registry.

By default, all users have implicit (and irrevocable) View permission on organizations.

> To view the list of organizations

- Using the Typeahead Search. In the Search  box, click **Everything**, and type the name of the organization in the search text box.

As you type the partial text, CentraSite returns the list of organizations that meet your search text.

- Using the Browse functionality. Click the **Browse** link that is located in the upper-left corner of the menu bar.
 1. In the **Additional Search Criteria** list, select **Asset Types**, and then click **Choose**. This opens the **Choose Asset Types** dialog box.
 2. Click the chevron next to **Everything** option button. Then select the **Organization** check box and click **OK**.

This displays a list of defined organizations in the Search Results page.

3. To filter the list to see just a subset of the available organizations in the Search Results page, type a partial string in the **Keyword** text box. Add the specified keyword to the Search Recipe, by clicking the plus symbol next to the text box, or press Enter.

The Search Results page provides the following information about each organization:

Column	Description
Name	Name of the organization.
Description	The description for the organization.
Type	The asset type, Organization .
Last Updated	The date on which the organization detail was last modified.
Owner	The name of the user who created or imported this organization.

For each organization, the list includes various attributes of the organization such as the organization name and the owner. You can adjust the view to show or hide any of the available attributes by opening the drop-down list labeled **View** and selecting the attributes that you want to include in the view, and clearing the attributes that you do not want to view.

You can change the order in which the attributes are displayed by opening the drop-down list labeled **Sort by**. The list displays all the attributes that are selected in the **View** drop-down list. The order in which the attributes appear in the **Sort by** drop-down list is the order in which the attributes appear for each displayed organization. To change the order in which any given attribute is displayed, select the attribute in the drop-down list **Sort by** and use the arrows to move the attribute to the required position.

- Using the Organizations activity. In the CentraSite Business UI activity bar, click **Organizations**.

This displays a list of defined organizations in the Organizations page.

The Organizations page provides the following information about each organization:

Column	Description
Name	Name of the organization.
Description	The description for the organization.
Last Updated	The date on which the organization detail was last modified.
Primary Contact	The primary contact of the organization.
Web Page	The website URL of the organization.

Also, the actions bar displays the set of actions that are available for working with organizations.

For each organization, the list includes various attributes of the organization such as the organization name and the owner. You can adjust the view to show or hide any of the available attributes by opening the drop-down list labeled **View** and selecting the attributes that you want to include in the view, and clearing the attributes that you do not want to view.

You can change the order in which the attributes are displayed by opening the drop-down list labeled **Sort by**. The list displays all the attributes that are selected in the **View** drop-down list. The order in which the attributes appear in the **Sort by** drop-down list is the order in which the attributes appear for each displayed organization. To change the order in which any given attribute is displayed, select the attribute in the drop-down list **Sort by** and use the arrows to move the attribute to the required position.

Modifying Organization Details

Pre-requisites:

To modify organization details, you must have the Manage Organizations permission in CentraSite.

You modify an existing organization information using the Search Results page or the Organizations page in CentraSite Business UI.

The following general guidelines apply when modifying an existing organization information:

- You cannot modify the name of the Default Organization (even if you have the default permissions associated with the CentraSite Administrator role).
- You can change the Organization Administrator and the Primary Contact of an organization when required, however you cannot leave these positions unassigned.
- You cannot change an organization's parent assignment (that is, you cannot move a child organization from one parent to another).

➤ To modify organization details

1. In CentraSite Business UI, display the list of organizations in one of the following ways:
 - **Using the Typeahead Search.** In the **Search** drop-down list, click **Everything**, and type the name of the organization in the search text box.

As you type the partial text, CentraSite returns the list of organizations that meet your search text.
 - **Using the Browse functionality.** Click the **Browse** link that is located in the upper-left corner of the menu bar.
 1. In the **Additional Search Criteria** list, select **Asset Types**, and then click **Choose**. This opens the **Choose Asset Types** dialog box.
 2. Click the chevron next to **Everything** option button. Then select the **Organization** check box and click **OK**.

This displays a list of defined organizations in the Search Results page.

- **Using the Organizations activity.** In the CentraSite Business UI activity bar, click **Organizations**.

This displays a list of defined organizations in the Organizations page.

2. Click an organization whose details you want to modify.

This opens the Organization Details page. The details include:

- The organization's basic information (organization's postal address, the web page URL, the name of the organization's administrator, the primary contact person, and a general description of the organization).
- The immediate child organizations of this parent organization.
- The users who belong to this organization.
- The groups which belong to this organization.
- The roles that belong to this organization.

Also, the actions bar displays the set of actions that are available for working with the displayed organization.

3. To modify the organization's details displayed in the **Basic Information** profile, click **Edit**.
4. Modify the values for the organization's fields in the Organization Details page as required.
5. Click **Save** to update the organization information.

Deleting Organizations

Pre-requisites:

To delete an organization, you must have the Manage Organizations permission in CentraSite.

- To delete a top-level organization, you must have the Manage Organizations permission at the system level.
- To delete a child organization, you must have the Manage Organizations permission on the organization's parent (or other antecedent).
- To delete an organization that has one or more child organizations, you must first delete the child organizations before you can delete the organization.

You delete an existing organization from the CentraSite Registry Repository using the Search Results page or using the Organizations page.

You cannot delete an organization if:

- It is the Default Organization (even if you have the default permissions associated with the CentraSite Administrator role).
- One of its users serves as the primary contact of another organization.

> To delete organizations

1. In CentraSite Business UI, you can display the list of organizations in one of the following ways:

- **Using the Typeahead Search.** In the **Search** drop-down list, click **Everything**, and type the name of the organization in the search text box.

As you type the partial text, CentraSite returns the list of organizations that meet your search text.

- **Using the Browse functionality.** Click the **Browse** link that is located in the upper-left corner of the menu bar.

1. In the **Additional Search Criteria** list, select **Asset Types**, and then click **Choose**. This opens the **Choose Asset Types** dialog box.

2. Click the chevron next to **Everything** option button. Then select the **Organization** check box and click **OK**.

This displays a list of defined organizations in the Search Results page. Also, the actions bar displays the set of actions that are available for working with organizations.

- **Using the Organizations activity.** In the CentraSite Business UI activity bar, click **Organizations**.

This displays a list of defined organizations in the Organizations page. Also, the actions bar displays the set of actions that are available for working with organizations.

2. Select the check box of a single organization, or select the check boxes of multiple organizations you want to delete.
3. On the actions bar of the Search Results page or the Organizations page, click **Delete**.
4. Click **Yes** in the confirmation dialog box.

The organization is permanently removed from the CentraSite Registry Repository.

Adding Child Organization

Pre-requisites:

To create child organizations from an organization, you must have the Manage Organizations permission for the organization in which you want to create the child organization or that permission for one of that organization's antecedents.

You create a new child organization using the **Add Organization** action in the Organization Details page.

Each organization can have one or more child organizations, each of which in turn can also have child organizations, and so on. This allows you to model hierarchies such as, for example, when a large corporation consists of many independently operating companies that in turn have regional subdivisions.

> To add a child organization

1. In the CentraSite Business UI activity bar, click **Organizations**.

This displays a list of defined organizations in the Organizations page.

2. Click an organization to which you want to add the child organization.

This opens the Organization Details page. Also, the actions bar displays the set of actions that are available for working with the displayed organization.

3. Click **Add Child Organization**.

4. Provide the values for the child organization's fields in the Organization Details page as required.

The fields displayed are the same fields as that for the parent organization.

In this field...	Do the following...
Name	Type a name for the child organization. An organization name can contain any character (including spaces). <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note: An organization name does not need to be unique within the CentraSite registry. However, to reduce ambiguity, you should avoid giving the same name to multiple organizations.</p> </div>
Description	<i>Optional.</i> Type a description for the child organization. This description appears when a user displays the list of organizations on the CentraSite Control.
Administrator(s)	Assign an administrator for this organization. <p>Each organization requires at least one administrator. The administrator receives all permissions required to perform any administrator-level operation on the organization. As you enter characters in this field, the dialog displays all known users whose user ID begins with the characters you have entered. If your user repository is based on LDAP, the search additionally looks for any user that has an LDAP attribute value that matches the characters you type.</p> <p>You can use wildcards in this field. For example <code>*abc</code> finds users with user IDs such as <code>org1abc</code> or <code>department52abc</code>, or users with LDAP attributes that have these values. The wildcard <code>*</code> represents any number of characters; the wildcard <code>%</code> represents any single character.</p> <p>The list of user IDs returned displays users who are already registered in the organization, as well as users in the user repository who are not yet registered as users in CentraSite. If you select one of the unregistered</p>

In this field...	Do the following... users, this user becomes a registered user in the organization and is assigned to be an administrator of the organization. You can define more than one administrator for the organization. Click + beside the field labeled Administrator(s) to add additional administrators.
Web Page	<i>Optional.</i> Type the website URL of the organization.
Address	<i>Optional.</i> Provide the address information for the primary location of this organization.
Contact Information	Provide the contact information for the primary contact of this organization. a. Enable the Select Administrator as Primary Contact option if you want the individual specified in the Administrator field to serve as the organization's primary contact. —OR— Click Pick Existing or Create New to select a user from CentraSite's user database or from an external directory, respectively. b. Provide the phone and fax numbers for the primary contact. You can provide multiple phone and fax numbers.

5. Click **Save**.

Modifying Child Organization's Details

Pre-requisites:

To modify child organization details, you must have the Manage Organizations permission on the organization's parent (or other antecedent).

You modify an existing child organization information using the Search Results page or using the Organization Details page.

➤ To view or modify a child organization

1. In the CentraSite Business UI activity bar, click **Organizations**.

This displays a list of defined organizations in the Organizations page.
2. Click the parent organization that contains the required child organization.
3. In the navigation panel, select **Child Organizations**.

This displays a list of all the child organizations of this parent organization.

4. Click a child organization whose details you want to modify.

The details include:

- The child organization's basic information (organization's postal address, the web page URL, the name of the organization's administrator, and a general description of the organization).
- The immediate child organizations of this organization.
- The users who belong to this organization.
- The groups which belong to this organization.
- The roles that belong this organization.

Also, the actions bar displays the set of actions that are available for working with the displayed child organization.

5. To modify the child organization's details displayed in the **Basic Information** profile, click **Edit**.
6. Modify the values for the child organization's fields in the Child Organization Details page as required.
7. Click **Save** to update the organization information.

Deleting Child Organizations

Pre-requisites:

To delete a child organization, you must have the Manage Organizations permission on the organization's parent (or other antecedent).

You delete a child organization from the CentraSite Registry Repository using the Search Results page or using the Organization Details page.

You have to delete all the child organizations and their associations with the parent organization if you want to delete the parent organization.

> To delete a child organization

1. In the CentraSite Business UI activity bar, click **Organizations**.

This displays a list of defined organizations in the Organizations page.

2. Click the parent organization that contains the required child organization.

3. In the navigation panel, select **Child Organizations**.

This displays a list of all the child organizations of this parent organization.

4. Hover over the child organization you want to delete.

This displays icons for one or more actions that you can perform on the organization.

5. Click **Delete**.

6. When you are prompted to confirm the delete operation, click **Yes**.

The organization is permanently removed from the CentraSite Registry Repository.

Managing Organizations through CentraSite Control

This section describes operations you can perform to manage the organizations through CentraSite Control.

Adding Organization

Pre-requisites:

To create a new organization, you must have the Manage Organizations permission in CentraSite.

- To create a top-level organization, you must have the Manage Organizations permission at the system level.
- To create a child organization, you must have the Manage Organizations permission on the organization's parent (or other antecedent).

By default, CentraSite contains the Default Organization. You use the following procedure to create other top-level organizations (that is, organizations that are siblings of the Default Organization).

When you add a new organization to CentraSite, CentraSite creates the new organization in the registry and populates the organization with a set of default objects (roles, users, policies, a folder in the supporting document library, and so on).

➤ To add an organization

1. In CentraSite Control, go to **Administration > Organizations**.

This displays a list of defined organizations in the Organizations page.

2. Click **Add Organization**.

3. In the **Organization Information** panel, provide the required information for each of the displayed data fields:

In this field...	Do the following...
Name	Type a name for the new organization. An organization name can contain any character (including spaces). Note: The organization name does not need to be unique within the CentraSite registry. However, to reduce ambiguity, you should avoid giving the same name to multiple organizations.
Description	<i>Optional.</i> Type a description for the new organization. This description appears when a user displays the list of organizations on the CentraSite Control.
Administrator	Assign an administrator for this organization. <ul style="list-style-type: none"> ■ To assign an existing user to this position (that is, a user that is already defined within CentraSite), click Pick Existing and select the user from CentraSite's existing database. ■ To add a new user to CentraSite to serve in this position, click Create New and select the user from the external directory.
Web Page	<i>Optional.</i> Type the website URL of the organization.

4. In the **Address Information** tab, provide the required information for each of the displayed data fields:

In this panel...	Do the following...
Address	<i>Optional.</i> Provide the address information for the primary location of this organization.
Contact Information	Provide the contact information for the primary contact of this organization. <ul style="list-style-type: none"> a. Enable the Select Administrator as Primary Contact option if you want the individual specified in the Administrator field to serve as the organization's primary contact. —OR— Click Pick Existing or Create New to select a user from CentraSite's user database or from an external directory, respectively. b. Provide the phone and fax numbers for the primary contact. You can provide multiple phone and fax numbers.

5. To specify any custom properties (key-value pairs) for the organization, select the **Object-Specific Properties** tab, and specify the key-value pairs as follows:

- a. Click **Add Property**.
 - b. In the **Add Object-Specific Properties** dialog box, type a keyword and value for the property. You can add multiple values for a single property.
 - The name of the property can consist of letters, numbers, and the underscore character (_). It cannot contain a space or other special characters.
 - You can optionally supply a namespace for the property.
 - c. Click **OK**.
6. Select the **Attributes** tab and specify the attributes as necessary.

Attributes that are marked with an asterisk (*) are required. You must specify all the required attributes.

Note:

You see the **Attributes** tab only if a user with administration permissions has added custom attributes to the Organization object type definition.

7. Click **Save**.

Modifying Organization Details

Pre-requisites:

To modify organization details, you must have the Manage Organizations permission in CentraSite.

By default, all users have implicit (and irrevocable) view permissions on organizations. You cannot modify the name of the Default Organization (even if you have the default permissions associated with the CentraSite Administrator role). You can change the organization administrator and the primary contact of an organization when required, however you cannot leave these positions unassigned. You cannot change an organization's parent assignment (that is, you cannot move a child organization from one parent to another).

➤ To modify an organization's properties

1. In CentraSite Control, go to **Administration > Organizations**.

This displays a list of defined organizations in the Organizations page.

2. To view a subset of the available organizations, type a partial string in the **Search** field.

CentraSite applies the filter to the **Name** column. The **Search** field is a type-ahead field, so as you type the characters, the display is updated to display only those organizations whose name contains the specified characters. The wildcard character % is supported.

3. Click an organization whose details you want to modify.

4. From the organization's context menu, select **Details**.
5. Modify the values for the child organization's fields in the Edit Organization page as required.
6. Click **Save** to update the organization information.

Deleting Organizations

Pre-requisites:

To delete an organization, you must have the Manage Organizations permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

- To delete a top-level organization, you must have the Manage Organizations permission at the system level.
- To delete a child organization, you must have the Manage Organizations permission on the organization's parent (or other antecedent).
- To delete an organization that has one or more child organizations, you must first delete the child organizations before you can delete the organization.

You cannot delete an organization if:

- It is the Default Organization (even if you have the default permissions associated with the CentraSite Administrator role).
- One of its users serves as the primary contact of another organization.

➤ To delete an organization

1. In CentraSite Control, go to **Administration > Organizations**.

This displays a list of defined organizations in the Organizations page.

2. Right-click an organization you want to delete.

You can select multiple organizations to delete them.

3. Click **Actions > Delete**.

4. Click **OK** in the confirmation dialog box.

Each selected organization is permanently removed from the CentraSite registry or repository.

Adding Child Organization

Pre-requisites:

To create child organizations from an organization, you must have the Manage Organizations permission for the organization in which you want to create the child organization or that permission for one of that organization's antecedents.

Each organization can have one or more child organizations, each of which in turn can also have child organizations, and so on. This allows you to model hierarchies such as, for example, when a large corporation consists of many independently operating companies that in turn have regional subdivisions.

> To add a child organization

1. In CentraSite Control, go to **Administration > Organizations**.

This displays a list of defined organizations in the Organizations page.

2. Select the organization for which you want to create a child organization.
3. From the organization's context menu, select **Details**.
4. Select the **Child Organizations** tab and click **Add Child Organization**.
5. Provide the required information for the child organization.
6. Click **Save**.

Deleting Organizations through Command Line Interface

Pre-requisites:

To delete an organization through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

Before you run the tool, make sure that the target user for transferring the ownership of referenced objects is activated.

In some circumstances, you may not be able to delete an organization because internal objects that reference it cannot be deleted. This can happen when there are internal references to an organization object even though the organization is no longer the owner of any assets. There can also be references to the organization object in the audit log.

CentraSite provides a command tool named `delete Organization` for this purpose. The tool does the following:

- Transfers ownership of objects to the target user.

- Redirects internal references to the target user.
- Creates an OWNERSHIPTRANSFERRED event in the audit logs for every registry object that references the organization object.
- Remove the GUI configuration and the organization object.

Important:

The operation to delete an organization requires several steps that cannot run within a single transaction. This means every parallel running transaction is able to see intermediate results. Make sure no other activity is in progress while you run the tool. Moreover if there is a failure of any of the steps during the execution, the registry have an intermediate state. The original state cannot be recovered by rolling back the complete operation.

➤ To delete an organization

- Run the command `delete Organization`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd delete Organization [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -targetuser <TARGET-USER> -organizationName <ORGANIZATION-NAME>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
ORGANIZATION-NAME	The name of the organization you want to remove from the registry.
TARGET-USER	The name or UUID of a CentraSite user to whom you want to transfer ownership of the objects that are referenced by the organization identified by the parameter <code>-organizationName</code> .

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd delete Organization -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-targetuser LDAPDomain\Claire -organizationName MISOrganization
```

The response to this command could be:

```
Executing the command : delete Organization
```

Successfully executed the command : delete Organization

3 User Management

- Introduction to Users 102
- Managing Users through CentraSite Business UI 104
- Managing Users through CentraSite Control 115
- Managing Users through Command Line Interface 128
- Selecting Users or Groups from Repository 136

Introduction to Users

Users are individuals that are known to CentraSite. To interact with a CentraSite registry, you must create a user account for that user in the registry. A user account specifies basic attributes such as the name, email address, and phone number for an individual and any other attributes you add.

User Authentication

Although CentraSite maintains its own database of user accounts, it authenticates users externally. By default, CentraSite authenticates users against the local operating system. However, this configuration is not suitable for an enterprise-wide implementation of CentraSite. When you deploy CentraSite for actual use within your enterprise, you need to configure it to authenticate users against a production-quality authentication system such as Active Directory or an LDAP server. You must complete this configuration step before you begin creating organizations and setting up users, groups, and roles.

CentraSite allows you to define multiple user repositories for authentication, but only one is the default at any given time. Users whose user names reside in the default authentication system can log on to CentraSite with their user names. Users whose user names are not in the default authentication system must log on to CentraSite with their user names prefixed by the Domain ID that was defined for the respective authentication system.

If you are working in a distributed environment, where one or more Application Server Tiers and a separate registry or repository are involved, you must configure CentraSite to use an external authentication system. If you are working in a mixed Windows and UNIX environment, CentraSite can use Active Directory or LDAP as the user repository for both. If the CentraSite registry or repository is installed on a UNIX or Linux system, you can only use Active Directory as the user repository if it is configured using the LDAP interface.

When you configure CentraSite to use Active Directory or LDAP for user authentication, you map the user metadata (for example, name, phone number, email address) from the authentication system to the User object in CentraSite. CentraSite imports this metadata from the external directory when you create an account for a user in CentraSite.

Note:

After the information is imported, CentraSite does not attempt to keep it synchronized with the authentication system. Any change of the external user management is not synchronized with CentraSite. If a user is removed from the external user management (for instance, on the operating system level) the corresponding CentraSite user is not automatically deactivated. The CentraSite user associated with a deleted external user must be deactivated manually in CentraSite.

Predefined Users

CentraSite comes with the two predefined user accounts.

- The DefaultUser is an internal user that owns all predefined objects installed with CentraSite. The default user exists for CentraSite's internal use. You cannot edit or delete this account. You cannot use the default user account to log on to CentraSite.

- The bootstrap user is the user who installed CentraSite. This user belongs to the Default Organization and becomes the initial Organization Administrator and Primary Contact for that organization. This user is also given the CentraSite Administrator role, which gives the user super admin privileges. You can assign these roles to other users later in the deployment process. Generally, the bootstrap user creates the initial set of organizations, but other users can perform this task if the bootstrap user adds those users to CentraSite and gives them the CentraSite Administrator role.

In an organization, you must at minimum identify the users below.

- Organization Administrators who perform administrative tasks for the organization, such as:
 - Adding users to the organization.
 - Defining groups and roles in the organization.
 - Defining custom lifecycle models for the organization.
 - Creating child organizations.
 - Editing and deleting assets, policies, or lifecycle models that belong to the organization or its child organizations.

An organization must have at least one user in the Organization Administrator role. The same user can serve as Organization Administrator for multiple organizations.

- The Primary Contact for the organization. An organization has just one primary contact.

CentraSite also comes with a predefined user called the Guest user. Users with the Guest role can access the registry anonymously without a user account. By default, guests can only browse the asset catalog from CentraSite.

Login Users

Login users are defined in the user repositories that CentraSite uses for user authentication. Login users can log in to CentraSite's graphical UIs.

To create and manage users in an organization, you must belong to a role that has the Manage Users organization-specific permission. The Manage Users permission enables you to manage (create, view, edit, and delete) all users (including the groups and roles) within an organization.

Note:

Users that belong to a role that includes the Manage Organizations permission have the Manage User permission by implication.

Each user that you add within the context of a specific organization has permissions to access information within that organization, but does not have permission to access information belonging to any other organization.

Activating or Deactivating Users

An administrator can activate or deactivate a user account. Both active and inactive users exist in the registry, but only active users can log on to CentraSite, and only active users can be granted permissions and ownership of assets.

You generally deactivate users who leave the company or cease to be valid users of the registry. Inactive users retain ownership of assets they owned when they were active. They cannot be assigned to groups, and cannot be a part of the approval group. You can also create inactive users who are actors within your SOA environment but are not actual users of the registry. For example, you might model certain line-of-business managers as users in the CentraSite registry so that you can express associations between these individuals and various assets in the registry. Such users might never log on to CentraSite themselves, and do not require an account that is active and linked to the external authentication system. However, the registry will know of these users, so assets can be associated with them. Points-of-contact for external parties such as suppliers and distributors are additional individuals that you might want to model as inactive users.

Managing Users through CentraSite Business UI

This section describes operations you can perform to manage the users through CentraSite Business UI.

Adding User to an Organization

To create a new user, you must have the Manage Organizations permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

The user must be already registered in the current default user repository.

You create a new user in CentraSite Business UI using the **Add User** action in the Organization Details page.

➤ To add a user to an organization

1. In the CentraSite Business UI activity bar, click **Organizations**.

This displays a list of defined organizations in the Organizations page.

2. Click an organization to which you want to add the user.

This opens the Organization Details page. Also, the actions bar displays a set of actions that are available for working with the displayed organization.

3. On the actions bar of the Organization Details page, click **Add User**.

4. In the **Add User** dialog box, type the user ID of the user in the search field.

If your current default user repository is LDAP-based, you can instead type the value of any mapped LDAP property of the user you want to search for. You must be logged in as LDAP user to add LDAP users. For information on creating an Administrator user in a freshly installed site, refer to [“Creating an Administration User” on page 128](#).

Please see "Creating an Administration User" for information about getting the first LDAP user into CentraSite.

You can also specify just the first characters of the user ID or the LDAP property's value. In this case, CentraSite will find all users whose user ID begins with these characters, or all users who have at least one mapped LDAP property that begins with these characters.

5. Click the **Search** icon.

This displays a list of users matching the search criteria is displayed.

6. In the list of users, select one or more users you want to add to the organization, and then click **Add**.

The newly created user(s) are added to the organization.

Viewing the User List

You can view the list of users in your organization using the Search Results page (if you have the Manage Users permission) or using the Organization Details page (if you have the Manage Organizations permission) in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

In CentraSite Business UI, you can view the list of users in one of the following ways:

- Using the Typeahead Search.
- Using the Browse functionality.
- Through the Organization Details page.

> To view the list of users

- Using the Typeahead Search. In the Search box, click **Everything**, and type the name of the user in the search text box.

As you type the partial text, CentraSite returns the list of users that meet your search text.

- Using the Browse functionality. Click the **Browse** link that is located in the upper-left corner of the menu bar.

1. In the **Additional Search Criteria** list, select **Asset Types**, and then click **Choose**. This opens the **Choose Asset Types** dialog box.
2. Click the chevron next to **Everything** option button and select the **User** check box, and then click **OK**.

This displays a list of defined users in the Search Results page.

3. To filter the list to see just a subset of the available users in the Search Results page, type a partial string in the **Keyword** text box. Add the specified keyword to the Search Recipe, by clicking the plus symbol next to the text box, or press **Enter**.

The Search Results page provides the following information about each user:

Column	Description
Name	Name of the user.
Description	The description for the user.
Type	The asset type, User .
Last Updated	The date on which the user profile was last modified.
Owner	The name of the user who created or imported this user.
Organization	The organization to which the user belongs.

You can adjust the view to show or hide any of the available attributes by opening the drop-down list labeled **View** and selecting the attributes that you want to include in the view, and clearing the attributes that you do not want to view.

You can change the order in which the attributes are displayed by opening the drop-down list labeled **Sort by**. The list displays all the attributes that are selected in the **View** drop-down list. The order in which the attributes appear in the **Sort by** drop-down list is the order in which the attributes appear for each displayed user. To change the order in which any given attribute is displayed, select the attribute in the drop-down list **Sort by** and use the arrows to move the attribute to the required position.

- Through the Organization Details page. If you have the Manage Organizations permission, you can also use the **Organizations** activity to view the list of users in a particular organization. Click an organization whose list of users you want to view, and then select the **Users** profile in the Organization Details page.

This displays a list of defined users in the Organization Details page.

The Organization Details page provides the following information about each user:

Column	Description
Display Name	Name of the user.

Column	Description
Last Updated	The date on which the user profile was last modified.

In addition to the basic details, the **Users** profile includes the **Delete** action to remove a particular user from the organization.

Viewing User Details

You can view the detailed user profile information using the User Details page (if you have the Manage Users permission) or using the Organization Details page (if you have the Manage Organizations permission) in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

> To view the details of a user

1. In CentraSite Business UI, you can display the list of users in one of the following ways:

- **Using the Typeahead Search.** In the Search drop-down list, click **Everything**, and type the name of the user in the search text box.

As you type the partial text, CentraSite returns the list of users that meet your search text.

- **Using the Browse functionality.** Click the **Browse** link that is located in the upper-left corner of the menu bar.
 1. In the **Additional Search Criteria** list, select **Asset Types**, and click **Choose**. This opens the **Choose Asset Types** dialog box.
 2. Click the chevron next to **Everything** option button and select the **User** check box and click **OK**.

This displays a list of defined users in the Search Results page.

- **Through the Organization Details page.** If you have the Manage Organizations permission, you can also use the **Organizations** activity to view the list of users in an organization. Click an organization whose list of users you want to view, and then select the **Users** profile in the Organization Details page.

This displays a list of defined users in the Organization Details page.

2. In the list of users, click a user for which you want to display the details.

This opens the User Details page. Also, the actions bar displays a set of actions that are available for working with the displayed user.

You can hover over the **info** symbol next to an attribute to display the tooltip text, which describes the purpose of the attribute. The tooltip text displays the values of the attribute's Name, and Description fields contained in the User type definition.

The user details are displayed in the following profiles:

- **The Basic Information Profile:** Displays the generic attributes that include details about basic user information - Display Name, User ID, Organization, Description, Number of Consumers, Number of Consumed Assets.
- **The Assets Profile:** Displays a list of assets that are owned by the user.
- **The Groups Profile:** Displays a list of groups in that the user is a designated member. You can click each group to display the Group Details page.

In the **Groups** profile, you can add the user to a set of groups and remove the user from a set of groups.

- **The Roles Profile:** Displays a list of roles that are assigned to the user. You can click each role to display summary information about the role.

In the **Roles** profile, you can assign roles to the user and remove roles from the user.

Modifying User Details

To modify user details, you must have the Manage Organizations permission or at least the Manage Users permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

You can perform the user modification tasks in the User Details page. The modification is broken up across the different profiles in the User Details page, which means that modifications done in each profile are independent of each other and must be saved individually.

If you are a user with the Organization Administrator role, you can change the values of the following predefined attributes - **User Name** and **User Email Address**.

If you are a user with the CentraSite Administrator role, you can additionally change the value of the predefined attribute - **Owning Organization**.

> To modify the basic details of a user

1. In CentraSite Business UI, you can display the list of users in one of the following ways:

- **Using the Typeahead Search.** In the Search box, click **Everything**, and type the name of the user in the search text box.

As you type the partial text, CentraSite returns the list of users that meet your search text.

- **Using the Browse functionality.** Click the **Browse** link that is located in the upper-left corner of the menu bar.
 1. In the **Additional Search Criteria** list, select **Asset Types**, and click **Choose**. This opens the **Choose Asset Types** dialog box.
 2. Click the chevron next to **Everything** option button and select the **User** check box and click **OK**.

This displays a list of defined users in the Search Results page.

- **Through the Organization Details page.** If you have the Manage Organizations permission, you can also use the **Organizations** activity to view the list of users in an organization. Click an organization whose list of users you want to view, and then select the **Users** profile in the Organization Details page.

This displays a list of defined users in the Organization Details page.

2. Click the user whose details you want to modify.

This opens the User Details page. Also, the actions bar displays a set of actions that are available for working with the displayed user.

3. To modify the user's details displayed in the **Basic Information** profile, click **Edit**.
4. Modify the values for the user's fields in the User Details page as required.
5. Click **Save** to update the user information.

Adding User to a Group Through User Details Page

To add a user to an existing group, you must have the Manage Organizations permission or at least the Manage Users permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

You can add a user to a group in CentraSite Business UI in the following ways:

- From the User Details page
- From the Group Details page

Instructions for adding a user from the Group Details page is provided in [“Adding User to a Group Through Group Details Page”](#) on page 145.

➤ To add a user to a group from the User Details page

1. In CentraSite Business UI, you can display the list of users in one of the following ways:

- **Using the Typeahead Search.** In the Search box, click **Everything**, and type the name of the user in the search text box.

As you enter the partial text, CentraSite returns the list of users that meet your search text.

- **Using the Browse functionality.** Click the **Browse** link that is located in the upper-left corner of the menu bar.
 1. In the **Additional Search Criteria** list, select **Asset Types**, and click **Choose**. This opens the **Choose Asset Types** dialog box.
 2. Click the chevron next to **Everything** option button and select the **User** check box and click **OK**.

This displays a list of defined users in the Search Results page.

- **Through the Organization Details page.** If you have the Manage Organizations permission, you can also use the **Organizations** activity to view the list of users in an organization. Click an organization whose list of users you want to view, and then select the **Users** profile in the Organization Details page.

This displays a list of defined users in the Organization Details page.

2. Click the user you want to add to a group.

This opens the User Details page. Also, the actions bar displays a set of actions that are available for working with the displayed user.

3. On the actions bar of the User Details page, click **Add to Group**.
4. To see a list of all available groups, click the **Search** icon.

You can also type the first few characters of the group name in the search field, then click the **Search** icon. This will display all groups whose name starts with the given characters. You can use wildcard characters (* or %) in the search field. You can sort the groups based on the generic attributes - Name, Description, and Organization, by using the **Sort By** list. You can also configure the group attributes that you want to view by using the **View** list.

5. In the list of groups, select one or more groups to which you want to add the user.
6. Click **Add**.

This adds the user to the selected group(s).

Deleting User from a Group Through User Details Page

To delete a user from a group, you must have the Manage Organizations permission or at least the Manage Users permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

You can delete a user from a group using CentraSite Business UI in the following ways:

- From the User Details page
- From the Group Details page

Instructions for deleting a user from the Group Details page is provided in [“Deleting User from a Group Through Group Details Page” on page 146](#).

The following general guidelines apply when deleting a user from a group:

- When you delete a user from a group, you delete the association between the user and the group, but both the user and the group continue to exist.
- You cannot remove a user from predefined groups such as `Users` or `Members`.

➤ **To delete a user from a group from the User Details page**

1. In CentraSite Business UI, you can display the list of users in one of the following ways:

- **Using the Typeahead Search.** In the Search box, click **Everything**, and type the name of the user in the search text box.

As you type the partial text, CentraSite returns the list of users that meet your search text.

- **Using the Browse functionality.** Click the **Browse** link that is located in the upper-left corner of the menu bar.
 1. In the **Additional Search Criteria** list, select **Asset Types**, and click **Choose**. This opens the **Choose Asset Types** dialog box.
 2. Click the chevron next to **Everything** option button and select the **User** check box and click **OK**.

A list of defined users for your organization is displayed in the Search Results page.

- **Through the Organization Details page.** If you have the Manage Organizations permission, you can also use the **Organizations** activity to view the list of users in an organization. Click an organization whose list of users you want to view, and select the **Users** profile in the Organization Details page.

A list of defined users for that particular organization is displayed.

2. Click the user you want to remove from a group.
3. In the User Details page, click the **Groups** profile.
4. Hover over the group from which you want to remove the user.

This displays icons for one or more actions that you can perform on the group.

5. Click **Delete**.
6. Click **Yes** in the confirmation dialog box.

The user is removed from the group.

Viewing Assets Owned by a User

To view assets owned by a user, you must have the Manage Organizations permission or at least the Manage Users permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

> **To view the assets owned by a user**

1. In CentraSite Business UI, you can display the list of users in one of the following ways:
 - **Using the Typeahead Search.** In the Search drop-down list, click **Everything**, and type the name of the user in the search text box.

As you type the partial text, CentraSite returns the list of users that meet your search text.
 - **Using the Browse functionality.** Click the **Browse** link that is located in the upper-left corner of the menu bar.
 1. In the **Additional Search Criteria** list, select **Asset Types**, and click **Choose**. This opens the **Choose Asset Types** dialog box.
 2. Click the chevron next to **Everything** option button and select the **User** check box and click **OK**.

A list of defined users for your organization is displayed in the Search Results page.
 - **Through the Organization Details page.** If you have the Manage Organizations permission, you can also use the **Organizations** activity to view the list of users in an organization. Click an organization whose list of users you want to view, and then select the **Users** profile in the Organization Details page.

A list of defined users for that particular organization is displayed.
2. Click a user whose assets you want to view.
3. In the User Details page, click the **Assets** profile.

This displays the assets owned by the particular user.

Synchronizing LDAP Users

Pre-requisites:

To synchronize a LDAP user, you must have the Manage Organizations permission in CentraSite.

You synchronize LDAP user IDs imported to the CentraSite registry with the users and user groups in an external authentication system, for example, LDAP directory server. Synchronization simplifies the maintenance by eliminating the need to update two systems when the user information changes in the external authentication system.

You might consider synchronizing a LDAP user in CentraSite if you want to update any changes to the user, for example, modification to the user name, that is performed in the LDAP directory.

When CentraSite executes the synchronization, it accesses the external authentication system to update the user information. It performs the synchronization for each user who is a member of an imported LDAP group and is also a registered user on CentraSite.

The following limitations apply when synchronizing user information to CentraSite from LDAP directory:

- CentraSite allows one-way synchronization from the LDAP directory. If you change user information on the CentraSite registry, the changes are not synchronized back to the configured LDAP directory.
- Users imported to CentraSite through LDAP are always authenticated in CentraSite through the configured LDAP directory. If the LDAP directory is unavailable for any reason, the LDAP imported users cannot log in to CentraSite.

You can synchronize LDAP users in CentraSite in the following ways:

- Through CentraSite Business UI: You can synchronize an individual user with the LDAP directory.
- Through Command Line Interface: You can synchronize a set of users through group synchronization with the LDAP directory.

CentraSite provides a command tool named `sync Ldap Group` for this purpose. Instructions for synchronizing LDAP users through the LDAP group synchronization is provided in [“Synchronizing LDAP Groups” on page 157](#).

➤ To synchronize LDAP users

1. In the CentraSite Business UI activity bar, click **Organizations**.

This displays a list of defined organizations in the Organizations page.

2. Click an organization to which the user belongs.
3. In the Organization Details page, click the **Users** profile.

4. Click the LDAP user you want to synchronize the user information from the LDAP directory.

This opens the User Details page. Also, the actions bar displays a set of actions that are available for working with the displayed user.

5. Click **Synchronize**.

CentraSite accesses the LDAP directory and updates the user information in the registry.

Deleting Users

To delete a user, you must have the Manage Organizations permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

You might consider deleting a user in CentraSite if you want to:

- Suspend access to the CentraSite registry objects (either on a temporary or permanent basis).
- Delete the user permanently from the CentraSite registry.

The following general guidelines apply when deleting users in CentraSite:

- You cannot delete a user who is the only administrator or the primary contact of an organization or owns assets in CentraSite.
- Deleting a user permanently removes a user from the CentraSite registry or repository but does not delete the user from the external authentication system.
- Make sure at least one active user with the CentraSite Administrator role always resides in the Default Organization. Even if you plan to switch CentraSite's user authentication from one domain to another (such as, from the operating system to an Active Directory or LDAP domain), to prevent a system lockout, make sure you have at least one user in the CentraSite Administrator role.

> To delete users

1. In the CentraSite Business UI activity bar, click **Organizations**.

This displays a list of defined organizations in the Organizations page.

2. Click an organization to which the user belongs.
3. In the Organization Details page, click the **Users** profile.
4. In the list of users, hover over the user you want to delete.

This displays icons for one or more actions that you can perform on the user.

5. Click **Delete**.
6. Click **Yes** in the confirmation dialog box.

The user is permanently removed from the CentraSite registry.

Managing Users through CentraSite Control

This section describes operations you can perform to manage the users through CentraSite Control.

Adding User to an Organization

To create a new user, you must have the Manage Organizations permission or at least the Manage Users permission for an organization in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

Important:

The user must be already registered in the current default user repository.

You can create a new user in CentraSite Control in the following ways:

- Create a user and associate with an external authentication system account
- Bulk load users from the external authentication system
- Create users from an external authentication system from an organization's **Users** tab

Adding a User and Associating with External Authentication System Account

➤ **To add a user and associate with an external authentication system**

1. In CentraSite Control, go to **Administration > Users > Users**.
2. In the **Users** page, click **Add User**.
3. In the **Organization** field, click the organization to which you want to add the user.
4. Click **Associate** and select a user in the external authentication system.

For more information, see [“Selecting Users or Groups from Repository” on page 136](#).

Note:

You can only select users that are stored in the same repository as the user who is logged into CentraSite Control and is performing the current operation. For example, if your system

has both internal users and LDAP users, an internal user cannot search for users that are stored in the LDAP repository.

5. In the area labeled **User Information**, provide the required information for each of the displayed data fields.

Field	Description
First Name	First name of the user.
Middle Name	<i>Optional.</i> Middle name of the user.
Last Name	Last name of the user.
E-mail Address	<i>Optional.</i> E-mail address of the user. If you provide the user's email address, CentraSite can notify the user of certain events.

6. To specify additional information, click the **Address Information** tab, and provide the required information for the displayed data fields.

In this field...	Do the following...
Address	<i>Optional.</i> The postal address of the user.
Contact	<i>Optional.</i> The phone number and fax number of the user.

7. To specify custom properties (key-value pairs) for the user, click the **Object-Specific Properties** tab and provide the required information:

- a. Click **Add Property**.
- b. In the **Add Property** dialog box, provide a name and values for the property in the respective data fields as required.

The name can include letters, numbers, and underscores (_), but cannot include spaces or other special characters. You can supply a namespace for the property.

- c. Click **OK**.

8. Click the **Attributes** tab, and provide values for the attributes in the respective data fields as required.

Attributes marked with an asterisk (*) are required.

The **Attributes** tab is visible only if an administrator has added custom attributes to the User type definition.

9. To add the user to a group, click the **Groups** tab and perform the following:
 - a. Click **Add Group**.
 - b. In the **Add Group** dialog box, select the check box for one group, or select the check boxes for multiple groups to which you want to add the user.
 - c. To filter the list, type a partial string in the **Search** field.
CentraSite applies the filter to the **Name** column.
 - d. Click **OK**.
10. To assign roles to the user, click the **Roles** tab and perform the following:
 - a. Click **Add Role**.
 - b. In the **Add Role** dialog box, select the check box for one role, or select the check boxes for multiple roles you want to give the user.
 - c. To filter the list, type a partial string in the **Search** field.
CentraSite applies the filter to the **Name** column.
 - d. Click **OK**.
11. Click **Save**.

Bulk Loading Users from External Authentication System

» To bulk load users from an external authentication system

1. In CentraSite Control, go to **Administration > Users > Users**.
2. Click **Bulk Load Users from External Source**.
3. Select one or more users to add to the organization.
For more information, see [“Selecting Users or Groups from Repository” on page 136](#).
4. In the **Import to Organization** field, select the organization to which you want to add the users.
5. Scroll through the user list to confirm that the selected users are added successfully.

6. Examine each new user that you added to the specified organization and update the user's attributes as necessary. (If you selected users from an Active Directory or LDAP system, many of the new users' attributes will already be populated.)

Adding Users from External Authentication System

You must have the Manage Organizations permission on the organization to which you want to add users.

➤ To add users from an external authentication system

1. In CentraSite Control, go to **Administration > Organizations**.
2. Right-click an organization to which you want to add users, and then click **Details**.

This opens the Edit Organization page.
3. To add users, click the **Users** tab and perform the following:
 - a. Click **Add Users**.
 - b. In the **Add Users** dialog box, select one or more users you want to add to the organization.

For more information, see [“Selecting Users or Groups from Repository”](#) on page 136.
 - c. Click **OK**.
 - d. Scroll through the user list to confirm that the selected users are added successfully.
4. Click **Save**.
5. Examine each new user that you added to the specified organization and update the user profile information as necessary. (If you selected users from an Active Directory or LDAP system, many of the new users' attributes will already be populated.)

Viewing the User List

To view the user list, you must have the Manage Organizations permission or at least the Manage Users permission for an organization in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

In CentraSite Control, you can view the list of users in one of the following ways:

- Through the Users page.

- Through the Edit Organization page.

➤ **To view the list of users**

- Through the Users page. Go to **Administration > Users > Users** to display a list of users currently defined in CentraSite. To filter the list to see just a subset of the available users, type a partial string in the **Search** field.

CentraSite applies the filter to the **Name** column. The **Search** field is a type-ahead field, so as soon as you type any characters, the display is updated to show only those users whose name contains the specified characters. The wildcard character % is supported.

If you type...	CentraSite displays
b	Names that contain b
bar	Names that contain bar
%	All names

The Users page provides the following information about each user:

Column	Description
Name	Name of the user.
User ID	Log in ID of the user.
Organization	Name of the organization to which the user belongs.
Can Log On	Status of the user.

Icon	Description
	The user is active (can log in to CentraSite).
	The user is inactive (cannot log in to CentraSite).

You can adjust the view to show or hide the individual columns by using the **Select Columns** icon that is located in the upper-right corner of the Users page.

The shortcut menu of a particular user displays one or more actions that you can perform on that user.

Action	Description
Details	Displays the details page of the user.
Delete	Deletes the user.
Move	Transfers the user from current organization to another organization.

Action	Description
Activate	Activates an inactive user.
Deactivate	Deactivates an active user.
Impact Analysis	Helps to easily visualize the associations that exist between the user and registry objects.
Add to List	Adds the user to a list in My Favorites .

- Through the Edit Organization page. Go to **Administration > Organizations**.

1. Right-click an organization whose users you want to view, and then click **Details**.

This opens the Edit Organization page.

2. Click the **Users** profile.

This displays a list of users defined in the organization.

Viewing User Details

To view the user details, you must have the Manage Organizations permission or at least the Manage Users permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

> To view the details of a user

1. In CentraSite Control, go to **Administration > Users > Users**.
2. Right-click a user for which you want to display the details, and click **Details**.

This opens the Edit User page. The area labeled **User Information** displays the generic attributes that includes details about basic user information - First Name, Middle Name, Last Name, E-mail Address, Organization, Associated External User, Login Status.

The user details are displayed in the following tabs:

- **The Contact Information Tab:** Displays the postal address and contact numbers of the user.
- **The Groups Tab:** Displays a list of groups in that the user is a designated member.

In the **Groups** tab, you can add the user to a set of groups and remove the user from a set of groups.

- **The Roles Tab:** Displays a list of roles that are assigned to the user. You can click each role to display summary information about the role.

In the **Roles** tab, you can assign roles to the user and remove roles from the user.

- **The Assets Tab:** Displays a list of assets that are owned by the user.
- **The Object-Specific Properties Tab:** Displays a list of custom properties (key-value pairs) for the user.

In the **Object-Specific Properties** tab, you can add custom properties to the user and remove custom properties from the user.

- **The Attributes Tab:** Displays a list of custom attributes for the user.

In the **Attributes** tab, you can specify values for the custom attributes if an administrator has added the custom attributes to the User type definition.

Modifying User Details

To modify user details, you must have the Manage Organizations permission or at least the Manage Users permission for an organization in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

You can perform the user modification tasks in the Edit User page. The modification is broken up across the different tabs in the Edit User page, which means that modifications done in each tab are independent of each other and must be saved individually. The modifications you can perform in each tab is outlined in the subsequent sections.

➤ To modify the basic details of a user

1. In CentraSite Control, go to **Administration > Users > Users**.
2. Right-click a user whose details you want to modify, and click **Details**.

This opens the Edit User page.

3. In the area labeled **User Information**, modify the values for the user's fields as required.

If you belong to the CentraSite Administrator role, you can move a user to a new organization without moving the user's asset by changing the value of the **Organization** field.

4. After you have made the required changes, click **Save** to update the user information.

Adding User to a Locally Managed Group

To add a user to a group, you must have the Manage Organizations permission or at least the Manage Users permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

Important:

To add a user to a group, the following conditions must be satisfied:

- The CentraSite user must be registered in the current default user repository.
- The CentraSite user must be active.

If these conditions are not satisfied, the **Add User to Group** button will be disabled in the Edit User page.

> To add a user to a group

1. In CentraSite Control, go to **Administration > Users > Users**.
2. Right-click a user whose group assignments you want to modify, and then click **Details**.
This opens the Edit User.
3. To add the user to one or more groups, click the **Groups** tab and perform the following:
 - a. Click the **Add User to Group** button.
 - b. In the **Add User to Group** dialog box, select one or more roles to be assigned to the user.
 - c. Click **OK**.

Assigning Roles to a User

To assign roles to a user, you must have the Manage Organizations permission or at least the Manage Users permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

Important:

To assign roles to a user, the following conditions must be satisfied:

- The CentraSite user must be registered in the current default user repository.
- The CentraSite user must be active.

If these conditions are not satisfied, the **Assign Role** button will be disabled in the Edit User page.

➤ **To assign roles to a user**

1. In CentraSite Control, go to **Administration > Users > Users**.
2. Right-click a user whose role assignments you want to modify, and click **Details**.
The Edit User page is displayed.
3. To assign roles to the user, click the **Roles** tab and perform the following:
 - a. Click **Assign Role**.
 - b. In the **Assign Role** dialog box, select one or more roles to be assigned to the user.
 - c. Click **OK**.

Activating or Deactivating Users

You can activate or deactivate a user in CentraSite by using the **Activate** or **Deactivate** action in the Users page or the Edit User page (if you have the Manage Users permission) or the Edit Organization page (if you have the Manage Organizations permission) in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

Activating a user allows the user to log in to CentraSite. Deactivating a user prevents the user from logging in to CentraSite. You would typically do this when a user leaves your organization. You might have to deactivate those users after transferring ownership of the assets to other users.

You can activate or deactivate a user in CentraSite from the:

- Users Page
- Edit User Page
- Edit Organization Page

The following general guidelines apply when activating or deactivating a user:

- If there is only one user in the CentraSite Administrator role, or only one user in the Organization Administrator role within an organization, you cannot deactivate that user. You cannot deactivate a user who is an authorized approver for an approval flow that is in the Pending state.

- Inactive users cannot be assigned to groups. If you deactivate a user, the user does not be able to receive automatic email notifications from CentraSite. If you deactivate a user who was part of an approval group or a user who is the only member of an approval group, the policy with that particular approval group is marked as fail.
- To keep your audit trail intact when a user leaves the registry, deactivate that user but leave his or her existing assets in place. If you delete the user or transfer the user's assets to someone else, the audit trail for those assets is lost.

Activating or Deactivating Users from the Users Page

> To activate or deactivate users from the Users page

1. In CentraSite Control, go to **Administration > Users > Users**.
2. Right-click a user you want to activate or deactivate, and click **Activate** or **Deactivate**.
You can also or select multiple users, click the **Actions** menu, and click **Activate** or **Deactivate**.
3. Make sure the user's state has changed by checking the icon in the **Can log on** column.

Icon	Description
	The user is active (can log in to CentraSite Control).
	The user is inactive (cannot log in to CentraSite Control).

Activating or Deactivating a User from the Edit User Page

> To activate or deactivate a user from the Edit User page

1. In CentraSite Control, go to **Administration > Users > Users**.
A list of defined users is displayed in the Users page.
2. Right-click a user you want to activate or deactivate, and click **Activate** or **Deactivate**.

Activating or Deactivating a Users from the Edit Organization Page

> To activate or deactivate users from the Edit Organization page

1. In CentraSite Control, go to **Administration > Organizations**.
2. Right-click an organization to which the user belongs, and click **Details**.

The Edit Organization page is displayed.

3. Click the **Users** profile.

A list of users who belong to the organization is displayed.

4. Right-click a user you want to activate or deactivate, and click **Activate** or **Deactivate**.

You can also select multiple users, click the **Actions** menu, and click **Activate** or **Deactivate**.

Viewing Assets Owned By a User

To view the list of assets owned by a user, you must have the Manage Organizations permission or at least the Manage Users permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

➤ To view the assets owned by a user

1. In CentraSite Control, go to **Administration > Users > Users**.
2. Right-click a user whose assets you want to view, and click **Details**.

The Edit User page is displayed.

3. In the Edit User page, click the **Assets** profile.

This displays the assets owned by the particular user.

Moving Users to a Different Organization

To move a user, you must have the Manage Organizations permission or at least the Manage Users permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

The following general guidelines apply when moving users to a different organization:

- If a user transfers to a department or work group in another organization within your enterprise, you can mirror that change in CentraSite. When you move a user to another organization, you can also move the user's assets to the target organization or you can leave them with their current organization.
- CentraSite treats the move operation as an update to the User object. When you move a user to another organization, CentraSite does the following:

- Removes the user from the system groups in the user's old organization and adds the user to the Users and Members system groups in the target organization. The user retains all other group memberships.
- Triggers pre- and post-update policies of the target organization on the User object. If a pre-update policy fails, the user is not moved into the target organization.
- Optionally, transfers the assets owned by the user to the target organization. If you do not select this option, the user's existing assets remain in the organization to which they are currently assigned, and the user continues to serve as their owner.
- Sends a notification to the inbox of the user when the move is complete.
- Records the user's organization change in the audit log.
- Members of the Users group for an organization have implicit View permission on the organization's assets. Because CentraSite transfers users from one Users group to another during a move, moved users lose implicit access to the assets in their former organization (except for the assets that they own) and receive implicit access to the assets in the target organization. If users require continued access to the assets in their former organization, consider granting the Asset Consumer role (in the old organization) to them after the move.
- If there are any explicit instance-level or role-based permissions assigned to the Users and Members groups in their former organization, moved users will lose those permissions. Moving users to another organization does not affect any instance-level permissions or role-based permissions that are granted directly to their user accounts or to any non-system groups to which they belong. Therefore, other than losing access to certain assets as a result of leaving the Users and Members groups in their former organization, users continue to have access to the same set of assets as they had before the move.
- You cannot move the default user or any other internal user that is installed by CentraSite. You can transfer active or inactive users.

> To move a user to a different organization

1. In CentraSite Control, go to **Administration > Users > Users**.

2. Right-click a user you want to move, and click **Move**.

You can also select multiple users, click the **Actions** menu, and then click **Move**.

3. Select an organization to which you want to move the users.

You can filter the organization list by typing a partial string in the search field.

4. Select the **Move Assets owned by the selected user(s) to the new organization** option button to transfer the assets owned by the selected user.

Deleting Users

You can delete a user in CentraSite Control by using the Users page (if you have the Manage Users permission) or the Edit Organization page (if you have the Manage Organizations permission) in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

You might consider deleting a user in CentraSite if you want to:

- Suspend access to the CentraSite registry objects (either on a temporary or permanent basis).
- Delete the user permanently from the CentraSite registry.

The following general guidelines apply when deleting users in CentraSite:

- You cannot delete a user who is the only administrator or the primary contact of an organization or owns assets in CentraSite.
- Deleting a user permanently removes a user from the CentraSite registry or repository but does not delete the user from the external authentication system.
- Make sure at least one active user with the CentraSite Administrator role always resides in the Default Organization. Even if you plan to switch CentraSite's user authentication from one domain to another (such as, from the operating system to an Active Directory or LDAP domain), to prevent a system lockout, make sure you have at least one user in the CentraSite Administrator role.

Deleting Users from the Users Page

➤ **To delete a user from the Users page**

1. In CentraSite Control, go to **Administration > Users > Users**.

2. Right-click a user you want to delete, and click **Delete**.

You can also select multiple users, click the **Actions** menu, and then click **Delete**.

3. Click the **Delete** icon.

4. Click **OK** in the confirmation dialog box.

Each selected user is permanently removed from the CentraSite registry or repository. If the user had an associated user account in the external authentication system, that account is not affected.

Deleting Users from the Edit Organization Page

➤ **To delete users from the Edit Organization page**

1. In CentraSite Control, go to **Administration > Organizations**.

2. Right-click an organization to which the user belongs, and click **Details**.

This opens the Edit Organization page.

3. Click the **Users** profile.

This displays a list of users who belong to the organization.

4. Right-click a user you want to delete, and click **Delete**.

You can also select multiple users, click the **Actions** menu, and click **Delete**.

5. Click **OK** in the confirmation dialog box.

Each selected user is removed from the organization.

Managing Users through Command Line Interface

This section describes operations you can perform to manage the users through the CentraSite Command Line Interface (CLI).

Creating an Administration User

Pre-requisites:

To create an administration user through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

The user domain of the default authentication configuration must contain at least one user who is defined in CentraSite with the CentraSite Administrator role. Under certain circumstances, like the ones mentioned below there may be not be any administrator available in the user repository:

- the CentraSite is freshly installed and the initial administrator must be configured.
- the user repository is currently not available (for example, LDAP Server currently unavailable).
- all the users who have the CentraSite Administrator role in CentraSite have been deleted from the user repository.

In such cases, there are no users any more who can log in to CentraSite as a user with the CentraSite Administrator role, so no CentraSite administration tasks can be performed.

To resolve this problem, a mechanism is available to create a user in the user repository, assigned to the CentraSite Administrator role in CentraSite.

CentraSite provides a command tool named `add Admin` for this purpose. The tool automatically grants the new external user the same permissions that were granted for the old external user.

> To create an administration user

- Run the command `add Admin`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd add Admin [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -domain <DOMAIN> -domainUser <DOMAIN-USER-ID> -domainPassword <DOMAIN-PASSWORD> -organization <ORGANIZATION>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
DOMAIN	The domain name of the user repository associated with the configuration.
DOMAIN-USER-ID	The user name for a new user to be created in the user repository. A user with this name is created in CentraSite and have the role CentraSite Administrator.
DOMAIN-PASSWORD	The user repository password for the user specified in <code>DOMAIN-USER-ID</code> .
ORGANIZATION	The organization to which the newly created CentraSite user will belong.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd add Admin -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-domain LDAPDomain -domainUser AdminUser -domainPassword AdminPass -organization
MyOrganization
```

The response to this command could be:

```
Executing the command : add Admin
Successfully executed the command : add Admin
```

Changing Password of Predefined User

Pre-requisites:

To change the password of a predefined user through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

In certain circumstances, you may want to change the password of any of the predefined user in CentraSite.

User	Description	Predefined Password
DefaultUser	Owns all predefined objects. Change the password for this user as soon as possible after you install CentraSite.	PwdFor_CS21
guest	Configured to have view access to only some resources.	guest
UDDIsubscriptionUser	Used for communication between the application server and the CentraSite UDDI server.	UDDI4CentraSite
PurgeUser	Used to purge log records.	LogPurger4CS
EventsUser	Used by CentraSite Events Listener for authentication before persisting event data to the RuntimeEvents Collection database. You can change the password for this user or you can change the EventsUser to a login user by configuring the Event Receiver.	EventsManager4CS

CentraSite provides a command tool named `set Password` for this purpose.

Note:

If you change the password for a user, you must update the CentraSite registry because multiple Application Server Tiers can access a single registry and any password change that occurs on one Application Server Tier must be made known to the other Application Server Tiers.

Note:

After setting the password, you must restart Software AG Runtime for the changes to take effect.

> To change the password of a predefined user from the Command Line

- Run the command `set Password`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Password [-url <CENTRASITE-URL> -user <USER-ID> -password <PASSWORD> -predefinedUser <PREDEFINED-USER> -newPassword <NEW-PASSWORD>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
PREDEFINED-USER	A predefined user, for example, <code>DefaultUser</code> or <code>EventsUser</code> .
NEW-PASSWORD	The password for the CentraSite user identified by the <code>PREDEFINED-USER</code> parameter.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Password -url
http://localhost:53307/Centrasite/Centrasite -user AdminUser -password ABCXYZ123
-predefinedUser DefaultUser -newPassword MyPassword2
```

Changing Password of Login User in Password Store

Pre-requisites:

To change the password of a predefined user through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a secure password store for managing the passwords of login users whose credentials are required for internal communication between CentraSite components. Examples are the use of policies such as Promote Asset and Initiate Approval, which policies cause a lifecycle model state change that requires the approval of an authorized login user.

This password store exists in parallel to the user repository that CentraSite uses for authentication of users. There is no automatic synchronization of passwords between the user repository and the password store. The password for a given login user in the password store must be the same as the password for the same login user in the user repository. If you change a password in the user repository, you must manually update the password in the password store to the same new password.

CentraSite provides a command tool named `set Password` for this purpose.

Note:

If you change the password for a user, you must update the CentraSite registry because multiple Application Server Tiers can access a single registry and any password change that occurs on one Application Server Tier must be made known to the other Application Server Tiers.

Note:

After setting the password, you must restart Software AG Runtime for the changes to take effect.

> To change the password of a login user from the Command Line

- Run the command `set Password`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Password [-url <CENTRASITE-URL> -user <USER-ID> -password <PASSWORD> -userToStore <USER-TO-STORE> -passwordToStore <PASSWORD-TO-STORE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
USER-TO-STORE	The login user for which the password is stored.
PASSWORD-TO-STORE	The password for the CentraSite user identified by the <code>USER-TO-STORE</code> parameter.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Password -url
http://localhost:53307/CentraSite/CentraSite -user AdminUser -password ABCXYZ123
-userToStore LOGIN\user -passwordToStore MyPassword2
```

Reassociating Users

Pre-requisites:

To reassociate a user through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

Before you run the tool, we strongly recommend that you create a database backup.

In addition to the database backup, make sure the following conditions are satisfied:

- There is a unique registry object for the old external user.
- The old external user can be uniquely identified in the security configuration.
- There is no registry object for the new external user.
- There is no security configuration for the new external user.
- The domain of the new external user exists in the security configuration.

- A GUI configuration does not exist for the new external user.

You can change the association of a CentraSite user from one external user to another. This can be necessary, for example, if the responsibility for certain CentraSite assets moves from one person to another person in the same authentication domain. By reassociating the user, you can keep the name of the CentraSite user unchanged. Another possible use would be to handle user IDs when the default domain name changes; for example, when switching from operating system authentication to LDAP authentication.

CentraSite provides a Java tool named `ReassociateUsers.jar` for this purpose.

The tool automatically grants the new external user the same permissions that were granted for the old external user.

- Run the Java tool `ReassociateUsers.jar`.

- Given the user ID:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd ReassociateUsers.jar
<CentraSite URL> <admin user id> <password> users <old user id> <new user id>
```

- Given the domain ID:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd ReassociateUsers.jar
<CentraSite URL> <admin user id> <password> domains <old domain id> <new domain
id>
```

- Given the user mapping details:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd ReassociateUsers.jar
<CentraSite URL> <admin user id> <password> file <user mapping file path>
```

The input parameters are:

Parameter	Description
CentraSite URL	The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
admin user id	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
password	The password for the CentraSite user identified by the parameter <code>admin user id</code> .
old user id	The domain ID of the old external user.
new user id	The domain ID of the new external user.
user mapping file path	The absolute or relative path to the user mapping file. If relative, the path should be relative to the location from where the command is executed.

Parameter	Description
	<p>The user mapping file contains a map of key-value pairs, and defines the user objects as simply lines of comma separated old user and new user pairs (for example, "OLDDOMAIN\oldUser, NEWDOMAIN\newUser").</p> <div style="background-color: #f0f0f0; padding: 5px;"><p>Note: The same user ID must not be specified more than once in the mapping file.</p></div>
	<p>The reassociation may take some time. The tool's progress is reported to standard output.</p>
	<p>Examples (all in one line):</p> <pre>C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd ReassociateUsers.jar http://localhost:53307/CentraSite/CentraSite DOMAIN\admin pAsSw0rD users OLDDOMAIN\oldUser NEWDOMAIN\newUser C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd ReassociateUsers.jar http://localhost:53307/CentraSite/CentraSite DOMAIN\admin pAsSw0rD domains OLDDOMAIN NEWDOMAIN C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd ReassociateUsers.jar http://localhost:53307/CentraSite/CentraSite DOMAIN\admin pAsSw0rD file users.txt</pre>

Synchronizing LDAP Users

Pre-requisites:

To synchronize a LDAP user through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

Before you run the tool, Software AG recommends that you create a database backup.

You might consider synchronizing a LDAP user in CentraSite to update any changes to the user, for example, modification to user name, that is performed in the LDAP directory. You can synchronize a set of users through group synchronization with the LDAP directory.

CentraSite provides a command tool named `sync Ldap Group` for this purpose.

Instructions for synchronizing LDAP users through the LDAP group synchronization is provided in [“Synchronizing LDAP Groups” on page 157](#).

Deleting Users

Pre-requisites:

To delete a user from the through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

Before you run the tool, deactivate the user and activate the target user for transferring the ownership of referenced objects.

In some circumstances, a user object cannot be deleted because internal objects that reference it cannot be deleted. This can happen when there are internal references to a user object even though the user is no longer the owner of any assets. There can also be references to the user object in the audit log.

CentraSite provides a Java tool named `CentraSiteDeleteUser.jar` for this purpose. The tool does the following:

- Transfers ownership of objects, access rights, and group memberships to the target user.
- Assigns the role to the target user if the user was the primary contact of an organization.
- Redirects internal references to the target user.
- Creates an `OWNERSHIPTRANSFERRED` event in the audit logs for every registry object that references the user object.
- Removes the GUI configuration and the user object.

Important:

The operation to delete a user requires several steps that cannot run within a single transaction. This means every parallel running transaction is able to see intermediate results. Make sure no other activity is in progress while you run the tool. Moreover if there is a failure of any of the steps during the execution, the registry has an intermediate state. The original state cannot be recovered by rolling back the complete operation.

➤ **To delete user**

- Run the Java tool `CentraSiteDeleteUser.jar`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd CentraSiteDeleteUser.jar <CentraSite URL> <admin user id> <password> <id or key of user to be deleted> <id of target user>`

The input parameters are:

Parameter	Description
CentraSite URL	The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
admin user id	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
password	The password for the CentraSite user identified by the parameter <code>admin user id</code> .
id or key of user to be deleted	The user ID or UDDI key of an existing CentraSite user you want to remove from the registry.

Parameter	Description
id of target user	The user ID of the new user you want to add to the CentraSite registry.

Examples (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd CentraSiteDeleteUser.jar  
http://localhost:53307/CentraSite/CentraSite DOMAIN\admin pAsSw0rD DOMAIN\oldUser  
DOMAIN\newUser
```

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd CentraSiteDeleteUser.jar  
http://localhost:53307/CentraSite/CentraSite DOMAIN\admin pAsSw0rD  
uddi:1e5aff10-f3e3-11df-86fc-a6e2fa0ea483 DOMAIN\newUser
```

Selecting Users or Groups from Repository

Select Users or Groups from Operating System User Repository

The following general guidelines apply when searching for users or groups in the operating system user repository:

- When searching for users, CentraSite searches the user ID attribute, not the user name attribute. CentraSite automatically filters out users that have already been added to CentraSite.
- To list all users or groups, type % or * as the search string. You cannot combine either wildcard with other characters. For example, the search strings ab% and %ab are not valid.
- To find specific users or groups, type a search string that specifies the characters with which the user ID or group name begins. The domain portion of the name is not included in the search. For example, if a user has the user ID MyDomain\AdminUser01, a search for Ad will find the user, whereas a search for User01 or My will not.
- Search strings are not case or accent sensitive.
- If the user is not known to the local system but is known to a domain server to which the local operating system is connected, type the user's domain-qualified name into the **Type Domain Name field**.

Note:

If you type a user ID in the **Type Domain Name field**, CentraSite ignores any selections you have made in the user list.

Select Users or Groups from Active Directory or LDAP Repository

The following general guidelines apply when searching for users or groups in an Active Directory or LDAP repository:

- The Active Directory or LDAP authentication system performs a user search based on the attribute mapping specified in the authentication configuration, and displays the users that fit the search criteria.
- In the **Search Criteria** panel, create the search criteria by selecting the attribute and the condition from the list boxes, typing the search string in the text box and selecting **Equals** or **NotEquals**.
- CentraSite treats the search string as a partial string. For example, if you enter a1, then Alex, Allen, and Sally all fit the search criteria.
- You can use an asterisk (*) as a wildcard in the search string. CentraSite replaces the wildcard symbol with as many characters as necessary.
- Search strings are not case or accent sensitive.
- To add more search conditions, click the plus button. If all conditions must be met, select **AND Condition**. If only one condition must be met, select **OR Condition**.

4 Group Management

■ Introduction to Groups	140
■ Managing Groups through CentraSite Business UI	142
■ Managing Groups through CentraSite Control	147
■ Managing Groups through Command Line Interface	155
■ Selecting Users or Groups from Repository	158

Introduction to Groups

A group represents a specified set of users. A group always belongs to one organization, but can contain users from different organizations.

In CentraSite, groups are used for the following purposes:

- To assign roles to groups of users. Assigning roles to a group confers the permissions associated with the role to each member of the group.
- To give a group of users access to a specific object in the registry.
- To identify the group of individuals who are authorized to approve certain types of requests.
- To identify the target audience for certain policy actions. For example, the intended recipients of an email action.

System Groups

System groups are shipped with CentraSite. When a user is added to CentraSite, CentraSite automatically adds the user to a specified system group depending on the organization to which the user belongs. The membership of these groups cannot be manually updated or deleted by an administrator. CentraSite provides the following system-defined groups:

System Group Contains...

Everyone	All users, including guests. Ensure users who publish assets to your registry know that this group includes guest users, and that if they grant access to this group, they enable access by anonymous users. This group should only be granted permission to view registry objects. It should not be granted permission to modify or delete registry objects.
Users	All users in an organization. Every organization has a Users group. By default, the Asset Provider and Asset Consumer roles are assigned to this group, which gives these roles to every user in the organization.
Members	All users in an organization or any of its descendant organizations (children, children's children, and so on). Every organization has a Members group. Every user you add to CentraSite automatically becomes a member of the Everyone group.

CentraSite manages the membership of these groups automatically. You cannot delete system groups or edit their membership. You can, however, assign roles and instance-level permissions to system group and use them in all of the same ways as you can a regular user-defined group.

Custom Groups

CentraSite supports static groups and nested groups.

You can create locally managed groups that are defined and maintained within CentraSite. This is a group that consists entirely of active users who are registered in CentraSite. The membership of the group is maintained in CentraSite. You can perform administrative tasks manually on the group in CentraSite, such as adding or removing users from the group. You can switch a locally managed group to an externally managed group.

You can also create externally managed groups that are imported from the external authentication system. You cannot change the name or membership of an externally managed group within CentraSite; CentraSite maintains the membership of externally managed groups by automatically synchronizing with the external authentication system. If the externally managed group includes members who are not existing users of CentraSite, those members does not become CentraSite users as a result of adding the group to CentraSite. If you subsequently add those individuals to CentraSite, however, they automatically becomes members of this group. You cannot switch an externally managed group to a locally managed group.

When you import a group from CentraSite's external authentication system, CentraSite fetches the group's details from the authentication system, creates an externally managed group, and synchronizes (updates) the group's membership in CentraSite.

Whenever a new user is added from the external authentication system, CentraSite queries the external system to determine in which groups the user is a member. If any of those groups have been imported into CentraSite, the user is automatically added to the corresponding externally managed groups in CentraSite. The newly added user automatically receives the permissions and roles that are associated with the corresponding groups.

The removal of a user from a group can be done only in the external authentication system. Whenever a user is removed from the external authentication system, the corresponding user no longer receives the permissions and roles that are associated with the corresponding externally managed groups in CentraSite.

Note:

After the group information is imported, CentraSite does not attempt to keep it synchronized with the authentication system. Any change of the externally managed group is not synchronized with CentraSite. If a user is newly added to the externally managed group in the authentication system at a later stage, in order to keep CentraSite synchronized with the authentication system, you must manually reimport the external group in CentraSite. Likewise, if a user is removed from the externally managed group the authentication system, the corresponding CentraSite user is not automatically deactivated. The CentraSite user associated with a deleted external user must be deactivated manually in CentraSite.

When CentraSite executes a request that references an externally managed group, it accesses the external authentication system to resolve the group's membership. It performs the requested activity for each user who is a member of the specified group and is also a registered user on CentraSite. Users that are named in the externally managed group but are not registered as CentraSite users are ignored.

Assume that User1, User2, User3, User4, and User5 are defined on the external authentication system, and do not belong to any group on the external authentication system. Assume that all of these users except User1 have already been imported from the external authentication system to CentraSite, but do not yet belong to any group in CentraSite. Now assume that a group called

GroupA is created in the external authentication system, and GroupA has members User1, User2, and User3.

If GroupA is imported to CentraSite, the registered CentraSite users User2 and User3 become members of GroupA in CentraSite, as the membership of the group is maintained in external authentication system (User 1 is not registered in CentraSite, therefore it is not available as a member in Group A). You cannot add more users manually to GroupA in CentraSite, since CentraSite just refers to the external authentication system for the membership details. However, if User4 and User5 are added to GroupA in the external authentication system, they also become members of the GroupA in CentraSite when the automatic synchronization occurs.

In this scenario, User1 is not yet a member of GroupA in CentraSite, since User1 is not a registered user in CentraSite. To add User1 to the group in CentraSite, you would define User1 as a user in CentraSite and associate this user with GroupA in the external authentication system.

If your external authentication system already defines groups of users who are significant to your SOA environment (for example, SOA Architects, SOA Project Review Team, or SOA Managers), add them to CentraSite as externally managed groups. Doing so simplifies maintenance by eliminating the need to update two systems when the membership of a group changes.

Note:

Groups that are nested in the external authentication are supported by CentraSite. If you are using LDAP, only the recurse up option is supported for group resolution. The recurse down option is not supported.

Managing Groups through CentraSite Business UI

This section describes operations you can perform to manage groups through CentraSite Business UI.

Adding Group to an Organization

To create a new group, you must have the Manage Organizations permission in CentraSite.

The following general guidelines apply when adding a group:

- A group can be empty.
- Each user can be assigned to zero, one, or more than one group.

> To add a group to an organization

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click an organization to that you want to add the group.

This opens the Organization Details page. Also, the actions bar displays a set of actions that are available for working with the displayed organization.

3. On the actions bar of the Organization Details page, click **Add Group**.
4. In the **Add Group** dialog box, do one of the following:
 - To add a new local group, type the name of the group in the field labeled **Create a new Group**, click **Add** and assign users to the group.
 - To import a group from an external user repository, type the name of the external group in the field labeled **Import an external Group**, and click **Add**.

When you import an external group, you can also select the option **Import all group members as users**. This ensures that all of the users defined in the external group are also added as users to the group in CentraSite. Since the group is assigned to a specific organization, the users is assigned to the same organization.

The newly created group is added to the organization and to the CentraSite registry or repository.

Viewing the Group List

To view the group list, you must have the Manage Organizations permission in CentraSite.

➤ To view the list of groups

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click an organization to which the group belongs.
3. In the Organization Details page, select the **Groups** profile.

This displays a list of defined groups for the organization.

The Organization Details page provides the following information about each group:

Column	Description
Name	Name of the group.
Description	The description for the group.

In addition to the basic details, the **Groups** profile includes the **Delete** action to remove a particular group from the organization.

Viewing Group Details

To view the group details, you must have the Manage Organizations permission in CentraSite.

➤ To view the details of a group

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click an organization to which the group belongs.
3. In the Organization Details page, select the **Groups** profile.
4. In the list of groups, click a group.

This opens the Group Details page. Also, the actions bar displays a set of actions that are available for working with the displayed group.

The group details are displayed in the following profiles:

- **The Basic Information Profile:** Displays the generic attributes that include details about basic group information - Name, Description, Number of Consumers, Number of Consumed Assets.

- **The Group Members Profile:** Displays a list of users that are designated as members of the group. You can click each user to display the User Details page.

In the **Group Members** profile, you can assign users to the group and remove users from the group.

- **The Roles Profile:** Displays a list of roles that are assigned to the group. You can click each role to display the Role Details page.

In the **Roles** profile, you can assign roles to the group and remove roles from the group.

Modifying Group Details

To modify group details, you must have the Manage Organizations permission in CentraSite.

You can perform the group modification tasks from the Group Details page. The modification is broken up across the different profiles in the Group Details page, which means that modifications done in each profile are independent of each other and must be saved individually. The modifications you can perform in each profile is outlined in the subsequent sections.

The following general guidelines apply when modifying an existing group information:

- If you are a user with the Organization Administrator role, you can change the values of the following attributes - **Group Name** and **Group Description**.
- If you are a user with the CentraSite Administrator role, you can additionally change the value of the attribute - **Group Organization**.
- If the group is an external repository group, then the **Edit** icon is disabled, since modifying an external repository group is not supported.

> To modify the basic details of a group

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click the organization to which the group belongs.
3. In the Organization Details page, click the **Groups** profile.
4. Click a group whose details you want to modify.

This opens the Group Details page. Also, the actions bar displays the set of actions that are available for working with the displayed group.

5. To modify the group's details displayed in the **Basic Information** profile, click **Edit**.
6. Modify the values for the group's fields in the Group Details page as required.
7. Click **Save** to update the group information.

Adding User to a Group Through Group Details Page

To add a user to a group, you must have the Manage Organizations permission in CentraSite.

You can add a user to a group in CentraSite Business UI in the following ways:

- Through the Group Details page
- Through the User Details page

Instructions for adding a user from the User Details page is provided in [“Adding User to a Group Through User Details Page” on page 109](#).

➤ To add a user to a group (from the Group Details page)

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click an organization to which the user belongs.
3. In the Organization Details page, click the **Groups** profile.
4. Click a group to which you want to add a user.

This opens the Group Details page. Also, the actions bar displays the set of actions that are available for working with the displayed group.

5. On the actions bar of the Group Details page, click **Assign User**.
6. To see a list of all available users, click the **Search** icon.

You can also type the first few characters of the user ID in the search field, then click the **Search** icon.

This displays all users whose name starts with the given characters.

You can use wildcard characters (* or %) in the search field.

You can sort the users based on attributes, such as user name or the owning organization of the user, by using the **Sort By** list. You can also configure the user attributes that you want to view by using the **View** list.

7. In the list of users, select one or more users you want to add to the group, and then click **Add**.

This add(s) the selected users to the displayed group.

Deleting User from a Group Through Group Details Page

To delete a user from a group, you must have the Manage Organizations permission in CentraSite.

You can delete a user from a group in CentraSite Business UI in the following ways:

- Through the Group Details page
- Through the User Details page

Instructions for deleting a user from the User Details page is provided in [“Deleting User from a Group Through User Details Page” on page 110](#).

The following general guidelines apply when deleting a user from a group:

- When you delete a user from a group, you delete the association between the user and the group, but both the user and the group continue to exist.
- You cannot remove a user from pre-defined group such as Users or Members.

➤ To delete a user from a group (from the Group Details page)

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click an organization to which the user belongs.
3. In the Organization Details page, click the **Groups** profile.
4. Click a group from that you want to remove the user.
5. In the Group Details page, click the **Group Members** profile.
6. In the list of users, hover over the user you want to remove from the group.

This displays icons for one or more actions that you can perform on the user.

7. Click **Delete**.
8. Click **Yes** in the confirmation dialog box.

The user is removed from the group.

Deleting Groups

To delete a group, you must have the Manage Organizations permission in CentraSite.

You might consider deleting a group in CentraSite if you want to:

The following general guidelines apply when deleting groups in CentraSite:

- When you delete a group, you delete all of the assignments of users to the group, but the users continue to exist without the group.
- You cannot delete the `Users` or `Members` groups of an organization. These are pre-defined groups and are created automatically when an organization is created. They will only be deleted if you delete the organization that they belong to.
- Deleting a group from CentraSite does not delete the associated group from the external authentication system.

➤ To delete groups

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click an organization to which the group belongs.
3. In the Organization Details page, click the **Groups** profile.
4. In the list of groups, hover over the group you want to delete.

This displays icons for one or more actions that you can perform on the group.

5. Click **Delete**.
6. Click **Yes** in the confirmation dialog box.

The group is permanently removed from the CentraSite registry or repository.

Managing Groups through CentraSite Control

This section describes operations you can perform to manage groups through CentraSite Control.

Adding Group to an Organization

To create a new group, you must have the Manage Organizations permission or at least the Manage Users permission for an organization in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

You can create a new group in CentraSite Control in the following ways:

- Create a locally managed group
- Bulk load users from the external authentication system
- Create an externally managed group

Adding a Locally Managed Group

> To add a locally managed group

1. In CentraSite Control, go to **Administration > Users > Groups**.
2. In the Groups page, click **Add Group**.
3. In the area labeled **Group Information**, provide the required information for each of the displayed data fields.

Field	Description
Name	Name of the group. This is the name that users will see when they search for groups in CentraSite. A group name can contain any characters (including spaces), and must be unique within an organization.
Description	<i>Optional.</i> The description for the group. This description appears when a user displays the list of groups in the Groups page.
Organization	The organization to which you want to add the group. (The Organization list only displays organizations for which you have the Manage Users permission.)

Important:

- | Field | Description |
|-------|---|
| | Select the organization carefully. You cannot change the organization assignment later. |
4. To add users to the group, click the **Users** tab and perform the following:
 - a. Click **Add User**.
 - b. In the **Add User** dialog box, select the check box for one user, or select the check boxes for multiple users you want to add to the group.
 - c. To filter the list, type a partial string in the **Search** field.
CentraSite applies the filter to the **Name** column.
 - d. Click **OK**.
 5. To assign roles to the group, click the **Roles** tab and perform the following:
 - a. Click **Assign Role**.
 - b. In the **Assign Role** dialog box, select the check box for one role, or select the check boxes for multiple role you want to give the group.
 - c. To filter the list, type a partial string in the **Search** field.
CentraSite applies the filter to the **Name** column.
 - d. Click **OK**.
 6. Click **Save**.

Bulk Loading Groups from External Authentication System

➤ To bulk load groups from an external authentication system

1. In CentraSite Control, go to **Administration > Users > Groups**.
This displays a list of defined groups in the Groups page.
2. In the **Import to Organization** field, select the organization to which you want to add the groups.
3. Click **Bulk Load Groups from External Source**, and select the groups you want to add.

For more information, see [“Selecting Users or Groups from Repository”](#) on page 136.

4. To assign roles to each group, click the **Roles** tab in the Edit Group page and perform the following:
 - a. Click **Assign Role**.
 - b. In the **Assign Role** dialog box, select one or multiple roles you want to assign to the group.
 - c. To filter the list, type a partial string in the **Search** field.
CentraSite applies the filter to the **Name** column.
 - d. Click **OK**.

Adding an Externally Managed Group

➤ To add an externally managed group

1. In CentraSite Control, go to **Administration > Users > Groups**.

This displays a list of defined groups in the Groups page.

2. Click **Add Group**, and select the organization to which you want to add the group.

Note:

You cannot change the organization assignment later.

3. Click **Associate**, and select the group you want to add.

For more information, see [“Selecting Users or Groups from Repository”](#) on page 136.

4. To assign roles to the group, click the **Roles** tab and perform the following:
 - a. Click **Add Role**.
 - b. In the **Add Role** dialog box, select one or multiple roles you want to assign to the group.
 - c. To filter the list, type a partial string in the **Search** field.
CentraSite applies the filter to the **Name** column.
 - d. Click **OK**.

Viewing the Group List

To view the group list, you must have the Manage Organizations permission or at least the Manage Users permission for an organization in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

In CentraSite Control, you can view the list of groups in one of the following ways:

- Through the Groups page.
- Through the Edit Organization page.

➤ **To view the list of groups**

- Through the Groups page. Go to **Administration > Users > Groups** to display a list of groups currently defined in CentraSite. To filter the list to see just a subset of the available groups, type a partial string in the **Search** field.

CentraSite applies the filter to the **Name** column. The **Search** field is a type-ahead field, so as soon as you type any characters, the display is updated to show only those groups whose name contains the specified characters. The wildcard character % is supported.

If you type...	CentraSite displays
b	Names that contain b
bar	Names that contain bar
%	All names

The Groups page provides the following information about each group:

Column	Description
Name	Name of the group.
Organization	Name of the organization to which the group belongs.
Description	Short description of the group.

You can adjust the view to show or hide the individual columns by using the **Select Columns** icon that is located in the upper-right corner of the Groups page.

The shortcut menu of a particular group displays one or more actions that you can perform on that group.

Action	Description
Details	Displays the details page of the group.
Delete	Deletes the group.
Impact Analysis	Helps to easily visualize the associations that exist between the group and registry objects.

- Through the Edit Organization page. Go to **Administration > Organizations**.

1. Right-click an organization and click **Details**.

This opens the Edit Organization page.

2. Click the **Groups** profile.

This displays a list of defined groups in the organization.

Viewing Group Details

To view the group details, you must have the Manage Organizations permission or at least the Manage Users permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

> To view the details of a group

1. In CentraSite Control, go to **Administration > Users > Groups**.
2. Right-click a group and click **Details**.

This opens the Edit Group page.

The area labeled **Group Information** displays the generic attributes that includes details about basic group information - Name, Description, Organization, Associated External Group.

The group details are displayed in the following tabs:

- **The Users Tab:** Displays a list of users that are designated as members of the group. You can click each user to display the User Details page.

In the **Users** tab, you can add users to the group and remove users from the group.

- **The Roles Tab:** Displays a list of roles that are assigned to the group. You can click each role to display the Role Details page.

In the **Roles** tab, you can assign roles to the user and remove roles from the user.

Modifying Group Details

To modify group details, you must have the Manage Organizations permission or at least the Manage Users permission for an organization in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

You can perform the group modification tasks from the Edit Group page. The modification is broken up across the different tabs in the Edit Group page, which means that modifications done in each tab are independent of each other and must be saved individually. The modifications you can perform in each tab is outlined in the subsequent sections.

➤ To modify the basic details of a group

1. In CentraSite Control, go to **Administration > Users > Groups**.

2. Right-click a group and click **Details**.

This opens the Edit Group page.

3. In the area labeled **Group Information**, modify the values for the group's fields as required.

4. To modify the role assignments, click the **Roles** tab and perform the following:

a. Click **Assign Role**.

b. In the **Assign Role** dialog box, select one or more roles you want to assign to or remove from the group.

c. To filter the list, type a partial string in the **Search** field.

CentraSite applies the filter to the **Name** column.

d. Click **OK**.

5. If this is a locally managed group:

a. To modify the group's membership, click the **Users** profile, and add or remove users.

b. To turn the group into an externally managed group, click **Associate**, and associate an external group. The group's current name and membership is replaced by the name and membership of the externally managed group.

The group's current name and membership is replaced by the name and membership of the externally managed group.

For more information, see [“Selecting Users or Groups from Repository”](#) on page 136.

6. Click **Save** to update the group information.

Assigning Roles to Group

To assign roles to a group, you must have the Manage Organizations permission or at least the Manage Users permission for an organization in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

➤ **To assign roles to a group**

1. In CentraSite Control, go to **Administration > Users > Groups**.
2. Right-click a group and click **Details**.

This opens the Edit Group page.
3. Click the **Roles** tab and perform the following:
 - a. Click **Assign Role**.
 - b. In the **Assign Role** dialog box, select one or multiple roles you want to assign to the group.
 - c. To filter the list, type a partial string in the **Search** field.

CentraSite applies the filter to the **Name** column.
 - d. Click **OK**.

Deleting Groups

You can delete a group in CentraSite Control by using the Groups page (if you have the Manage Users permission) or the Edit Organization page (if you have the Manage Organizations permission) in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

You might consider deleting a group in CentraSite if you want to:

The following general guidelines apply when deleting groups in CentraSite:

- When you delete a group, you delete all of the assignments of users to the group, but the users continue to exist without the group.
- You cannot delete the *Users* or *Members* groups of an organization. These are pre-defined groups and are created automatically when an organization is created. They will only be deleted if you delete the organization that they belong to.
- Deleting a group from CentraSite does not delete the associated group from the external authentication system.

➤ To delete groups

1. In CentraSite Control, go to **Administration > Users > Groups**.
2. Right-click a group and click **Delete**.

You can also select multiple groups, click the **Actions** menu, and click **Delete**.

3. Click **OK** in the confirmation dialog box.

Each selected group is permanently removed from the CentraSite registry or repository.

Managing Groups through Command Line Interface

This section describes operations you can perform to manage the groups through the CentraSite Command Line Interface (CLI).

Reassociating Groups

Pre-requisites:

To reassociate a group through the CentraSite Command Line (.cmd) Interface, you must have the CentraSite Administrator role.

Before you run the tool, we strongly recommend that you create a database backup.

In addition to the database backup, make sure the following conditions are satisfied:

- There is a unique registry object for the old external group.
- The old external group can be uniquely identified in the security configuration.
- There is no registry object for the new external group.
- There is no security configuration for the new external group.
- The domain of the new external group exists in the security configuration.
- A GUI configuration does not exist for the new external group.

You can change the association of a CentraSite group from one external group to another. This can be necessary, for example, if the permission assignments for certain CentraSite assets moves from one group to another group in the same authentication domain. By reassociating the group, you can keep the name of the CentraSite group unchanged. Another possible use would be to handle group IDs when the default domain name changes; for example, when switching from operating system authentication to LDAP authentication.

CentraSite provides a Java tool named `ReassociateGroups.jar` for this purpose.

The tool automatically grants the new external group the same permissions that were granted for the old external group.

- Run the Java tool `ReassociateGroups.jar`.

- Given the group ID:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd ReassociateGroups.jar  
<CentraSite URL> <admin user id> <password> groups <old group id> <new group id>
```

- Given the group mapping details:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd ReassociateGroups.jar  
<CentraSite URL> <admin user id> <password> file <group mapping file path>
```

The input parameters are:

Parameter	Description
CentraSite URL	The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
admin user id	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
password	The password for the CentraSite user identified by the parameter <code>admin user id</code> .
old group id	The domain ID of the old external group.
new group id	The domain ID of the new external group.
group mapping file path	The absolute or relative path to the group mapping file. If relative, the path should be relative to the location from where the command is executed.

The group mapping file contains a map of key-value pairs, and defines the group objects as simply lines of comma separated old group and new group IDs (for example, "OLDDOMAIN\oldGroup, NEWDOMAIN\newGroup").

Note:

The same group ID must not be specified more than once in the mapping file.

The reassociation may take some time. The tool's progress is reported to standard output.

Examples (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd ReassociateGroups.jar
http://localhost:53307/CentraSite/CentraSite DOMAIN\admin pAsSw0rD groups
OLDDOMAIN\oldGroup NEWDOMAIN\newGroup
```

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd ReassociateGroups.jar
http://localhost:53307/CentraSite/CentraSite DOMAIN\admin pAsSw0rD file groups.txt
```

Synchronizing LDAP Groups

Pre-requisites:

To synchronize a LDAP group through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

Before you run the tool, Software AG recommends that you create a database backup.

In addition to the database backup, make sure the following conditions are satisfied:

- There is a unique registry object for the external LDAP group.
- The external LDAP group can be uniquely identified in the security configuration.

You might consider synchronizing a LDAP group in CentraSite if you want to update any changes to the group, for example, user additions, user deletions, and user transfers between user groups from the LDAP directory. The synchronization affects users and groups and the user-to-group relationship only.

Assume that User1, User2, User3, and User4 are defined on the external authentication system, and a group called GroupA has members User1, User2, and User3, on the external authentication system. Assume that the users User1, User2, and User4 (except User3) and the Group A have already been imported from the external authentication system to CentraSite.

Now assume that User3 is added to GroupA in the external authentication system, User3 also becomes a member of the GroupA in CentraSite when the synchronization occurs.

CentraSite provides a command tool named `sync Ldap Group` for this purpose.

➤ To synchronize LDAP group

- Run the command `sync Ldap Group`.

```
The syntax is of the format: C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd
sync Ldap Group [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -groupName
<GROUP-NAME>
```

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, Administrator.
PASSWORD	The password for the registered CentraSite user identified by the parameter USER-ID.
GROUP-NAME	The name of the LDAP group to synchronize the group's membership and the user information from the LDAP directory.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd sync Ldap Group -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-groupName SAG\ManageAssets
```

The response to this command could be:

```
Executing the command: sync Ldap Group
Group SAG\ManageAssets synchronised successfully.
Successfully executed the command: sync Ldap Group
```

Selecting Users or Groups from Repository

Select Users or Groups from Operating System User Repository

The following general guidelines apply when searching for users or groups in the operating system user repository:

- When searching for users, CentraSite searches the user ID attribute, not the user name attribute. CentraSite automatically filters out users that have already been added to CentraSite.
- To list all users or groups, type % or * as the search string. You cannot combine either wildcard with other characters. For example, the search strings ab% and %ab are not valid.
- To find specific users or groups, type a search string that specifies the characters with which the user ID or group name begins. The domain portion of the name is not included in the search. For example, if a user has the user ID MyDomain\AdminUser01, a search for Ad will find the user, whereas a search for User01 or My will not.
- Search strings are not case or accent sensitive.
- If the user is not known to the local system but is known to a domain server to which the local operating system is connected, type the user's domain-qualified name into the **Type Domain Name field**.

Note:

If you type a user ID in the **Type Domain Name** field, CentraSite ignores any selections you have made in the user list.

Select Users or Groups from Active Directory or LDAP Repository

The following general guidelines apply when searching for users or groups in an Active Directory or LDAP repository:

- The Active Directory or LDAP authentication system performs a user search based on the attribute mapping specified in the authentication configuration, and displays the users that fit the search criteria.
- In the **Search Criteria** panel, create the search criteria by selecting the attribute and the condition from the list boxes, typing the search string in the text box and selecting **Equals** or **NotEquals**.
- CentraSite treats the search string as a partial string. For example, if you enter a1, then Alex, Allen, and Sally all fit the search criteria.
- You can use an asterisk (*) as a wildcard in the search string. CentraSite replaces the wildcard symbol with as many characters as necessary.
- Search strings are not case or accent sensitive.
- To add more search conditions, click the plus button. If all conditions must be met, select **AND Condition**. If only one condition must be met, select **OR Condition**.

5 Role Management

■ Introduction to Permissions and Roles	162
■ Managing Roles through CentraSite Business UI	169
■ Managing Roles through CentraSite Control	176

Introduction to Permissions and Roles

Permissions determine the operations users can perform and the set of objects users can access.

Instance-Level Permissions

This following table lists instance-level permissions and the actions they enable a user or group to perform:

Permission	Enables specified user or group to...
View	<ul style="list-style-type: none">■ View an object■ View a folder and its properties■ View a file and its properties
Modify	<ul style="list-style-type: none">■ Edit an object■ Create files and subfolders in a folder■ Edit a file and its properties■ View instance-level permissions
Full	<ul style="list-style-type: none">■ Edit an object■ Create files and subfolders in a folder■ Edit a file and its properties■ View instance-level permissions

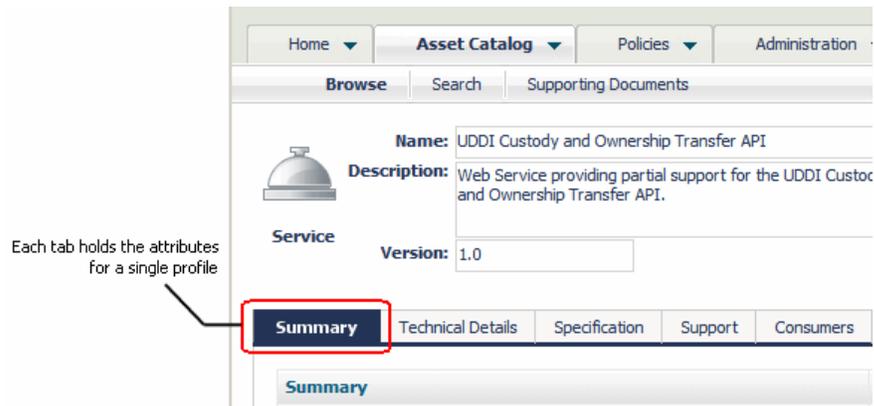
By default, a user has implicit and irrevocable Full permission on all the objects the user owns.

The object types that support access control at the instance level are assets, repository folders and files, report templates, design/change-time policies, run-time policies, and taxonomies. All CentraSite users, including guests, have implicit and irrevocable permission to view all instances of report templates, policies, and taxonomies. However, you can use instance-level permissions to restrict the ability to edit and delete them.

Access to other types of objects is controlled using the broader role-based permissions or is enabled contextually. An object is a constituent of some other access-controlled object. For example, the individual operations and bindings associated with a web service are objects that can only be accessed within the context of the Service object itself. Therefore, the permissions that control access to the Service object also control access to the service's constituent objects.

You can set instance-level permissions through CentraSite Control and CentraSite Business UI, and you can create design/change-time policies to automate the assignment of instance-level permissions on certain types of objects (specifically, assets and policies). For example, you might use a design/change-time policy to automatically extend access to specified groups of consumers when an asset switches to the Deployed state.

Assets include an additional level of access control called a *profile permission*. Profile permissions enable you to control access to individual *profiles* within an instance of an asset. A profile is a collection of attributes. It is used to group the metadata for an asset when the asset is displayed in the user interface. Profiles enable CentraSite Control, CentraSite Business UI, and the CentraSite plug-in for Eclipse to present the details for an asset in an organized manner. In CentraSite Control, for example, all attributes associated with a particular profile are grouped on a separate tab.



Profile permissions determine the profiles a user sees when he or she views an asset with CentraSite Control, CentraSite Business UI, or the CentraSite plug-in for Eclipse. You might use profile permissions, for example, to limit the amount of information that consumers see for an asset. Profile permissions restrict access at the UI level but not the API level. At the API level, profile permissions are irrelevant. A user with view permission on an asset can access all the asset's metadata through the API, regardless of whether profile permissions exist for the asset.

Roles and Role-Based Permissions

A role is a set of role-based permissions. Role-based permissions enable users to access areas of the CentraSite Control and CentraSite Business UI and to create and manage (view, edit, and delete) certain types of registry and repository objects. You can assign roles to a user or a group; the user or users in the group receive the permissions specified in the role.

By default, all CentraSite users, including guests, have permission to use the asset details of the CentraSite Control and CentraSite Business UI, and all users except guests have permission to provide and consume assets within their own organization. To use the other parts of the user interface or have access to other objects, a user must belong to a role that includes the appropriate permission.

Role-based permissions are at two levels:

- *System-wide permissions* grant access to objects that are available to all organizations, such as taxonomies and asset types. Additionally, some system-wide permissions grant access to all objects of given type in any organization in the registry or repository. System-wide permissions are generally given only to a small group of high-level administrators.
- *Organization-specific permissions* grant access to all objects of a certain type within one organization. Permissions that enable access to assets, policies, and lifecycle models are organization-specific.

Name	Organization
View Supporting Documents	All
Manage System-wide Lifecycle Models	All
Manage System-wide Design/Change-Time Policies	All
Manage Asset Types	All
Create UDDI Subscriptions	All
Manage System-wide Roles	All
Use the Operations UI	All
View Assets	Customer Care
Create Assets	Customer Care
Modify Assets	Customer Care
Manage Assets	Customer Care

System-wide permissions are granted for all organizations

Organization-specific permissions are granted for a specific organization

CentraSite comes with predefined roles for each level of permissions. If necessary, you can create custom roles.

- **System-level roles** contain permissions that enable users to work with system-wide objects. Some of these roles also enable users to manage (view, edit, or delete) all instances of a given object type in any organization in the CentraSite registry or repository.
- **Organization-wide roles** contain permissions for working with organization-specific objects. CentraSite maintains a set of organization-level roles for each organization in the registry or repository.

System-Level Role-Based Permissions and Predefined Roles

The following table lists system-level permissions:

Permission	Grant the right to...
Use the Administration UI	Use the indicated area in CentraSite Control.
View {Policy Log Approval History}	View the indicated item. Administration UI permission is implied.
Register as Consumer	Register as a consumer of assets.
Manage Organizations	Manage all organizations in the CentraSite instance. Manage Organizations (org-level permission for every organization), Manage System-wide Design/Change-Time Policies, Manage System-wide Run-Time Policies, Manage Lifecycle Models (org-level permission for every organization), and Manage Report Templates permissions are implied.

Note:

All organizations are visible to everyone.

Permission	Grant the right to...
Manage System-wide Lifecycle Models	Manage lifecycle models. Administration UI, Manage System-wide Design/Change-Time Policies, Manage System-wide Runtime Policies, and Manage Lifecycle Models (org-level permission for every organization) permissions are implied.
Manage System-wide Design/Change-Time Policies	Manage the indicated policies as well as Action Categories, Action Templates, and Action Parameters. Policy UI permission is implied.
Manage Report Templates	Manage system-wide report templates. Reports UI permission is implied.
Manage System-wide Roles	Manage system-level roles. Administration UI permission is implied.
Manage UDDI Subscriptions	Manage UDDI Subscriptions. View and Create UDDI Subscriptions permissions are implied.
Create UDDI Subscriptions	Create new UDDI Subscriptions and view existing UDDI Subscriptions. View UDDI Subscriptions permission is implied.
View UDDI Subscriptions	View UDDI Subscriptions. Administration UI permission is implied.
Manage {Taxonomies Asset Types}	Manage the indicated items. Administration UI permission is implied.
Manage Supporting Documents	Manage the content of all folders in the Supporting Documents Library (SDL). View Supporting Documents permission is implied.
View Supporting Documents	View the content of all folders in the SDL.

This following table lists predefined system-level roles provided with CentraSite.

System-Level Role	Permissions
CentraSite Administrator (CSA)	All system-wide permissions. Can create, delete, or change the permission set for a system-level role. This role cannot be modified or deleted. At least one user must be assigned to this role at all times.
Asset Type Administrator	Use Home and Administration UIs; view policy log and approval history; manage system-wide lifecycle models, design/change-time policies, taxonomies, and asset types.
Operations Administrator	Use Home, Policy, Reports, and Operations UIs; view policy log and approval history, manage system-wide lifecycle models,

System-Level Role	Permissions
	design/change-time policies, runtime policies, report templates, runtime gateways, and event types; manage and create UDDI subscriptions.
Guest	Browse and use the asset catalog.

Organization-Specific Role-Based Permissions and Predefined Roles

This following table lists organization-specific permissions.

Permission	Grants the right to...
Manage Assets	Manage assets and supporting documents within an organization. View, Create, and Modify Assets permissions are implied.
Create Assets	Create assets within an organization.
Modify Assets	Modify all assets and supporting documents within an organization. Also important for performing consumer registrations. View Assets permission is implied.
View Assets	View all assets and supporting documents within an organization.
Manage Design/Change-Time Policies	Manage design/change-time policies within an organization. Implies the right to manage all policy-related objects such as policy conditions and policy parameters. Policy UI permission is implied.
Manage Run-Time Policies	Manage run-time policies within an organization. Implies the right to manage all policy-related objects such as policy conditions and policy parameters. Policy UI permission is implied.
Manage Lifecycle Models	Manage lifecycle models (LCMs) within an organization. Administration UI, Modify Assets, Manage Design/Change-Time Policies, and Manage Runtime Policies permissions are implied.
Manage Users	Manage users, groups, and roles within an organization. Administration UI permissions is implied.
Manage Organizations	Manage an organization, all its child organizations, and the organization folder in the SDL. Manage Users, Manage Design/Change-Time Policies, Manage Runtime Policies, Manage Lifecycle Models, and Manage Assets permissions are implied.

Note:

By default, the content of an organization folder is visible to every user of that organization. All organizations are visible for everyone.

This following table lists predefined organization-level roles provided with CentraSite.

Organization-level Permissions

Role

API Gateway Administrator	<p>Manage API Gateway instances in an organization.</p> <p>Note: If you are upgrading to CentraSite 10.1 from an earlier version, the users who had been assigned the Mediator Administrator role in the previous versions of CentraSite are automatically assigned the API Gateway Administrator role.</p>
API Gateway Publisher	<p>Publish and unpublish APIs to and from API Gateway instances in an organization.</p> <p>Note: If you are upgrading to CentraSite 10.1 from an earlier version, the users who had been assigned the Mediator Publisher role in the previous versions of CentraSite are automatically assigned the API Gateway Publisher role.</p>
Asset Administrator	Use Home and Reports UIs; view policy log and approval history; register as consumer; manage assets and lifecycle models.
Asset Consumer	Use Home and Reports UIs; register as consumer, view assets. By default, all users in an organization receive this role.
Asset Provider	Use Home and Reports UI; register as consumer; create assets. By default, all users in an organization receive this role.
Mediator Administrator	Manage Mediator gateway instances in an organization.
Mediator Publisher	Publish and unpublish run-time policies and APIs to and from Mediator gateway instances in an organization.
Organization Administrator	All organization-specific permissions for an organization. Can use the Home, Policy, Administration, and Reports UIs, and view, edit or delete any object within an organization or the organization's descendants. This role cannot be modified or deleted. At least one user in an organization must be assigned to this role at all times.
Policy Administrator	Use Home and Policy UIs; view policy log and approval history; view assets, manage design/change-time and runtime policies.
API Portal Administrator	Manage API Portal gateway instances in an organization.
API Portal Publisher	Publish and unpublish APIs to and from API Portal gateway instances in an organization.
API Runtime Provider	Manage run-time policies and configure run-time actions for virtual APIs in an organization.

Considerations when Working with Instance-Level and Role-Based Permissions

- A user always receives the *union* of all permissions that are granted.
- If you grant access to an object type using a role-based permission, the users of that role can access all objects of that type within the organization. You cannot selectively hide objects from certain users.
- If you grant access to an asset using a role-based permission, the users of that role can view all profiles for the asset. You cannot selectively hide profiles from certain users. If you need to hide or reveal certain profiles as an asset progresses through its lifecycle states, consider creating policies to automatically set the appropriate profile permissions when the asset switches state.
- Users that have been granted a role-based permission receive the specified level of access (View, Modify, or Manage). You can selectively increase this level of access for individual users, but you cannot selectively reduce it.
- Grant instance-level permissions to groups rather than individual users unless you have a specific reason to do so. Doing so gives you greater flexibility and makes permission changes easier to manage.
- If you grant access using instance-level permissions, you configure permissions on each asset individually. If you routinely use instance-level permissions, consider creating a policy to do this for you automatically.
- If you grant instance-level permissions to an external group (that is, a group that is defined and managed in your external authentication system), it might take CentraSite longer than usual to remove those permission assignments from a registry object.

Configuring the Default Roles that CentraSite Assigns to Users in an Organization

By default, CentraSite assigns the Asset Provider and Asset Consumer roles to an organization's Users group. Consequently, every user that you add to an organization is assigned with these roles.

If you do not want users in your organization to have these roles assigned automatically, or if you want to customize the set of permissions that users are assigned by default, you can do any of the following:

- Remove the Asset Provider and Asset Consumer roles from the organization's Users group.
- Modify the set of permissions associated with the Asset Provider and Asset Consumer roles that are assigned to the Users group.
- Create custom roles and assign them to the organization's Users group (instead of, or in addition to, the Asset Provider and Asset Consumer roles).

For example, if you want to create an organization whose users can only consume assets, you have to remove the Asset Provider role from that organization's Users group. Doing this ensures that users added to the organization only receive permission to view assets. (An administrator could,

of course, selectively give specific users in the organization permission to publish assets as necessary.)

Important Things to Know When Upgrading to API Gateway 10.1

After upgrading to CentraSite 10.1 from an earlier version, the users who were created with Mediator Administrator and Mediator Publisher roles in previous versions of CentraSite and transferred to CentraSite 10.1 will automatically have the API Gateway Administrator and API Gateway Publisher roles.

Managing Roles through CentraSite Business UI

This section describes operations you can perform to manage roles through CentraSite Business UI.

Adding Role to an Organization

To create a new role, you must have the Manage Organizations permission in CentraSite.

The following general guidelines apply when adding a role:

- Custom roles can contain both system-wide and organization-specific permissions. A custom role can contain organization-specific permissions for multiple organizations (for example, you can create a role that allows a user to manage the policies in two different organizations).
- Do not create a role that is equivalent to the CentraSite Administrator role. The CentraSite Administrator role is specifically optimized to maximize performance. An equivalent role does not perform as efficiently as the predefined CentraSite Administrator role that is installed with CentraSite.
- You can define a role without assigning it (yet) to a user or group. Each user or group can have zero, one, or more than one role assignments.

➤ To add a role to an organization

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click an organization you want to specify the role assignment.

This opens the Organization Details page. Also, the actions bar displays a set of actions that are available for working with the displayed organization.

3. On the actions bar of the Organization Details page, click **Add Role**.
4. In the **Add Role** dialog box, type a name for the new role and provide a description.

Also select the permissions that have to be assigned to the role. The dialog box displays only the permissions that are appropriate for the logged-in user. If, for example, you are a user with the role CentraSite Administrator, the dialog box displays all available permissions (that is,

organization-specific and system-wide permissions), otherwise the dialog box displays just the organization-specific permissions.

5. Click **Add**.

The newly created role is added to the organization and to the CentraSite Registry Repository.

Viewing the Role List

To view the role list, you must have the Manage Organizations permission in CentraSite.

> To view the list of roles

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click an organization to which the role belongs.
3. In the Organization Details page, select the **Roles** profile.

This displays a list of defined roles for the organization.

The Roles section provides the following information about each role:

Column	Description
Name	Name of the role.
Description	The description for the role.

In addition to the basic details, the **Roles** profile includes the **Delete** action to remove a particular role from the organization.

Viewing Role Details

To view the role details, you must have the Manage Organizations permission in CentraSite.

> To view the details of a role

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click an organization to which the role belongs.
3. In the Organization Details page, select the **Roles** profile.
4. In the list of roles, click the role for which you want to display the details.

This opens the Role Details page. Also, the actions bar displays a set of actions that are available for working with the displayed role.

The role details are displayed in the following profiles:

- **The Basic Information Profile:** Displays the generic attributes that include details about basic role information - Name, Description.
- **The Users Profile:** Displays a list of users that are assigned with the role. You can click each user to display the User Details page.

In the **Users** profile, you can add the role to a set of users and remove the role from a set of users.

- **The Groups Profile:** Displays a list of groups that are assigned with the role. You can click each group to display the Group Details page.

In the **Groups** profile, you can add the role to a set of groups and remove the role from a set of groups.

- **The Permissions Profile:** Displays a list of permissions that are assigned to the role.

Modifying Role Details

To modify role details, you must have the Manage Organizations permission in CentraSite.

You can perform the role modification tasks from the Role Details page. The modification is broken up across the different profiles in the Role Details page, which means that modifications done in each profile are independent of each other and must be saved individually. The modifications you can perform in each profile is outlined in the subsequent sections.

If you are a user with the Organization Administrator role, you can change the values of the following attributes - **Role Name** and **Role Description**.

If you are a user with the CentraSite Administrator role, you can additionally change the value of the attribute - **Owning Organization**.

➤ To modify the basic details of a role

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click an organization to which the role belongs.
3. In the Organization Details page, click the **Roles** profile.
4. Click a role whose details you want to modify.

This opens the Role Details page. Also, the actions bar displays a set of actions that are available for working with the role.

5. To modify the role's details displayed in the **Basic Information** profile, click **Edit**.

6. Modify the values for the role's fields in the Role Details page as required.
7. Click **Save** to update the role information.

Assigning Permissions to Role

To modify role permissions, you must have the Manage Organizations permission in CentraSite.

The permissions originally assigned to a role are defined when you create the role. You can assign additional permissions to a Role.

> To assign permissions to a role

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click an organization to which the role belongs.
3. In the Organization Details page, click the **Roles** profile.
4. Click a role whose permissions you want to modify.

This opens the Role Details page. Also, the actions bar displays the set of actions that are available for working with the displayed role.

5. On the actions bar of the Role Details page, click **Assign Permissions**.
6. In the **Assign Permissions** dialog box, select one or more permissions you want to assign to the role, and then click **Add**.

Removing Permission from Role

To modify role permissions, you must have the Manage Organizations permission in CentraSite.

The permissions originally assigned to a role are defined when you create the role. You can remove permission assigned to a role. when required.

> To remove permissions from a role

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click an organization to which the role belongs.
3. In the Organization Details page, click the **Roles** profile.
4. Click a role whose permissions you want to remove.

This opens the Role Details page. Also, the actions bar displays a set of actions that are available for working with the displayed role.

5. Click the **Permissions** profile.
6. In the list of permissions, hover over the permission you want to remove from the role.
This displays icons for one or more actions that you can perform on the permission.
7. Click **Delete**.
8. Click **Yes** in the confirmation dialog box.

Assigning Role to a User or Group

To assign a role to a user or a group, you must have the Manage Organizations permission in CentraSite.

You can assign a role to a user or a group in CentraSite Business UI using the **Assign Role** action in the User Details page or the Group Details page.

➤ To assign a role to a user or group

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click an organization to which the user or group belongs.
3. In the Organization Details page, do one of the following:
 - Click the **Users** profile. This displays a list of users belonging to the organization.
 - Click the **Groups** profile. This displays a list of groups belonging to the organization.
4. Click a user or group to which you want to assign a role.

This opens the User Details page or the Group Details page. Also, the actions bar displays a set of actions that are available for working with the displayed role or group.

5. On the actions bar of the User Details page or the Group Details page, click **Assign Role**.

This opens the **Assign Role** dialog box.

6. To see a list of all available roles, click the **Search** icon.

Alternatively, you can also type the first few characters of the role name in the search field, then click the **Search** icon. This displays all roles whose name starts with the given characters. You can use wildcard characters (* or %) in the search field.

You can sort the roles based on the role attributes available with the **Sort By** list. You can also configure the role attributes that you want to view by using the **View** list.

7. In the list of roles, select one or multiple roles that you want to assign to the user or group, and then click **Add**.

This add(s) the role to the user or the group.

8. Click **Save**.

Deleting Role Assignments from a User or Group

You can remove a role assignment from a user or a group in CentraSite Business UI using the User Details page or the Group Details page. To remove a role assignment, you must have the Manage Organizations permission in CentraSite.

Note:

If you delete a role assignment, the affected user or group loses the permissions associated with the role, unless the user or group has another role assignment that provides the same permissions.

The following general guidelines apply when deleting a role assignment from a user or group:

- You can remove the role from the set of roles defined for the user or group.
- You can remove the user or group from the set of users and groups to which this role has been assigned.

Deleting Role from Set of Roles Assigned to User or Group

➤ To delete role from set of roles assigned to user or group

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click an organization to which the user or group belongs.
3. In the Organization Details page, do one of the following:
 - Click the **Users** profile. This displays a list of users belonging to the organization.
 - Click the **Groups** profile. This displays a list of groups belonging to the organization.
4. Click a user or group from which you want to remove a role.
5. In the User Details page or the Group Details page, click the **Roles** profile.
6. In the list of roles, hover over the role you want to remove from the user or group.

This displays icons for one or more actions that you can perform on the role.

7. Click **Delete**.
8. Click **Yes** in the confirmation dialog box.

Removing User or Group from Set of Users and Groups Assigned to a Role

> To remove a user or group from the set of users and groups assigned to a role

1. In the CentraSite Business UI activity bar, click **Organizations**.
2. Click an organization to which the role belongs.
3. In the Organization Details page, click the **Roles** profile.
4. Click a role from which you want to remove a user or group.
5. In the Role Details page, do one of the following:
 - Click the **Users** profile. This displays a list of users assigned to the role.
 - Click the **Groups** profile. This displays a list of groups assigned to the role.
6. In the list of users or groups, hover over the user or group you want to remove.
This displays icons for one or more actions that you can perform on the user or group.
7. Click **Delete**.
8. Click **Yes** in the confirmation dialog box.

Deleting Roles

To delete a role, you must have the Manage Organizations permission in CentraSite.

You might consider deleting a role in CentraSite if you:

- No longer require a role that has been previously defined for an organization.

The following general guidelines apply when deleting roles in CentraSite:

- You can delete the role, provided that it is not currently assigned to any user or group.
- You cannot delete pre-defined system-wide roles from the Default Organization.

> To delete roles

1. In the CentraSite Business UI activity bar, click **Organizations**.

2. Click an organization to which the role belongs.
3. In the Organization Details page, click the **Roles** profile.
4. In the list of roles, hover over the role you want to delete.

This displays icons for one or more actions that you can perform on the role.

5. Click **Delete**.
6. Click **Yes** in the confirmation dialog box.

This removes the role from the organization.

Managing Roles through CentraSite Control

This section describes operations you can perform to manage roles through CentraSite Control.

Adding Role to an Organization

To create a new role, you must have the Manage Organizations permission or at least the Manage Users permission for an organization in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

The following general guidelines apply when adding a role to an organization:

- Custom roles can contain both system-wide and organization-specific permissions. A custom role can contain organization-specific permissions for multiple organizations (for example, you can create a role that allows a user to manage the policies in two different organizations).
- Do not create a role that is equivalent to the CentraSite Administrator role. The CentraSite Administrator role is specifically optimized to maximize performance. An equivalent role does not perform as efficiently as the predefined CentraSite Administrator role that is installed with CentraSite.

➤ To add a role to an organization

1. In CentraSite Control, go to **Administration > Users > Roles**.
2. In the **Roles** page, click **Add Role**.

This opens the **Add Role** dialog box.

3. In the **Role Information** section, provide the required information for each of the displayed data fields.

Field	Description
Name	<p>Name of the role.</p> <p>This is the name that users will see when they search for roles in CentraSite.</p> <p>A role name can contain any characters (including spaces), and must be unique within an organization.</p>
Description	<p><i>Optional.</i> The description for the role.</p> <p>This description appears when a user displays the list of roles in the Roles page.</p>
Organization	<p>The organization to which you want to add the role. (The Organization list only displays organizations for which you have the Manage Users permission.)</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Important: Select the organization carefully. You cannot change the organization assignment later.</p> </div>

4. To assign permissions to the role, click the **Permissions** tab. Follow these steps:
 - a. Click **Assign permissions**.
 - b. In the **Assign permissions** dialog box, select one or more permissions you want to add to the role.
 - c. Click **OK**.
5. Click **Save**.

Viewing the Role List

To view the role list, you must have the Manage Organizations permission or at least the Manage Users permission for an organization in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

In CentraSite Control, you can view the list of roles in one of the following ways:

- Through the Roles page.
- Through the Edit Organization page.

> To view the list of roles

- Through the Roles page. Go to **Administration > Users > Roles** to display a list of roles currently defined in CentraSite. To filter the list to see just a subset of the available roles, type a partial string in the **Search** field.

CentraSite applies the filter to the **Name** column. The **Search** field is a type-ahead field, so as soon as you type any characters, the display is updated to show only those roles whose name contains the specified characters. The wildcard character % is supported.

If you type...	CentraSite displays
b	Names that contain b
bar	Names that contain bar
%	All names

The Roles page provides the following information about each role:

Column	Description
Name	Name of the role.
Organization	Name of the organization to which the role belongs.
Description	Short description of the role.

You can adjust the view to show or hide the individual columns by using the **Select Columns** icon that is located in the upper-right corner of the Roles page.

The shortcut menu of a particular role displays one or more actions that you can perform on that role.

Action	Description
Details	Displays the details page of the role.
Delete	Deletes the role.
Impact Analysis	Helps to easily visualize the associations that exist between the role and registry objects.

- Through the Edit Organization page. Go to **Administration > Organizations**.
 1. Right-click an organization whose roles you want to view, and click **Details**.

This opens the Edit Organization page.
 2. Click the **Roles** profile.

This displays a list of defined roles in the organization.

Viewing Role Details

To view the role details, you must have the Manage Organizations permission or at least the Manage Users permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

➤ **To view the details of a role**

1. In CentraSite Control, go to **Administration > Users > Roles**.
2. Right-click a role for which you want to display the details, and click **Details**.

This opens the Edit Role page.

The **Role Information** section displays the generic attributes that includes details about basic role information - Name, Description, Organization.

The role details are displayed in the following tab:

- **The Permissions Tab:** Displays a list of permissions that are assigned to the role.

In the **Permissions** tab, you can assign permissions to the role and remove permissions from the role.

Modifying Role Details

To modify role details, you must have the Manage Organizations permission or at least the Manage Users permission for an organization in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

You can perform the role modification tasks from the Edit Role page. The modification is broken up across the different tabs in the Edit Role page, which means that modifications done in each tab are independent of each other and must be saved individually. The modifications you can perform in each tab is outlined in the subsequent sections.

➤ **To modify the basic details of a user**

1. In CentraSite Control, go to **Administration > Users > Roles**.
2. Right-click a role whose details you want to modify, and click **Details**.

This opens the Edit Role page.

3. In the **Role Information** section, modify the values for the role's fields as required.
4. To modify the permission assignments, click the **Permissions** tab and perform the following:
 - a. Click the **Assign Permission** button.
 - b. In the **Assign Permission** dialog box, select one or more permissions you want to assign to or remove from the role.
 - c. Click **OK**.
5. Click **Save**.

Deleting Roles

You can delete a role in CentraSite Control by using the Roles page (if you have the Manage Users permission) or the Edit Organization page (if you have the Manage Organizations permission) in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

You might consider deleting a role in CentraSite if you:

- No longer require a role that has been previously defined for an organization.

The following general guidelines apply when deleting roles in CentraSite:

- You can delete the role, provided that it is not currently assigned to any user or group.
- You cannot delete pre-defined system-wide roles from the Default Organization.

> To delete custom roles

1. In CentraSite Control, go to **Administration > Users > Roles**.
2. Right-click a role you want to delete, and click **Delete**.

You can also select multiple roles, click the **Actions** menu, and click **Delete**

3. Click **OK** in the confirmation dialog box.

Each selected role is permanently removed from the CentraSite registry or repository.

6 Type Management

■ Introduction to Types	182
■ Basic Components of Type	182
■ Classification of Types	200
■ Composite Asset Types	208
■ Managing Types through CentraSite Business UI	241
■ Managing Types through CentraSite Control	253
■ Managing Types through Command Line Interface	274

Introduction to Types

A *type* (also called an object type) is analogous to a class in object-oriented programming and describes a kind of object that the registry can store. Objects abstract the real-world entities and each object belongs to a particular type that defines its characteristics and behavior.

All items stored in the CentraSite registry are *objects* of a particular type. Users, policies, and taxonomies are examples of objects that are stored in the registry. An *asset* is a specific kind of object that represents an artifact in your SOA environment such as a Web service, a REST service, XML schema, or a business process. In other words, all assets are objects, but not all objects are assets.

Types are system-wide objects, that is, they apply to all organizations. Consequently, all organizations within a particular instance of CentraSite use (or have access to) the same global set of types.

CentraSite includes many predefined types. You can customize many of these predefined types and also create custom types of your own.

Basic Components of Type

A type consists of two basic components:

- Attributes that represent an individual characteristic, property or piece of information about an asset.
- Profiles that represent a logical collection of attributes.

Attributes

A type is made up of attributes. An attribute represents an individual characteristic, property, or piece of information about an asset. All asset types include a basic set of attributes for general information such as the asset's name, description, creation date, and owner. For example, the Service asset type includes attributes that identify the name of the service, provide the service's endpoints, identify the owner of the service, and supply links to programming documentation.

You define additional attributes to hold data that is specific to the type of asset that you want to store in the registry. When you define an attribute, you specify:

- The type of data that the attribute holds (for example, String, Number, Boolean).
- Whether the attribute holds a single value or multiple values (that is, an array).
- Whether an attribute is required or optional.
- Whether the attribute is read-only.

Basic Attributes

All types that represent assets include the basic set of attributes. The following table describes the basic attributes that are available in CentraSite Control and CentraSite Business UI:

Attribute	Description	CentraSite Control	CentraSite Business UI
Name	The name under which the asset is cataloged.	✓	✓
Description	A descriptive comment that provides additional information about an asset.	✓	✓
Key	The Universally Unique Identifier (UUID) that is assigned to the asset and uniquely identifies it within the registry. CentraSite automatically assigns a UUID to an asset when the asset is added to the registry.	✓	
Version	The user-assigned version identifier for an asset. The user-assigned identifier can include any sequence of characters. It is not required to be numeric.	✓	✓
System Version	<p>The system-assigned version number that CentraSite maintains for its own internal use. CentraSite automatically assigns this identifier to an asset when a version of the asset is created. The system-assigned identifier is always numeric and always has the format:</p> <p><i>MajorVersion.Revision</i></p> <p>Where:</p> <ul style="list-style-type: none"> ■ <i>MajorVersion</i> is an integer that represents the asset's version number. This value is incremented by one when a new version of the asset is generated, for example, 1.0, 2.0, 3.0. ■ <i>Revision</i> is an integer that represents an update to a particular version of an asset. When the revisioning feature is enabled for CentraSite, the <i>Revision</i> number is incremented each time a change is made to the asset (for example, 1.0, 1.1, 1.2). 	✓	

Attribute	Description	CentraSite Control	CentraSite Business UI
	An asset's System Version attribute cannot be deleted or modified by a user.		
Created	The date on which the asset was added to the catalog. CentraSite automatically sets this attribute when a user adds the asset to the catalog. Once it is set, it cannot be modified.	✓	✓
Last Modified	The date on which the catalog entry for the asset was last updated. CentraSite automatically updates this attribute when a user modifies any of the asset's attributes.	✓	
Organization	The organization to which the asset belongs.	✓	✓
Owner	The user who currently owns the asset. CentraSite automatically sets this attribute when a user adds the asset to the catalog.	✓	✓
Lifecycle State	The asset's current lifecycle state. If a lifecycle model has been associated with an asset type, CentraSite updates this attribute as the asset passes through its lifecycle.	✓	✓
Last Updated	The date on which an instance of the asset was last updated. CentraSite automatically updates this attribute when a user modifies any of the asset's attributes.		✓
Consumers	The list of users, applications, and arbitrary assets that are registered to consume the asset.		✓
Watchers	The list of users registered to receive notifications when changes are made to the asset.		✓
Pending Approvals	The list of pending approval requests.		✓

An asset can also have any number of additional attributes that are specific to the asset's type. For example, an asset might include attributes that do the following:

- Provide contact information for technical support (for example, phone numbers and email addresses).
- Classify the asset according to one or more taxonomies.
- Describe an asset's relationship to other assets or registry objects.
- Specify details regarding system requirements and technical specifications.
- Provide links to additional information such as program documentation, sample code, or usage notes.

Computed Attributes

CentraSite offers you the flexibility to add computed attributes into asset type definitions and profiles; this allows you to define attributes that require complex computation in Java, and then implement them as a Java plug-in, thus overcoming the limitations of predefined attribute types. You could, for example, make attribute values localizable by using the computed attributes.

A computed attribute must describe its scale for the rendering within the profile of the asset type.

For a Java-based plug-in for a computed attribute, you create a jar file that contains the plug-in definition and you load the jar file through CentraSite Control into the repository.

After you have added a computed attribute into a profile definition, you can perform administration tasks on the computed attribute in the same way as for normal attributes. For example, you can define the ordering of the attributes in a profile, regardless of whether they are standard attributes or computed attributes.

Attribute Data Types

When you add an attribute to a type, you specify the attribute's *data type*. The data type determines what kind of information the attribute can hold. After you add an attribute to a type, the attribute's data type cannot be changed.

The following table lists the data types that you can assign to an attribute. Most types can be configured to hold a single value or multiple values (that is, an array of values).

Data Type	Description
Boolean	Holds a true or false value.

Note:

When a Boolean value is displayed in the CentraSite user interface, its value is generally displayed as Yes (if the attribute's value is true) or No (if the attribute's value is false).

Computed Attribute	Holds a value that is supplied by a user-defined Java plug-in.
--------------------	--

Data Type	Description
	After you have defined a computed attribute, you can use it in CentraSite Control in the same way as any other attribute. You can, for example, assign the attribute to a profile or reorder the attribute position within a profile.
Date/Time	Holds a timestamp that represents a specific date and time.
Duration	Holds a value that represents a period of time as expressed in Years, Months, Days, Hours, Minutes, and Seconds.
Email	Holds an email address. This data type only accepts values in the format: <i>anyString@anyString</i>
	Note: When a user enters a value for an Email attribute, CentraSite verifies that the value conforms to the format above, but it does not attempt to validate the address itself.
International String	Holds a String attribute that can have different values for different locales.
IP Address	Holds a numeric IP address in the v4 or v6 format.
Multiline String	Holds a string of text. When this type of string is displayed in a CentraSite user interface, the string is displayed in a multi-line text box and lines of text are wrapped to fit the width of the box. (Compare this with the String data type described in this table.) The Internationalized option allows you to store the text in internationalized string format.
Number	Holds a numeric value. When you define an attribute of this type, you can specify the number of decimal positions that are to be shown when the attribute is displayed in a user interface. If you do not want to restrict the number of decimal positions that the user interface displays, select the Maximum Precision option to display all positions. You can optionally assign a label such as Seconds, tps, KB, EUR or \$ to attributes of this type and specify whether this label is to appear as a prefix or a suffix when the attribute's value is displayed in a user interface.
	Note: The underlying data type for this kind of attribute is a Java double.
String	Holds a string of text. When this type of string is displayed in a CentraSite user interface, it is displayed in a single-line text box. If a

Data Type	Description
	<p>value exceeds the width of the box, the excess characters are simply not displayed.</p> <p>The Internationalized option allows you to create a String attribute that holds different values for different locales. In CentraSite Control, for example, if a user logs on to CentraSite in an English locale and he or she assigns a value to an Internationalized String attribute, that value is visible to other users with English locales. If a user in a German locale were to view the attribute, the attribute would appear empty because it has no value for the German locale. If the German-locale user were to subsequently assign a value to the attribute, the attribute would then have two String values: one in English and one in German.</p> <p>When CentraSite Control displays an Internationalized String, it displays the value associated with the user's current locale. In the example described above, it would show the English value to users with English locales and the German value to users in German locales. Users in other locales would see an empty attribute until a value for their locale had been assigned to the attribute.</p> <p>The Enumeration option allows you to specify a list of allowed values for the attribute.</p>
URL/URI	<p>Holds a URL/URI. This type of attribute only accepts values in the form:</p> <p><i>protocol://host/ path</i></p> <p>Where:</p> <ul style="list-style-type: none"> ■ <i>protocol</i> is any protocol that java.net.URL supports ■ <i>host</i> is the name or IP address of a host machine ■ <i>path</i> (optional) is the path to the requested resource on the specified host

Besides basic data types such as String, Number, and Boolean, CentraSite supports the following special types:

Attribute Type	Description
Classification	Holds references to one or more categories in a specified taxonomy. You use this type of attribute to classify assets according to a specified taxonomy.

Attribute Type	Description
Relationship	Holds references to other registry objects. You use this type of attribute to express a relationship between an asset and another object in the registry.
File	Holds references to one or more documents that reside in CentraSite's supporting document library or at a specified URL. You can use this type of attribute to attach documents such as programming guides, sample code, and other types of files to an asset.

The inclusion of these attribute types facilitate many of the advanced features in CentraSite.

Attribute Names

An attribute that is one of the following types has two names associated with it: a *display name* and a *schema name*.

Boolean

Date

Duration

Email

Multiline String

IP Address

Number

String

URL

The display name for these types of attributes is the name that is displayed by the CentraSite Control, CentraSite Business UI, and CentraSite plug-in for Eclipse UI. An attribute's display name can consist of any combination of characters, including spaces.

The following are all valid display names:

```
Business Owner  
Amount (in $)  
Numéro de téléphone  
Avg. Invocations/Minute  
1099 Code
```

You can change an attribute's display name at any time.

The attribute's schema name is the name that CentraSite actually gives to the underlying JAXR-based slot that represents the attribute in the registry. This name must be NCName-conformant, meaning that:

- The name must begin with a letter or the underscore character (_).
- The remainder of the name can contain any combination of letters, digits, or the following characters: . - _ (that is, period, dash, or underscore). It can also contain combining characters and extender characters (for example, diacriticals).
- The name cannot contain any spaces.

If you do not specify a schema name for an attribute, CentraSite automatically generates a default schema name based on the attribute's display name. It does this by taking the attribute's display name and replacing any spaces in the name with underscore characters (_) and by removing any invalid character in the name.

If you explicitly specify a schema name that is not NCName-conformant, CentraSite will request that you change it to an NCName-conformant name.

The following table describes the default schema names that CentraSite would generate for the display names shown above:

For this Display Name...	CentraSite would generate this schema name...	The resulting schema name is...
Business Owner	Business_Owner	<i>Valid.</i> You do not have to change the schema name.
Amount (in \$)	Amount_in_	<i>Valid.</i> You do not have to change the schema name.
Numéro de téléphone	Numéro_de_téléphone	<i>Valid.</i> You do not have to change the schema name.
Avg. Invocations Minute	Avg._Invocations_Minute	<i>Valid.</i> You do not have to change the schema name.
1099 Code	_099_Code	<i>Valid.</i> You do not have to change the schema name.

Note:

An attribute's schema name must be unique within the type (that is, two attributes in a type cannot have the same schema name).

After the attribute is created, you can no longer change its schema name.

Profiles

When a type defines an asset (that is, if it is an asset type), the attributes that make up the type are assigned to profiles. Profiles determine how the type's attributes are grouped and presented when an instance of that type is displayed in CentraSite Control and CentraSite Business UI. In CentraSite Control, for example, the attributes associated with a particular profile are grouped together on a tab.

When you define an asset type, you specify the profiles on which its attributes are to be displayed. CentraSite does not require an attribute to be assigned to a profile. If you do not assign an attribute to a profile, the attribute does not be visible in the user interface. However, the attribute resides in the type definition. You can assign an attribute to multiple profiles if you want it to appear on multiple profiles (tabs) in the user interface.

You can define any number of profiles for an asset type. You can specify the order in which you want the profiles to appear when an instance of the type is displayed. You can also specify the order in which attributes are to be displayed within each profile.

Generic Profiles

In addition to the profiles that you define, CentraSite provides several predefined profiles, called *generic profiles*, which you can optionally include in an asset type.

The information on the generic profiles is generated by CentraSite. You cannot customize the content of these profiles or add attributes to them. You can, however, select which of these profiles you want CentraSite to include when it displays an asset of a defined type.

The following table describes the generic profiles that are available in CentraSite Control and CentraSite Business UI:

Profile	Description	CentraSite Control	CentraSite Business UI
Summary	<p><i>Applicable for Service, XML Schema, Application Server, BPEL Process, Interface and Operation types.</i></p> <p>For an instance of Service type, it displays the list of operations and bindings that the asset provides. For an instance of BPEL Process, Interface, or Operation, it displays the basic information such as the asset owner, but includes no type-specific attributes.</p>	✓	
Basic Information	Displays the general information about the asset, such as the asset's version, last modified date, asset type, owning organization, owning user, a description of the asset, and the number of watchers and consumers.		✓
Consumers	Displays the list of users, applications, and arbitrary assets that are registered to consume the asset.	✓	
Advanced Information	Displays additional information about the asset. The profile functions as a high-level container for a set of profiles defined in the asset's type definition. Beginning with CentraSite 9.8, the		

Profile	Description	CentraSite Control	CentraSite Business UI
	Advanced Information profile does not be visible in the user interface.		
Permissions	Displays the asset's instance-level permissions. It displays the controls for modifying the asset's instance-level permissions. ✓		
	To view all of the instance-level permissions for an asset, a user must have Modify or Full permissions on the asset. To edit the instance-level permissions for an asset, a user must have Full permissions on the asset. If a user has only View permission on an asset, the Permissions profile includes only that particular user's permissions for the asset.		
Versions	Displays the versioning history for the asset and provides links to earlier versions and revisions of the asset. It displays the controls for generating a new version of the asset, purging older versions of the asset and reverting to a previous version of an asset. ✓		
Subscriptions	Displays the list of users that are registered to receive notifications when changes are made to the asset. ✓		
Audit Log	Displays the update activity associated with the asset. It displays each change that has been made to an asset (including changes in an asset's lifecycle state) and identifies the user that made the change. ✓		
Object-Specific Properties	Displays the list of object-specific properties assigned to the asset. An object-specific property includes a key that identifies the name of the property and an optional String value, that contains the data associated with the property. (A property's value can be null.) ✓		
	Object-specific properties are used to hold information about an instance of asset when there is no predefined attribute to hold that data. Typically, they are used in one-off situations to attach ad-hoc data to an instance of an asset. For example, if you were managing a certification effort, you might use an object-specific property		

Profile	Description	CentraSite Control	CentraSite Business UI
	to identify the set of assets that required certification.		
Identification	<i>Applicable for Application type.</i>	✓	✓
	Displays the list of consume identifier tokens for accessing a Virtual Service (API).		
Identification (for API Key assets)	<i>Applicable for API Key type.</i> Displays the API key string and its expiration date.		✓
OAuth2 Identification Details (for OAuth Client assets)	<i>Applicable for OAuth2 Client type.</i> Displays the OAuth2 client details (Client ID, Client Secret, Client Name, Scope, and Refresh Token).		✓
Specification	Displays the list of external documents such as Functional Requirements, Error Messages, Release Notes and so forth that are attached to the asset.	✓	✓
Classifications	Displays the list of categories that are used to classify the asset. It displays controls for adding ad hoc classifiers to the asset.	✓	
Associations	Displays the list of objects that are related to the asset. It displays controls for establishing ad hoc relationships between an asset and other registry objects.	✓	
External Links	Displays the list of links to external documents and files that are attached to the asset. It displays controls for attaching documents and files to an asset.	✓	
Policies	Displays the list of design/change-time policies and run-time policies that are applicable to the asset (that is, it includes all the policies whose scope encompasses the displayed asset).	✓	✓

Note:

When you disable the **Policies** profile in Service asset type definition, CentraSite removes the profile from instances of the type Service, but continues to display the profile in instances of the type Virtual Service.

Profile	Description	CentraSite Control	CentraSite Business UI
Performance	<p>Displays the run-time performance metrics captured for the asset. If you are using webMethods Mediator, webMethods Insight, or another run-time monitoring component to log performance metrics for an asset, CentraSite includes those metrics on this profile.</p> <p>Note: When you disable the Performance profile in Service asset type definition, CentraSite removes the profile from instances of the type Service, but continues to display the profile for instances of Virtual Service type and its variants.</p>	✓	
Events	<p>Displays the run-time events associated with an asset. If you are using webMethods Mediator, webMethods Insight, or another run-time monitoring component to log run-time events for an asset, CentraSite displays those events on this profile.</p> <p>Note: When you disable the Events profile in Service asset type definition, CentraSite removes the profile from instances of the type Service, but continues to display the profile for instances of Virtual Service type and its variants.</p>	✓	
Processing Steps	<p><i>Applicable for Virtual Service, Virtual XML Service, and Virtual REST Service types.</i></p> <p>For an instance of Virtual Service type, Displays the protocol (HTTP, HTTPS, or JMS) and SOAP format (1.1 or 1.2) of the requests that the service will accept.</p> <p>For an instance of Virtual REST Service or Virtual XML Service type, Displays the protocol (HTTP or HTTPS) of the requests that the service will accept. Also, you specify the HTTP methods (GET, POST, PUT, DELETE or Use Context Variable) that are supported by the native service.</p> <p>It displays the request routing methods and protocol for authenticating the requests.</p>	✓	

Profile	Description	CentraSite Control	CentraSite Business UI
Deployment	<p data-bbox="342 327 959 394"><i>Applicable for Virtual Service, Virtual XML Service, and Virtual REST Service types.</i></p> <p data-bbox="342 422 959 527">Displays the list of virtual services that are ready for deploying in the webMethods Mediator gateway.</p>	✓	
Run-Time Metrics	<p data-bbox="342 554 959 758">Displays the run-time performance metrics that are available for the asset. If you are using webMethods Mediator, webMethods Insight, or another run-time monitoring component to log performance metrics for an asset, CentraSite includes those metrics on this profile.</p> <div data-bbox="342 779 959 999" style="background-color: #f0f0f0; padding: 5px;"> <p data-bbox="342 789 423 814">Note:</p> <p data-bbox="342 825 943 989">When you disable the Run-Time Metrics profile in Service asset type definition, CentraSite removes the profile from instances of the Service type, but continues to display the profile in instances of the Virtual Service type.</p> </div>		✓
Run-Time Events	<p data-bbox="342 1031 959 1234">Displays the run-time events that are available with an asset. If you are using webMethods Mediator, webMethods Insight, or another run-time monitoring component to log run-time events for an asset, CentraSite displays those events on this profile.</p> <div data-bbox="342 1255 959 1476" style="background-color: #f0f0f0; padding: 5px;"> <p data-bbox="342 1266 423 1291">Note:</p> <p data-bbox="342 1302 943 1472">When you disable the Run-Time Events profile in Service type definition, CentraSite removes the profile from instances of the type Service, but continues to display the profile in instances of the type Virtual Service.</p> </div>		✓
API Gateway Information	<p data-bbox="342 1507 959 1575"><i>Applicable for Virtual Service, Virtual REST Service, and Virtual OData types.</i></p> <p data-bbox="342 1602 959 1669">Displays the following details of the API Gateway instance(s) to which the API is published:</p> <ul data-bbox="342 1696 959 1797" style="list-style-type: none"> <li data-bbox="342 1696 959 1797">■ Name of the API Gateway instance along with a deep link to open the API gateway instance directly in the CentraSite user interface. 		✓

Profile	Description	CentraSite Control	CentraSite Business UI
	<ul style="list-style-type: none"> ■ Status of the API in API Gateway. The  icon indicates that the API is active and the  icon indicates that the API is inactive in the API Gateway instance. <p>Note: The status of APIs are synchronized based your configuration in the centrasite.xml file. For information, see “Configuring the API Gateway Synchronization Settings” on page 1478.</p> <ul style="list-style-type: none"> ■ View in API Gateway link. Click this link to view the API in the corresponding gateway instance. <p>Note: The View in API Gateway link is enabled, only when the API Gateway instance is reachable.</p> <ul style="list-style-type: none"> ■ View Runtime enforcement in CentraSite link. Click this link to view the policy enforcement configuration in a pop-up window. You can click the  icon in the pop-up window to view the parameter configuration values of the policy. <p>Note: This link is displayed only if the CentraSite run-time setting is enabled. For information on enabling run-time aspects from CentraSite, see “Enabling CentraSite Run-Time Aspects” on page 951.</p> <ul style="list-style-type: none"> ■ The list of effective policies applied to the API. Select a policy or level from the drop-down field of the required API Gateway instance. You can view either of the following details in the particular gateway instance: <ul style="list-style-type: none"> API-level, Resource-level, or Method-level policies for the REST APIs. <OR> 		

Profile	Description	CentraSite Control	CentraSite Business UI
	<p>API level or Operation-level policies for the SOAP based APIs.</p> <p>By default, API-level is selected in the drop-down field. This drop-down field is disabled, if the API Gateway instance is not reachable.</p> <p>You can click the  icon next to a policy to view its parameter configuration values.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: If there are more than three API Gateway instances for an API, then you can scroll the page horizontally using the right arrow and left arrow buttons to navigate between the instances.</p> </div>		
API Portal Information	<p><i>Applicable for API Key, Service , REST Service, and OData Service types.</i></p> <p>Displays the maturity status, grouping, subscription terms, and the access token types for the API.</p>		✓
API Key Scope	<p><i>Applicable for API Key, Virtual Service, Virtual REST Service, and Virtual OData Service types.</i></p> <p>Displays the name of the Native Service (API) that is associated to the API key. To view details of the Native API, click the hyperlinked name.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: CentraSite displays the API Key Scope profile in an instance of virtual API, only if the profile is included in the API Key type definition.</p> </div>		✓
ARIS Properties	<p><i>Applicable for Process type.</i></p> <p>Displays the ARIS-specific attributes that are of use when CentraSite is integrated with the ARIS products.</p>	✓	✓

Computed Profiles

CentraSite offers you the possibility to add computed profiles into asset type definitions; this gives you the option to define your own profile, which means that you can implement your own algorithms for calculating the values you wish to represent. You could for example aggregate or compute attribute information from embedded or linked objects.

You can combine the attribute specific profile and the generic profiles layout concept in a single computed profile.

Computed profiles let you create your own layout by using a UI Rendering Concept. You can also specify your own rendering logic to display the computed values. You could, for example, create a custom display of performance metrics as a graphic or an animation.

A computed profile can be implemented as a Java plug-in. For a Java-based plug-in for a computed profile, you create an archive file that contains the plug-in definition and you load the archive file through CentraSite Control into the repository.

After you have added a computed profile into the asset type definition, you can perform administration tasks on the computed profile in the same way as for normal profiles. For example, you can define profile-based permissions and you can define the order of the computed profile relative to the other profiles in the asset detail display.

In addition to the profiles that you define, CentraSite provides several predefined computed profiles.

Computed Profile	Description
API Gateway Communication	This is applicable for instances of API Gateway type. Displays the communication details that are of use for CentraSite to send data to API Gateway. The details include the URL of the API Gateway, the username and password credentials of a CentraSite user or a technical user of API Gateway, and the sandbox category that is used to classify the API Gateway instance.
API Portal Communication	This is applicable for instances of API Portal type. Displays the communication details that are of use for CentraSite to send data to API Portal. The details include the URL of the API Portal, the name of the API Portal's tenant, the username and password credentials of a CentraSite user or a technical user of API Portal, and the sandbox category that is used to classify the API Portal instance.
Assets	This is applicable for instances of Organization type. Displays the list of assets owned by the user.
CentraSite Communication	This is applicable for instances of API Gateway, API Portal, and Mediator types. Displays the communication details that are of use for API Gateway, API Portal, or Mediator to exchange data with CentraSite. The details

Computed Profile Description

include the URL of the Software AG Runtime and the username and password credentials of a CentraSite user.

Child Organizations

This is applicable for instances of Organization type.

Displays the general information about the organization, such as the organization's name, the description of the organization, the contact person for the organization and the address of the organization's web site.

Consumer Overview

This is applicable for instances of Service types.

Displays the list of all virtual endpoints defined for the API in API Gateway.

SOAP API. Displays the Consumer Service WSDL / WSDL URL, and the list of Access URIs and API keys for the SOAP API. In addition, if the SOAP API has **Enable REST Support** policy action applied during virtualization, this profile also displays the REST URL variant of the Virtual (SOAP) API.

REST API. Displays the list of Access URIs and API keys for the REST API.

Note:

The list of API endpoints are synchronized based your configuration in the **centrasite.xml** file. For information, see [“Configuring the API Gateway Synchronization Settings” on page 1478.](#)

Groups

This is applicable for instances of Organization type.

Displays the basic information about the group, such as the owning organization and a description of the group.

Group Members

This is applicable for instances of Organization type.

Displays the list of users who belong to the group.

Mediator Communication

This is applicable for instances of Mediator asset type.

Displays the communication details that are of use for CentraSite to send data to Mediator. The details include URL of the Mediator, and the username and password credentials of an Integration Server user.

Permissions

This is applicable for instances of Organization type.

Displays the list of permissions assigned to the role.

Provider Overview This is applicable for instances of Service types.

Computed Profile	Description
	<p>Displays the list of native and virtual endpoints defined for the asset in API Gateway. In this profile, a native endpoint is represented by the Binding, and a virtual endpoint is represented as an Alias that identifies a specific Access URI (address where the virtual endpoint is published). It displays details of the API keys, and controls for performing various functions (renew, revoke and delete) API keys.</p>
	<p>Note: The list of API endpoints are synchronized based your configuration in the centrasite.xml file. For information, see “Configuring the API Gateway Synchronization Settings” on page 1478.</p>
Published APIs	<p>This is applicable for instances of API Gateway and API Portal types.</p> <p>Displays the details of the APIs that were published to API Gateway and API Portal gateways. The details include the name of the API, the description of the API, the version identifier of the API, and a deep link to open the API details page directly in the API Gateway and API Portal user interface. The status of the published APIs will be same their status in API Gateway.</p>
Resources and Methods	<p>This is applicable for instances of REST Service and Virtual REST Service types.</p> <p>Displays the list of resources and methods defined for the REST-based API.</p>
OData Resources	<p>This is applicable for instances of OData Service and Virtual OData Service types.</p> <p>Displays the list of all first level resources. It displays the resource path, entity type, resource parameters, HTTP methods and outgoing navigation properties for the OData service.</p>
Roles	<p>This is applicable for instances of Organization type.</p> <p>Displays the basic information about the role, such as the owning organization and a description of the role.</p>
Runtime Metrics	<p>This is applicable for instances of Service types.</p> <p>Displays the run-time performance metrics that are available for the asset. If you are using webMethods Mediator, webMethods Insight, or another run-time monitoring component to log performance metrics for an asset, CentraSite includes those metrics on this profile.</p>

Note:

Computed Profile Description

When you disable the **Run-Time Metrics** profile in Service asset type definition, CentraSite removes the profile from instances of the type Service, but continues to display the profile in instances of the type Virtual Service.

Runtime Events This is applicable for instances of Service types.

Displays the run-time events that are available with an asset. If you are using webMethods Mediator, webMethods Insight, or another run-time monitoring component to log run-time events for an asset, CentraSite displays those events on this profile.

Note:

When you disable the **Run-Time Events** profile in Service asset type definition, CentraSite removes the profile from instances of the type Service, but continues to display the profile in instances of the type Virtual Service.

Technical Details This is applicable for instances of Service types.

Displays the technical information about an asset. For a SOAP-based asset, this profile includes the WSDL URL and a list of the operations and bindings. For an XML/REST-based asset, the profile includes the schema URL and a list of the resources.

Users This is applicable for instances of Organization type.

Displays the basic information about the user, such as the display name, user ID, email address, owning organization, a description of the user, telephone and mobile numbers.

Assigning Permissions on Profiles

You can restrict access to individual profiles by setting profile permissions on an instance of an asset. Doing this enables you to control who can view and edit the attribute values on a particular instance of a profile.

Important:

Profile permissions restrict access at the UI level but not at the API level. At the API level, profile permissions are irrelevant. A user with a view permission on an asset can access the asset's metadata through the API, regardless of whether profile permissions exist for the asset.

Classification of Types

CentraSite includes a set of types that are classified in the following categories:

- Predefined asset types
- Custom asset types

- Composite asset types
- Association types

Predefined Asset Types

CentraSite is installed with a number of predefined types. Some of these types are core types that belong to CentraSite itself. You can modify these types.

Other predefined types are installed to support the use of CentraSite by products such as the webMethods Product Suite. These types belong to other products, which expect the type definitions to remain unchanged. Modifying or deleting these types in CentraSite can lead to inconsistencies or errors in the product that uses the type. For example, if you modify or delete a type that is used by the webMethods Product Suite, components such as the webMethods Integration Server may no longer be able to publish assets to CentraSite. You must not modify these predefined asset types.

The following table identifies the predefined asset types that are installed with CentraSite and indicates to which product they belong.

Type Name	Owner
Alias	CentraSite
API-Key (Virtual type of Application)	CentraSite
API Gateway (Virtual type of Gateway)	CentraSite
API Portal (Virtual type of Gateway)	CentraSite
Application	CentraSite
Application Server	CentraSite
ApplinX Application	webMethods Product Suite
ApplinX External Web Operation	webMethods Product Suite
ApplinX External Web Service	webMethods Product Suite
ApplinX Flow Procedure	webMethods Product Suite
ApplinX Path Procedure	webMethods Product Suite
ApplinX Procedure Group	webMethods Product Suite
ApplinX Program Procedure	webMethods Product Suite
ApplinX Screen	webMethods Product Suite
ApplinX Screen Group	webMethods Product Suite
ApplinX Server	webMethods Product Suite

Type Name	Owner
BPEL Partner	CentraSite
BPEL Partner Link	CentraSite
BPEL Partner Link Type	CentraSite
BPEL Process	CentraSite
BPEL Role	CentraSite
BPM Package (Virtual type of Package)	webMethods Product Suite
BPM Process Project	webMethods Product Suite
CAF Security Role	webMethods Product Suite
CAF Task Rule	webMethods Product Suite
CAF Task Type	webMethods Product Suite
Decision Entity	webMethods Product Suite
E-form	webMethods Product Suite
Endpoint Alias (Virtual type of Alias)	CentraSite
Event Type	CentraSite
Gateway	CentraSite
Gateway Application	CentraSite
Interface	CentraSite
IS Connection	webMethods Product Suite
IS Package	webMethods Product Suite
IS Routing Rule	webMethods Product Suite
IS Server	webMethods Product Suite
IS Service	webMethods Product Suite
IS Service Interface	webMethods Product Suite
IS Specification	webMethods Product Suite
IS Type Definition	webMethods Product Suite
JDBC Datasource	webMethods Product Suite
Mediator (Virtual type of Gateway)	CentraSite
OAuth2 Client (Virtual type of Application)	CentraSite

Type Name	Owner
OData Service	CentraSite
Operation	CentraSite
Organization	CentraSite
Package	CentraSite
Portlet	webMethods Product Suite
Portlet Preference	webMethods Product Suite
Process	webMethods Product Suite
Process Pool	webMethods Product Suite
Process Step	webMethods Product Suite
Process Swimlane	webMethods Product Suite
REST Method (Virtual type of Operation)	CentraSite
REST Parameter	CentraSite
REST Payload	CentraSite
REST Resource	CentraSite
REST Service	CentraSite
Rule Action	webMethods Product Suite
Rule Data Model	webMethods Product Suite
Rule Event Model	webMethods Product Suite
Rule Parameter	webMethods Product Suite
Rule Project	webMethods Product Suite
Rule Set	webMethods Product Suite
Scheduled Report	CentraSite
Secure Alias (Virtual type of Alias)	CentraSite
Service	CentraSite
Simple Alias (Virtual type of Alias)	CentraSite
SmartList (Virtual type of Package)	CentraSite
TN Document Type	webMethods Product Suite
TN Group	webMethods Product Suite

Type Name	Owner
Transformation Alias	CentraSite
User	CentraSite
Virtual type OData Service (Virtual type of OData Service)	CentraSite
Virtual type REST Service (Virtual type of REST Service)	CentraSite
Virtual type Service (Virtual type of Service)	CentraSite
Virtual type XML Service (Virtual type of XML Service)	CentraSite
Web Application	webMethods Product Suite
Web Application Page	webMethods Product Suite
WS-Policy	CentraSite
XML Schema	CentraSite
XML Service	CentraSite

Predefined Virtual type Asset Types

Certain predefined types installed with CentraSite are classified as Virtual type types. A Virtual type type has the same set of attributes as its base type, but has its own set of profiles and properties, and adds its own behavior. A Virtual type type can also have its own lifecycle model and policies.

Virtual type types do not have a separate storage structure or a schema, so the instances of a Virtual type type are stored as regular assets. For example, a Virtual type Service asset inherits the same set of attributes as its base type, Service, and adds its own behavior, yet it is stored as a Service asset.

A Virtual type type inherits all of its attributes from its base type. Therefore, you cannot add attributes directly to a Virtual type type. To add new attributes to a Virtual type type, you add the attributes to the base type. You can selectively display these attributes on the profiles that you have defined in the type. Similarly, you cannot delete attributes from or edit the properties of attributes in the Virtual type type. All attribute creation, deletion, and definition is performed on the base type and those changes are applied to all of its Virtual type types.

A Virtual type type has its own set of **Advanced Settings**, which enables you to configure the following properties for a Virtual type type:

- Large and small icons
- Visible in asset browse
- Enable reports
- Policies can be applied
- Require consumer registration

- Enable versioning
- Top level type
- Enable lifecycle management
- Visible in search
- Inherit base type profiles
- Inherit base type policies
- Inherit base type LCMs
- Clone base type profiles

Additionally, the Virtual type type has an **Inherit Base Type** option, which determines whether the profiles, LCMs, and policies of the base type also apply to the Virtual type type. You can enable or disable this option for each Virtual type type.

Customizing Predefined Asset Types

Before using the predefined asset types in your environment, you should examine their type definitions and customize them as necessary.

With respect to customizing the predefined asset types installed with CentraSite, you can:

- Modify the asset type's existing properties and options (other than **Schema Name, Namespace,** and the **Base Type** (for Virtual type types only)).

Note:

Although CentraSite allows you to change the display name of the predefined types, Software AG recommends that you do not do this. Name changes may lead to problems with future upgrades of CentraSite.

- Add custom attributes to any asset type other than the predefined Virtual type types.
- Move certain attributes from one profile to another.
- Specify which profiles are to be displayed for the asset type.
- Change the type's system-property settings (for example, specify whether the type supports versioning or can be used with design/change-time policies)
- Add profiles, modify profiles, delete profiles, and rearrange the order of profiles within the asset type.
- You *cannot* delete any of the predefined attributes that belong to the type. You can, however, delete custom (that is, user-defined) attributes that belong to the type.
- You *cannot* modify the inherited profiles and attributes of the predefined Virtual type types.

Custom Asset Types

Besides customizing the predefined asset types that are installed with CentraSite, you can also define custom types of your own. For example, if you wanted to include items such as service requests, IT projects, and source code libraries in your registry, you would create a custom type for each of these entities.

Note:

Before creating a custom type, always check to see whether CentraSite provides a predefined type that you might be able to customize and use. Customizing one of CentraSite's predefined types will save you time, especially if the type requires a file importer.

Before creating a custom type, you must first decide which aspects of an entity you want to model in the registry. If you were creating a type to represent IT projects, for example, you might want to capture characteristics such as the name of the project requester, the lines of business the project is expected to affect, the project plan, the project manager and the project's expected completion date. After you decide which specific characteristics and qualities you want to model, you can create a custom type that includes a corresponding *attribute* for each of those characteristics or qualities.

Project		
Release Date	DateAndTime	Current target date for release.
Business Owner	String	Functional unit for which the project is being performed.
Status Reports	File	Weekly reports on project status.
Project Plan	File	Current project plan.
Managed By	Relationship	Project Manager(s)

Attributes for asset type "Project"

Note:

A custom type that you add to CentraSite is treated as an *asset type* (that is, instances of that type are treated as *assets*).

Composite Asset Types

Certain assets can be stored in CentraSite as a set of related registry objects. Such assets are called composite assets. For example, if a web service provides several operations, this is stored in CentraSite as a composite asset consisting of the Service asset plus a separate Operation object for each of the web service's operations.

The objects that are constituents of a composite asset are referred to as components. In a composite asset there is a root component and one or more sub-components that are related to the root component. In the above example, the Service asset is the root component and the Operation objects are the sub-components. A sub-component of a composite asset can itself be a composite asset.

Depending on the relationships defined, registry operations (such as deleting an asset or exporting an asset) performed on a component of a composite asset can cause the same operation to be performed automatically on other components of the composite asset.

The concept of relationships between different objects in a SOA environment follows the UML idea of association relationships. This is only one of several forms of relationship supported by UML, but most SOA Registry Repositories only offer this form. CentraSite extends this scope to provide aggregation and composition relationships in addition to the existing association relationships. Each of these relationship forms provides its own semantics that affect specific operations that can be performed on composite assets.

You can define composite assets for all asset types, including custom (that is, user-defined) asset types.

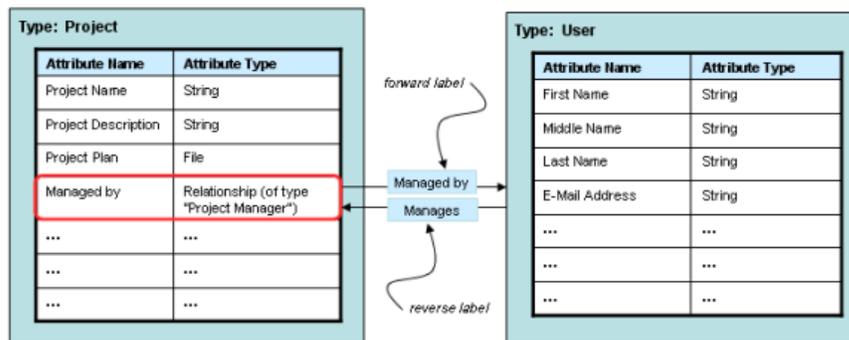
Association Types

An association type describes a type of relationship that can exist between objects in the registry.

An association type has a name, a forward label (which describes the relationship of the source object to a target object) and an optional reverse label (which describes the relationship of the target object to the source object).

You use association types to define Relationship attributes in an asset type. In the following example, a Relationship attribute called Managed By has been included in the Project asset type to associate a project asset with the user that manages the project.

You Use Association Types to Define Relationship Attributes in Object Types



When users publish assets to the registry, there are two ways in which they can relate an asset with other objects in the registry.

- **By establishing the relationship using an asset's Relationship attributes.** If an asset's type includes one or more Relationship attributes, users can relate an asset to other objects in the registry by simply setting these attributes.
- **By establishing an ad-hoc association using the asset's Associations profile.** If an asset's type includes the Associations profile, users can relate assets of that type with other objects on an ad hoc basis. Using this profile, users can relate an asset to Virtual typely any other object in the registry (assuming they have View permission on the target object).

When you include Relationship attributes in an asset type, you not only enable users to specify the objects to which an asset is related, you enable the relationships to be discovered and reported by the Asset Navigator feature.

Like asset types, association types are system-wide objects. They apply to all organizations defined in the registry (that is, all organizations within an instance of CentraSite have access to the same global set of association types). You cannot restrict the use of an association type to a specific organization.

Composite Asset Types

The CentraSite data model provides a means of representing composite assets and allows operations to be performed on the entire composite asset or on sub-components in a consistent and well-defined manner.

The following operations take the composition definitions into account:

- Deleting an asset
- Exporting an asset
- Creating a new version of an asset
- Setting the instance permissions on an asset
- Changing the owner of an asset
- Moving an asset to another organization

Note:

Lifecycle state propagation is not included in the above list, as such models can cause major problems in their definition and consistency rules. If such a model is required, then it should be implemented through a custom pre-state change or post-state change policy.

Shared and Nonshared Components

Sometimes a component can serve as a constituent of multiple composite objects. For example, XML schema, ABC, might contain schema, XYZ, as one of its components. Other services and schemas may also include schema XYZ as a component. Components that can belong to more than one composite object are referred to as *shared components*. Components that can only belong to a particular instance of a composite object are referred to as *nonshared components*. For example, the operations, bindings and interfaces associated with a Web service are considered *nonshared* components. These objects belong solely to the service and cannot function as constituents of other composite objects. Schemas, however, are considered *sharable*, meaning that they do not belong exclusively to a particular composite object.

Required Objects

Besides components, a composite object can also have *required objects*. Required objects are registry objects or repository items that are not actually part of the composite object itself, but support or augment the composite object in an essential way. For example, if a Service object has a WS-Policy attachment, the attached policy is treated as a required object because it specifies the WS-Policies that must be applied to the service when it is deployed.

Required objects, while not actually part of the composite object, must be present in the registry to make the object wholly complete or usable. (An asset's required objects are generally objects that the export process must bundle with the asset in order for the asset to be wholly represented and functional in another registry.)

Collectors

A collector is an internal process within CentraSite that identifies all of the constituents of a composite object. A collector examines a given object and returns lists that identify:

- The nonshared components associated with the composite object
- The shared components associated with the composite object
- The required objects associated with the composite object

Each composite type has its own collector. The lists produced by a collector are used by handlers that operate on instances of composite objects. For example, when you delete an XML Schema, the delete handler for schemas deletes the schema itself and all of the schema's nonshared components as identified by the collector for XML Schemas.

Definition of Composite Asset Types

The relationships between components of a composite asset are defined using relationship attributes available in the appropriate asset type definition(s). A relationship can be defined on the root component or on a sub-component.

In addition to using the predefined composite asset types, you can define your own composite asset types. A user-defined composite asset type consists of the following parts:

- a user-defined asset type; each instance of this type is the root component of a composite asset and
- other asset types or object types; instances of these types are related to the root component or to each other by means of relationship attributes.

The definition of a relationship may be changed at any time without affecting any instances.

You set up the associations between the components of a composite asset by using attributes of the data type Relationship in the asset type definition. A relationship indicates a coupling between two objects. A relationship has a direction, meaning that one of the related objects is the source of the relationship and the other object is the target of the relationship. When you define a relationship, you define it on the source object, not on the target object.

The semantic of a relationship is usually indicated by the name you select for the association type of the relationship attribute (for example, hasChild or hasParent). You can think of the association type as a label that does not affect the behavior of the composite asset (technically it is a classification on the relationship attribute), although it makes sense to select meaningful association types for the relationship attributes. To inform CentraSite about the semantics of the associations in your composite asset type, you need to define the relationship attributes.

CentraSite provides several forms of relationship that allow you to define plain relationships between assets as well as relationships for composite assets.

For our purposes, we use terms and concepts introduced by UML as follows:

- **Association.** The loosest form of coupling is provided by the *association* relationship. This is like a cross-reference between two components. It indicates that there is a dependency between the components but no aggregation or composition. In this case, registry operations performed on a component do not cause any operation to be performed automatically on the related component. For example, suppose Asset A contains an association relationship to Asset B and then Asset A is then deleted; in this case, the registry remains in a consistent state without having to delete or modify Asset B in any way.

Note:

When an asset instance has an incoming relationship it may not be deleted until that incoming relationship has been removed or the asset that is the source of the relationship is in the delete set.

- **Aggregation.** A tighter coupling is provided by the *aggregation* relationship. Aggregation is similar to a whole or part relationship in which components of a structure can also exist independently of the structure; this is like the contains semantic, whereby one component contains another component but does not own it. In this case, some operations performed on a component cause the same operation to be performed automatically on the related component. For example, if you want to export an asset, CentraSite automatically extends the export set by adding all of the components that are coupled by aggregation. However, if you want to delete an asset, CentraSite leaves the coupled components unchanged.
- **Composition.** The tightest coupling is provided by the *composition* relationship. Composition is similar to a whole/part relationship in which components of a structure cannot exist independently of the structure; this is like the owns semantic, whereby one component owns another component. In this case, all registry operations performed on a component cause the same operation to be performed automatically on the related components. For example, if you want to delete an asset, then CentraSite automatically extends the delete set by adding all of the components that are coupled by composition.

The form of relationship determines the way in which registry operations performed on one component affect the related components. In the following table, entries marked with Yes mean that an operation on a component causes the same operation to be performed on the related components, whereas table entries marked with No mean that the related components are not changed.

Operation on component	Form of Relationship:		
	Association	Aggregation	Composition
Move asset to another organization	No	No	Yes
Change asset owner	No	No	Yes
Delete asset	No	No	Yes

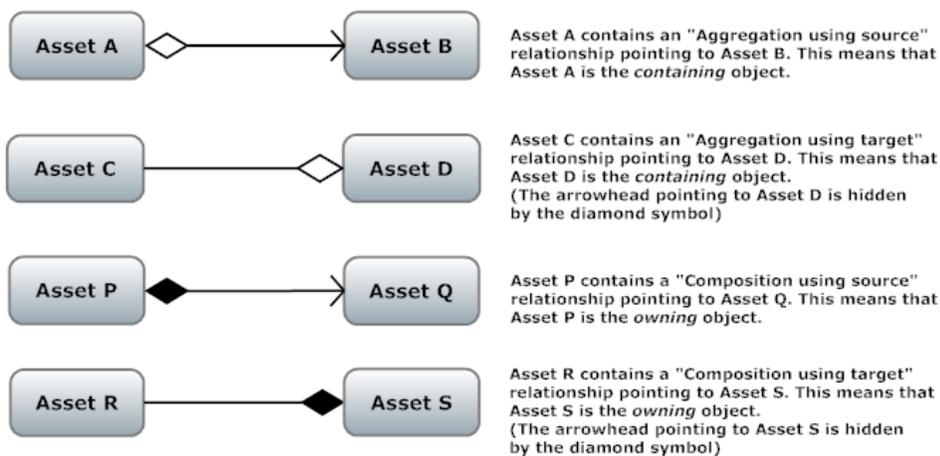
Operation on component	Form of Relationship:		
	Association	Aggregation	Composition
Export asset	No	Yes	Yes
Set instance permissions on an asset	No	Yes	Yes
Create new version of an asset	No	No	Yes

These operations are the primary set which are affected by different forms of relationships and are supported out-of-the-box by CentraSite.

Aggregation and Composition come in two forms, namely with source and with target:

- **Aggregation or Composition with source.** This means that the aggregation or composition treats the source component (that is, the component where the relationship is defined) as the containing or owning component and the target component (that is, the component that the relationship points to) as the contained or owned component.
- **Aggregation or Composition with target.** This means that the aggregation or composition treats the source component (that is, the component where the relationship is defined) as the contained or owned component and the target component (that is, the component that the relationship points to) as the containing or owning component.

You might find the following diagrams useful to illustrate the relationships in composite assets. They are similar to UML diagrams, but allow the aggregation or composition to be on the target component (in UML, they can only be on the source component). The forms with source and with target are represented using a diamond-shaped symbol to indicate the containing or owning component. Aggregation is indicated by a non-filled diamond symbol and Composition is indicated by a filled diamond symbol. An arrow points from the source component to the target component, with the arrowhead located at the target component. If the diamond symbol and the arrowhead are located at the same component, only the diamond is shown.



Semantics of Relationships and Operations

The following sections describe the semantics of relationships and operations that are possible with the types in CentraSite.

Association Relationship

Association relationships are the relationships that were available in later releases of CentraSite version 8 and are available with unchanged semantics in the current release.

Aggregation Relationship

The aggregation relationship changes the rules in the following way for operations:

Operation	Rules
Delete an asset	There are no changes in the delete rules when introducing aggregation.
Create a new version of an asset	<p>There are no changes in the versioning rules when introducing aggregation.</p> <p>However, for assets of type Service and XML Schema, there is an additional possibility: If you select the check box Propagate to dependent objects when you create a new version of the root component of a composite asset of one of these types, the versioning is propagated also to components of these types that are connected to the root component through aggregation relationships.</p>
Set instance permissions on an asset	Merges the permissions of the initiating component with those of the current component. The permissions assigned to the contained component are the union of the permissions of the containing asset and the contained component. If the user that performs the operation does not have Full permissions on a component, then it and all of its sub-components is skipped.
Move asset to another organization	There are no changes in the move organization rules when introducing aggregation.
Change the owner of an asset	There are no changes in the change owner rules when introducing aggregation.
Export an asset	If a component that has a <i>containing</i> aggregation is added to the export set, then the target is also added to the export set. The rules when selecting the check box Including instances in the user interface apply as before with the addition of the containing rules.

Note:

For Export, the usage of recursive relationships on the type and instance level must be taken into account. Whereby type level does not mean that the same instance is referenced.

Composition Relationship

Composition relationships affect all the defined operations to varying degrees.

Operation: Deleting an Asset

On deletion, if the root component is added to the set to be deleted, then the sub-components is added to the set to be deleted. The direction of the association does not play a role in defining the set, only the *containing* designation. This means it is possible for the deletion to fail if one of the assets added to the deletion set during this processing is referenced through the basic association relationship target rules.

This rule is applied recursively. For example, if we have three assets that have the relationships, A contains B contains C, then the following statements apply:

- When A is deleted, then B is deleted and finally because B is deleted, C is deleted.
- When deleting C, only C is deleted.

This fails if the deleting user does not have permission on any of the assets in the set acquired by traversing the graph. The delete is considered atomic - either all are deleted or none. This avoids inconsistencies in the outcome of the operation.

The relationship direction always plays a role in the deletion operation. An asset may not be deleted if it is the target of a relationship and the source is not part of the deletion set.

The deletion rules described here apply also when you purge old versions of an asset. In this case, the purge operation will be applied not only to the component being purged, but also to the related sub-components.

Operation: Creating a New Version of an Asset

On versioning, if the root component is added to the set to be versioned, then the sub-components is added to the set. The direction of the association does not play a role in defining the set, only the *containing* designation.

If you create a new version of an asset that is the root component of a composite asset and the root component is related to one or more of the other components through composition relationships, CentraSite automatically creates a new version of each of these other components.

Operation: Exporting an Asset

On export, if the root component is added to the set to be exported, then the sub-component is added to the set. The direction of the association does not play a role in defining the set, only the *containing* designation.

Operation: Setting Instance Permissions on an Asset

On setting permissions, when the root component is added to the set to which the permissions is applied, then the sub-components asset are also added if and only if the user has the permission to modify the permission of the target. If the user does not have permission, then the graph traversal for the target is not carried further for this sub-graph.

The permission set that is given to all sub-component assets is the merge based on what is to be modified. The permissions assigned to the owned asset are the union of the permissions of the owning asset and the owned asset.

Operation: Moving an Asset to Another Organization

On moving an asset to another organization, when the root component is added to the set to be moved, then the sub-components are also added. No permission checks are done during this operation as only users in the CentraSite Administrator role may perform this operation.

Operation: Changing the Owner of an Asset

On changing ownership, when the root component is added to the set to be changed, then the sub-components are also added to the set. No permission checks are done during this operation as only users in the CentraSite Administrator role may perform this operation.

Extended Rules

The following sections describe additional rules for updating relationships and assets.

Changing Relationships

As part of the support for Aggregation and Composition, CentraSite allows the relationship form to be changed after the type is created. This change affects all current instances and new instances. This means that after a relationship attribute is created, the form (for association the default form, for aggregation (both forms, that is, using source and using target) and for composite (both forms)) can be changed by an Asset Type Administrator. From that point onwards, the appropriate rules is applied when performing the defined operations.

Updating Assets

The following asset updates need to be taken into account when implementing models:

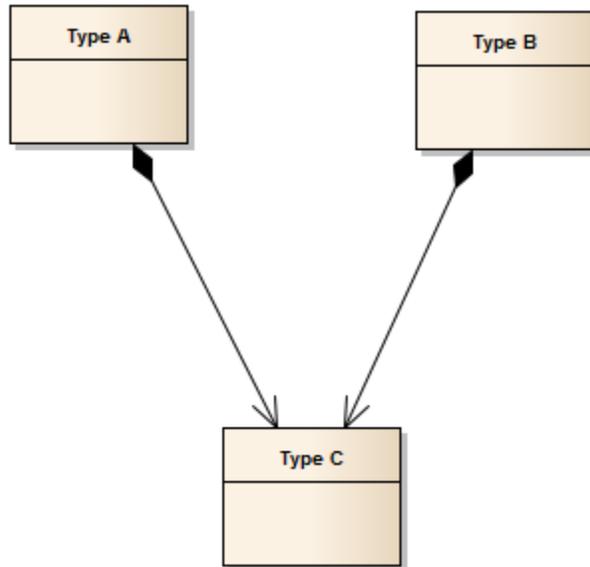
1. Adding relationships to existing instances. Given the below model:



It is perfectly legal to create instances of Type A and Type B independent of one another. In fact in CentraSite this characteristic is mandatory, as the creation of multiple assets at the same time is only allowed in a few places in the UI.

When adding the relationship between an instance of Type A and an instance of Type B, CentraSite does not do any extra operations to guarantee the consistency of permissions at this point.

2. Adding relationships to 2 different composites. Given the below model (which is a legal model):



The following restriction applies to user-defined asset types, but not predefined asset types: At runtime if an instance of Type A creates a composite relationship to an instance of Type C and then an instance of Type B tries to create a composite relationship to the same instance of Type C, this composition is rejected. This is because a contained asset (instance of Type C) can only have one owning asset (instance of Type A or instance of Type B).

Usage Scenarios

The outcome of each operation is given based on a very simple type and instance configuration in each of the usage scenarios.

Unless otherwise stated, the instances that each operation is performed on is:



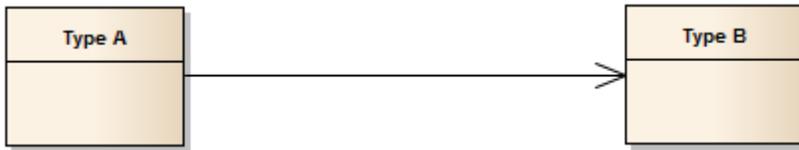
Key	Description
IoA	Instance of type A
IoB	Instance of type B

Delete Usage Scenarios

In the following sections, we describe some delete usage scenarios.

Association Relationship

Given the type model:



The result of the delete operation is:

Operation	Expected result
Delete IoB	Fail because of incoming relationship from IoA
Delete IoA	Success. Post-condition: IoB is left intact

Aggregation Relationship with Containing Constraint on Type A

Given the type model:



The result of the delete operation is:

Operation	Expected result
Delete IoB	Fail because of incoming relationship from IoA
Delete IoA	Success. Post-condition: IoB is left intact

Aggregation Relationship with Containing Constraint on Type B

Given the type model:



The result of the delete operation is:

Operation	Expected result
Delete IoB	Fail because of incoming relationship from IoA

Operation	Expected result
Delete IoA	Success. Post-condition: IoB is left intact

Composition Relationship with Containing Constraint on Type A

Given the type model:



The result of the delete operation is:

Operation	Expected result
Delete IoB	Fail because of incoming relationship from IoA
Delete IoA	Success. Post-condition: IoB is removed

Composition Relationship with Containing Constraint on Type B

Given the type model:



The result of the delete operation is:

Operation	Expected result
Delete IoB	Success. Post-condition: IoA is removed
Delete IoA	Success. Post-condition: IoB is left intact

Composition Relationship with Permission Scenario

Given the type model:



With the constraints:

- User who performs the deletion is Fred
- Fred has Full permission on IoA
- Fred has Read permission on IoB

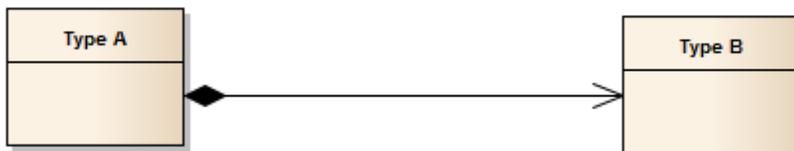
The result of the delete operation is:

Operation	Expected result
Delete IoB	Fail. User Fred does not have full permission on IoB
Delete IoA	Fail. User Fred does not have full permission on IoB

Versioning Usage Scenarios

Versioning for Association Relationship and Aggregation Relationship are the same and do not change from previous versions, therefore only Composition Relationship are shown below.

Given the type model:



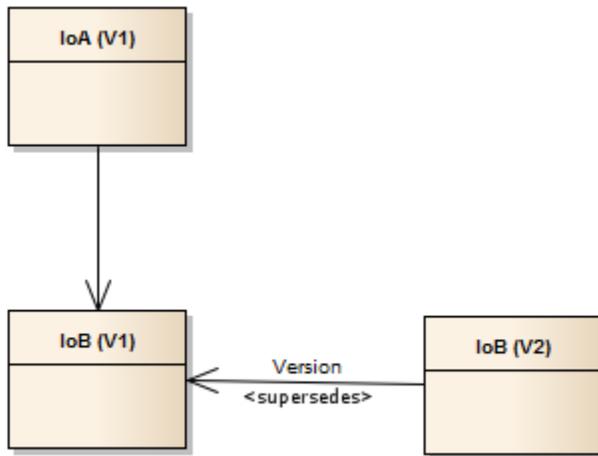
Note:

Both variants of a composite relationship (source and target) are supported and are orthogonal.

Based on the instances given above, the following scenarios are considered relevant.

Versioning of IoB

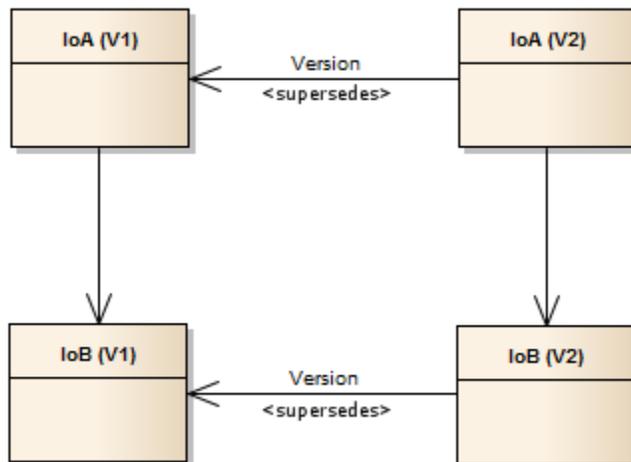
This causes just IoB to be versioned and the IoA version is left unchanged. Pictorially, this looks like:



Versioning of IoA

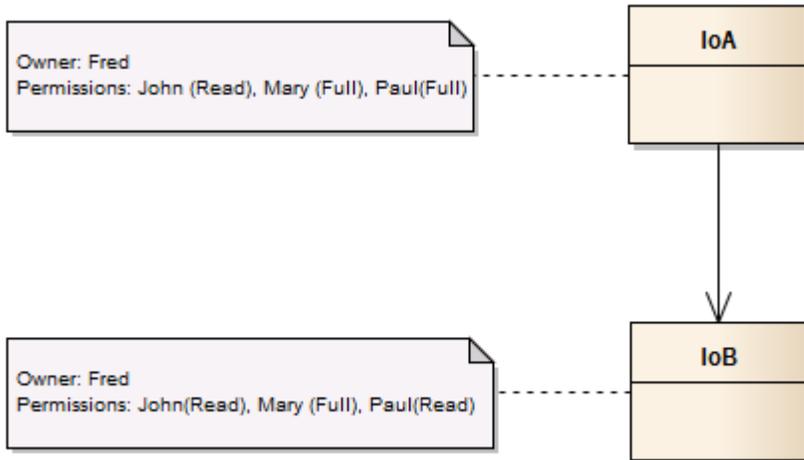
Versioning of IoA will result in the composite relationship being used to work out which other assets should be versioned at the same time. This will result in IoA and IoB being versioned together (if we fail to version either then neither is versioned).

This pictorially looks like:



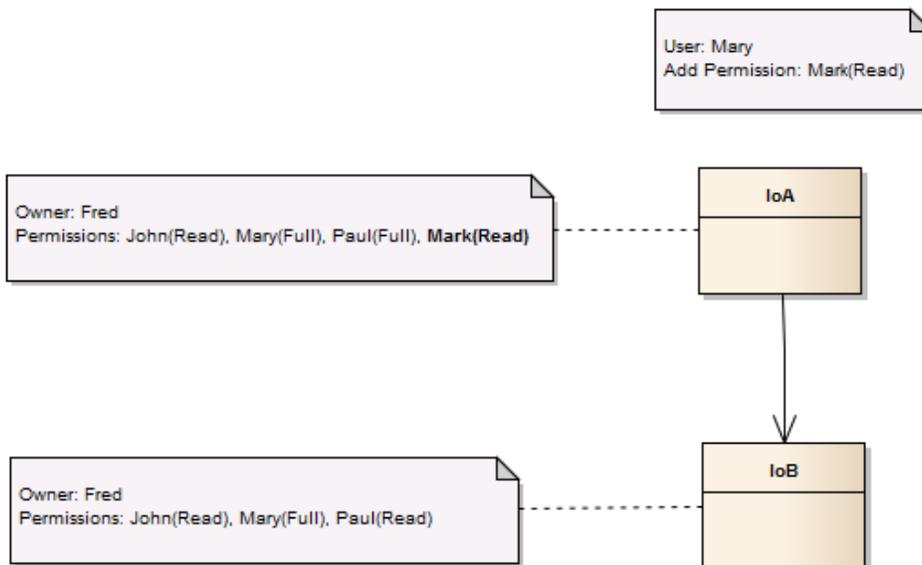
Permission Usage Scenarios

For the permission scenarios, the following instances with the annotations for the owner and permissions is used as basis.



Association Relationship Permission Propagation

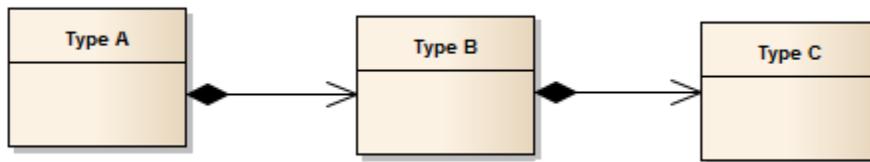
When an Association Relationship is used, the permission propagation does not take place. This means that if an instance permission is set, then only that instance's permission is affected. This means that if user Mary adds Read permission for user Mark on IoA, then Mark only gets permission to Read IoA. He does not get permission to Read IoB. Pictorially this looks like:



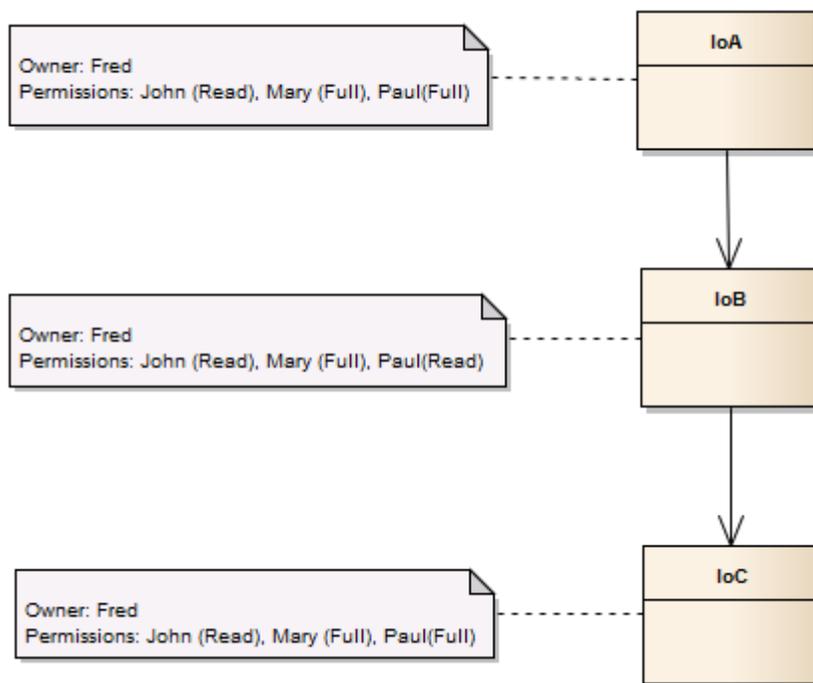
Composition or Aggregation with Weak Propagation

One of the key points of permission propagation is what happens if a user only has a Full permission on a subset of the assets to which the permissions need to be propagated. In this case, the usage of the so-called weak propagation rule comes into effect. The rule states that if a user does not have permission to propagate to all instances in the set, then the permissions is propagated to only the instances which are allowed.

For this scenario the following model is used:

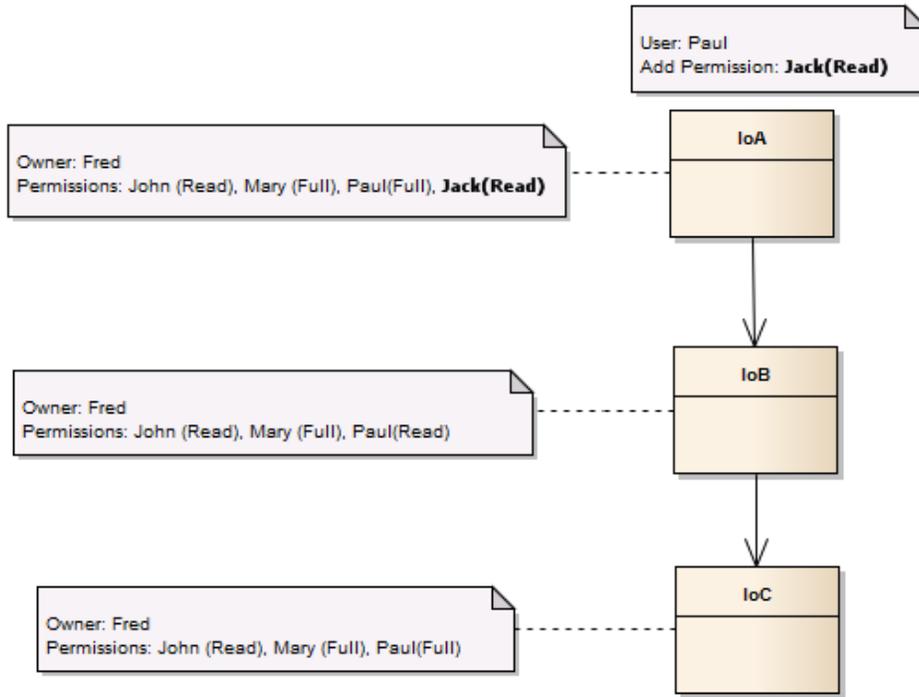


Based on this model, the instances and the permissions to start with should be:



Now if user Paul adds Read permission for user Jack to IoA, Jack will only get this permission on IoA as Paul does not have the rights to give Jack the permissions on IoB. Even though Paul has Full permission on IoC, because it is a child of IoB, Jack does not get the permissions for IoC because of weak propagation. We trim or terminate the propagation at IoB.

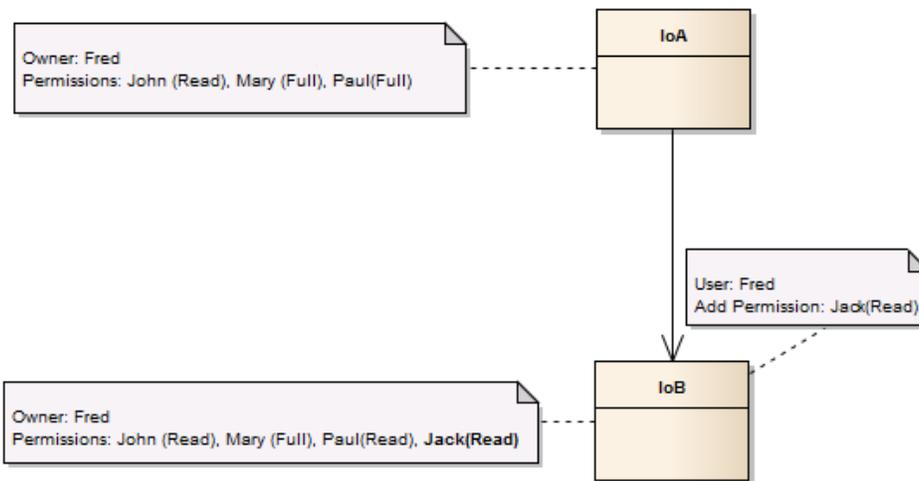
This pictorially looks like:



Composition or Aggregation Relationship Updating Sub-component

Updating the permissions of a sub-component without affecting the overall composition or aggregation is not affected with the changes. Therefore if user Fred wants to explicitly add Read permission for Jack on IoB, this is possible.

This pictorially looks like:

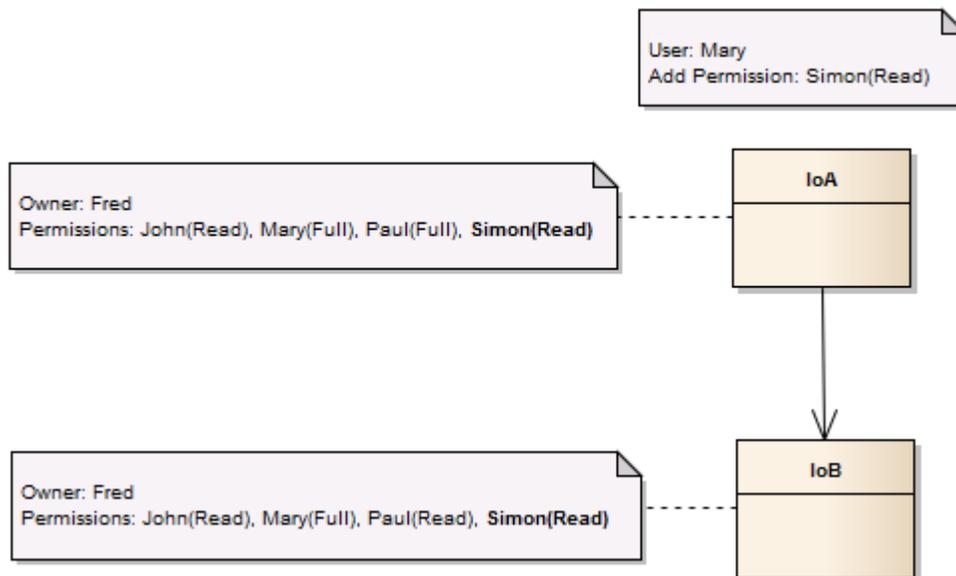


Composition or Aggregation Relationship with Full Propagation

Full propagation happens when all sub-components can be updated by the instigating user. For example, Mary wants to give Read permission to Simon on IoA with permission propagation

through the Composition or aggregation relationship. As Mary has Full permissions on IoA and IoB, the permissions are propagated over the relationship.

This pictorially results in:



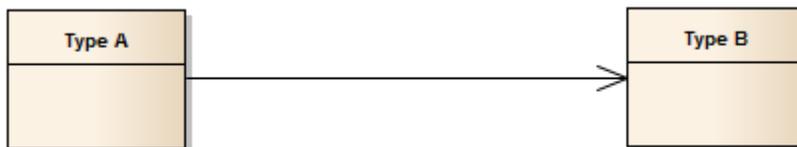
Export Usage Scenarios

In the following sections, we describe some export usage scenarios.

Export with Association Relationship

Export works the same as in previous versions - the usage given here assumes that no additional options are selected.

Given the model:



The result of the Export operation is:

Operation	Expected result
Export IoB	Export set contains IoB. It does not contain IoA
Export IoA	Export set contains IoA. It does not contain IoB

Export with Composition or Aggregation Relationship

For both Composition and Aggregation Relationships, the rules are exactly the same. When such a Relationship with the appropriate containing rule is found, then traverse the relationship and add sub-components to the set.

Given the model:



OR



The result of the Export operation is:

Operation	Expected result
Export IoB	Export set contains IoB. It does not contain IoA
Export IoA	Export set contains IoA and IoB.

Move Organization Usage Scenarios

Move organization is an administrative task and may only be performed by someone with appropriate administration rights. This means that permissions and ownership do not play a role when performing the move operation.

Move Organization with Association or Aggregation Relationship

When moving an asset from one organization to another, the Association and Aggregation Relationships do not change any related assets. This is because both of these relationships are considered to be loosely coupled.

Given the model:



OR



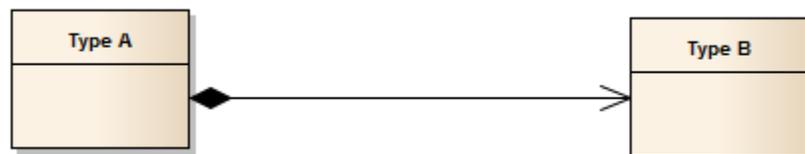
The result of the Move Organization operation is:

Operation	Expected result
Move IoB	Only IoB is moved to the new organization. It does not move IoA.
Move IoA	Only IoA is moved to the new organization. It does not move IoB.

Move Organization with Composition Relationship

When a Composition Relationship with the appropriate containing rule is found, then traverse the relationship and move the sub-components to the new organization.

Given the model:



The result of the Move Organization operation is:

Operation	Expected result
Move IoB	Only IoB is moved to the new organization. It does not move IoA.
Move IoA	IoA and IoB is moved to the new organization.

Change Ownership Usage Scenarios

Change Ownership is an administrative task and may only be performed by someone with appropriate administration rights. This means that permissions and ownership do not play a role when performing the change ownership operation.

Change Ownership with Association or Aggregation Relationship

When changing an asset's ownership from one user to another, the Association and Aggregation Relationships do not change any related assets. This is because both of these relationships are considered to be loosely coupled.

Given the model:



OR



The result of the Change Ownership operation is:

Operation	Expected result
Change Ownership of IoB	Change the ownership of IoB. It does not affect IoA.
Change Ownership of IoA	Change the ownership of IoA. It does not affect IoB.

Propagation of Profile Permissions

In addition to propagating permissions that control the access to an asset instance (as described above), it is also possible to propagate permissions that control the access to the asset instance's profiles.

Profile permissions of the root asset of a composite asset can be propagated to the other components if the components have the same type as the root asset. This restriction arises because different asset types can have different sets of profiles, whereas assets of the same type have the same set of profiles.

Propagation of profile permissions is activated when you select the check box **Propagate profile permissions** in the asset's **Permissions** tab. This checkbox can only be selected if you have also selected the check box **Propagate Permissions to dependent objects**.

Predefined Composite Asset Types

The following sections identify the nonshared components, shared components, and required objects that are associated with each of the predefined composite types installed with CentraSite.

Service

List all nonshared components, shared components, and required objects associated with the predefined composite type Service.

Nonshared Components	Description
Binding(s)	The objects that represent the specific ports that are defined in the WSDL. (A port defines a specific endpoint where the service is provided.)
Interface(s)	The objects that represent the portType that is a defined in the WSDL. (A portType defines a set of operations that the service provides.)
Operation(s)	The objects that represent the individual operations that the service provides.
Service WSDL	The ExternalLink to the WSDL file and the WSDL file itself.
Other WSDL (references to WSDLs that are imported or included in the main service WSDL)	The ExternalLinks to the referenced WSDL files and the referenced WSDL files themselves.
BPEL Partner Link Type	The object that represents the BPEL Partner Link Type to which the service is related (if such an association exists).
BPEL Role	The object that represents the BPEL Role to which the service is related (if such an association exists).
Shared Components	Description
XML Schema(s)	The entire graph of XML schemas that are related to the service. (The graph includes all of the XML schemas that the service references directly or indirectly). For each XML schema in the graph, the collector collects the ExternalLink to the XSD file and the actual XSD file itself.
Required Objects	Description
Service type	The Type object that defines the structure of a Service object in this registry (including all user-defined profiles that have been defined for the type).
WS-Policy	The WS-Policy objects that are associated with the service (if any).
Supporting Documents	ExternalLinks that point to files in the supporting document library plus the files themselves (if any).
	Note:

Required Objects**Description**

The list of supporting documents that the collector returns includes:

- documents that have been attached to the service using any of the predefined File attributes defined in the Service type
- documents that have been attached to the service using a custom File attribute
- any documents that have been attached to the service using an ad-hoc ExternalLink

Virtual Service

List all nonshared components, shared components, and required objects associated with the predefined composite type Virtual Service.

Nonshared Components**Description**

The set of nonshared components defined for the Service type. See nonshared components in **Service**.

WS-Policy

The WS-Policy object associated with the virtual service.

Processing Steps

The policy objects that represent the processing steps for the virtual service.

Extrinsic object

An internal copy of the WSDL that is maintained for the virtual service.

VSD

The virtual service's virtual service descriptor (VSD).

Shared Components**Description**

The set of shared components defined for the Service type. See shared components in **Service**.

Required Objects**Description**

Service type

The Type object that defines the structure of a Service object in this registry (including all user-defined profiles that have been defined for the type).

Native service

The Service object from which the virtual service was generated.

Supporting Documents

ExternalLinks that point to files in the supporting document library plus the files themselves.

Note:

Required Objects**Description**

The list of supporting documents that the collector returns includes:

- documents that have been attached to the service using any of the predefined File attributes defined in the Service type
- documents that have been attached to the virtual service using a custom File attribute
- any documents that have been attached to the virtual service using an ad-hoc ExternalLink

REST Service

List all nonshared components, shared components, and required objects associated with the predefined composite type REST Service.

Nonshared Components**Description**

ServiceBinding(s)	Each ServiceBinding object represents an <endpoint/> element of the REST Service WSDL20.
Binding Concept(s)	Each Binding Concept object represents a <binding/> element of the REST Service WSDL20.
Interface(s)	Each Interface object represents an <interface/> element of the REST Service WSDL20.
SpecificationLink(s)	SpecificationLinks are used to link the Interface and Binding Concept objects to a ServiceBinding object.
Operation(s)	Each Operation object represents a REST Resource and is represented as an element in the WSDL20.
Service WSDL20	The ExternalLink to the WSDL20 file (in the CentraSite repository) and the WSDL20 file itself.

Shared Components**Description**

XML Schema(s)	The entire graph of XML Schemas that are related to the REST Service. (The graph includes all of the XML Schemas that the REST Resource reference directly or indirectly). For each XML Schema in the graph, the collector collects the ExternalLink to the XSD file (in the CentraSite repository) and the actual XSD file itself.
---------------	---

Required Objects	Description
Service Type	The ObjectType Concept that defines the structure of a Service object in this registry (including all user defined profiles that have been defined for the type).
CentraSiteVirtualType	The CentraSiteVirtualType Concept that identifies the VirtualType of the Service object.
WS-Policy	The WS-Policy objects that are associated with the service (if any).
Supporting Documents	ExternalLinks that point to files in the supporting document library plus the files themselves (if any).

Note:

The list of supporting documents that the collector returns includes:

- documents that have been attached to the service using any of the predefined File attributes defined in the Service type
- documents that have been attached to the service using a custom File attribute
- any documents that have been attached to the service using an ad-hoc ExternalLink

OData Service

List all nonshared components, shared components, and required objects associated with the predefined composite type OData Service.

Nonshared Components	Description
OData Resources	Each OData Resource object represents an <EntitySet/>, <FunctionImport/>, <ActionImport/>, or <Singleton/> element of the OData Service EDMX.
Operation(s)	Each Operation object represents an HTTP method supported by an OData Resource.
Associated Documents	The ExternalLink to the EDMX file (Metadata document) and the EDMX file itself.

Shared Objects	Description
None	n/a

Required Objects	Description
Service type	The Type object that defines the structure of an OData Service object in this registry (including all user-defined profiles that have been defined for the type).

XML Service

List all nonshared components, shared components, and required objects associated with the predefined composite type XML Service.

Nonshared Components	Description
ServiceBinding(s)	Each ServiceBinding object represents an <endpoint/> element of the XML Service WSDL20.
Binding Concept(s)	Each Binding Concept object represents a <binding/> element of the XML Service WSDL20.
Interface(s)	Each Interface object represents an <interface/> element of the XML Service WSDL20.
SpecificationLink(s)	SpecificationLinks are used to link the Interface and Binding Concept objects to a ServiceBinding object.
Operation(s)	Each Operation object represents an XML Service and is represented as an element in the WSDL20.
Service WSDL20	The ExternalLink to the WSDL20 file (in the CentraSite repository) and the WSDL20 file itself.

Shared Components	Description
XML Schema(s)	The entire graph of XML Schemas that are related to the XML Service. (The graph includes all of the XML Schemas that the XML Service reference directly or indirectly). For each XML Schema in the graph, the collector collects the ExternalLink to the XSD file (in the CentraSite repository) and the actual XSD file itself.

Required Objects	Description
Service Type	The ObjectType Concept that defines the structure of a Service object in this registry (including all user defined profiles that have been defined for the type).
CentraSiteVirtualType	The CentraSiteVirtualType Concept that identifies the VirtualType of the Service object.
WS-Policy	The WS-Policy objects that are associated with the service (if any).

Required Objects	Description
Supporting Documents	ExternalLinks that point to files in the supporting document library plus the files themselves (if any).

Note:

The list of supporting documents that the collector returns includes:

- documents that have been attached to the service using any of the predefined File attributes defined in the Service type
- documents that have been attached to the service using a custom File attribute
- any documents that have been attached to the service using an ad-hoc ExternalLink

XML Schema

List all nonshared components, shared components, and required objects associated with the predefined composite type XML Schema.

Nonshared Components	Description
XSD File	The ExternalLink to the XSD file and the XSD file itself.
XML Schema(s) (referenced)	The entire graph of XML schemas that are related to this XML schema. (The graph includes all of the schemas that this XML schema references directly or indirectly). For each XML schema in the graph, the collector collects the ExternalLink to the XSD file and the actual XSD file itself.

Shared Objects	Description
None	n/a

Required Objects	Description
XML Schema type	The Type object that defines the structure of an XML Schema object in this registry (including all user-defined profiles that have been defined for the type).

BPEL Process

List all nonshared components, shared components, and required objects associated with the predefined composite type BPEL Process.

Nonshared Components	Description
BPEL File	The ExternalLink to the BPEL document and the BPEL document itself.
BPEL Partners	The objects that represent partners in the BPEL process.
PartnerLinks	The objects that represent partner links in the BPEL process.
Association Type (Service)	The association type that CentraSite uses to relate a BPEL process to a service.

Shared Objects	Description
-----------------------	--------------------

None	n/a
------	-----

Required Objects	Description
-------------------------	--------------------

BPEL type	The Type object that defines the structure of a BPEL object in this registry (including all user-defined profiles that have been defined for the type).
Services	The services that are referenced by the BPEL's PartnerLinks. (This includes all of the components and required objects associated with the referenced services. For a detailed list of these components and required objects, see Service .)
PartnerLinkTypes	The PartnerLinkTypes that are referenced by the BPEL's PartnerLinks.
Roles	The Roles that are referenced by the BPEL's PartnerLinks.

WS-Policy

List all nonshared components, shared components, and required objects associated with the predefined composite type WS-Policy.

Nonshared Components	Description
Policy	The object representing the policy itself.
Policy Parameter	The objects which form the parameters defined for the policy (if any).

Note:

There can be multiple levels of parameters which can be nested; parameters from all levels is included.

Nonshared Components	Description
Policy Condition (if any)	The object representing the conditions defined for the policy (for example, if the policy has conditions such as Name contains XML).

Shared Objects	Description
None	n/a

Required Objects	Description
Custom policy action	A custom action that is used by the policy and is not available as part of the predefined set.
Custom policy action parameters	All action parameters used by a custom action that the policy uses (if any).
Custom asset types	The custom defined asset types that the policy is defined for (if any).
Registry object	The registry object that the policy uses as a parameter (if any).

BPM Process Project

List all components associated with the predefined composite type BPM Process Project.

Components	Description
Process	Contains a set of Process Steps that invokes services and possibly other processes and has documents as inputs and outputs.

BPM Process Step

List all components associated with the predefined composite type BPM Process Step.

Components	Description
CAF Task Type	webMethods Task Engine human activity task.
CAF Security Role	A role which has the privilege to participate in CAF actions.
E-Form	The E-form object represents electronic forms that support forms-driven processes.
IS Service	Represents a webMethods IS Service Type.
Process Pool	Allows grouping of process steps into an internal or external process. More than one pool may exist within a process.

Components	Description
Process	Contains a set of Process Steps that invokes services and possibly other processes and has documents as inputs and outputs.
Service	A service is a software component that is described through a well defined interface and is capable of being accessed through standard network protocols such as, but not limited to, SOAP over HTTP. CentraSite is able to extract metadata of services based on a WSDL description.
User	The predefined JAXR-based type User.
XML Schema	A reference to an XML Schema file.

IS Package

List all components associated with the predefined composite type IS Package.

Components	Description
IS Specification	Represents a webMethods IS Specification Type.
IS Type Definition	Represents a webMethods IS Type Definition Type.
IS Routing Rule	Represents a webMethods IS Routing Rule Type.
IS Service Interface	Represents a webMethods IS Service Interface Type.
IS Connection	Represents a webMethods IS Connection Type.
IS Service	Represents a webMethods IS Service Type.

IS Service Interface

List all components associated with the predefined composite type IS Service Interface.

Components	Description
Binding(s)	The objects that represent the specific ports that are defined in the WSDL. (A port defines a specific endpoint where the service is provided.)
Interface(s)	The objects that represent the portType that is defined in the WSDL. (A portType defines a set of operations that the service provides.)
Operation(s)	The objects that represent the individual operations that the service provides.
IS Service	Represents an Integration Server (IS) Service operation.

Components	Description
REST Service	Represents a corresponding REST Service in the Integration Server (IS).

Process

List all components and required objects associated with the predefined composite type Process.

Components	Description
Process Step	Represents an activity in a Process.
Process Pool	Allows grouping of Process Steps into an internal or external Process. More than one Process Pool may exist within a Process.

Required Object	Description
BPM Process Project	A user defined project that allows users to group BPM assets.

Process Pool

List all components associated with the predefined composite type Process Pool.

Components	Description
Process Swimlane	Allows the grouping of Process Steps by actor.

Process Swimlane

List all components associated with the predefined composite type Process Swimlane.

Components	Description
Process Step	Represents an activity in a Process.

Web Application

List all components associated with the predefined composite type Web Application.

Components	Description
Web Application Page	Construct that is used to build User Interface pages for CAF web, portlet and task applications.

Components	Description
Portlet	Portlet built using webMethods CAF User Interface technology. Supports JSR 168.
CAF Task Type	webMethods Task Engine human activity task.
CAF Security Role	Java Web Application Security Role.
JDBC Datasource	Connection to a JDBC database.

Portlet

List all components associated with the predefined composite type Portlet.

Components	Description
Portlet Preference	Allows customized portlet behavior.
Web Application Page	Construct that is used to build User Interface pages for CAF web, portlet and task applications.

CAF Task Type

List all components associated with the predefined composite type CAF Task Type.

Components	Description
CAF Task Rule	webMethods Task Engine Task Assignment or Event execution mechanism.

TN Group

List all components associated with the predefined composite type TN Group.

Components	Description
TN Document Type	A description of a document type that is expected in a user's Trading Network.

Business Rules Project

List all components associated with the predefined composite type Business Rules Project.

Components	Description
Rule Set	A container for related rule metaphor assets.

Components	Description
Data Model	Defines a set of data elements available to a rule.
Rule Action	Represents some external behavior.

Data Model

List all components associated with the predefined composite type Data Model.

Components	Description
Rule Parameter	Represents the connection from a metaphor to a data model.

Virtual XML Service

List all nonshared components, shared components, and required objects associated with the predefined composite type Virtual XML Service.

Nonshared Components	Description
<i>The set of nonshared components defined for the XML Service type.</i>	See nonshared components in XML Service .
WS-Policy	The WS-Policy object associated with the XML service.
Processing Steps	The policy objects that represent the processing steps for the Virtual XML Service.
Extrinsic object	An internal copy of the WSDL20 that is maintained for the Virtual XML Service.
VSD	The Virtual XML Service's Virtual Service Descriptor (VSD).

Shared Components	Description
<i>The set of shared components defined for the XML Service type.</i>	See shared components in XML Service .

Required Objects	Description
Service type	The ObjectType Concept that defines the structure of a Service object in this registry (including all user-defined profiles that have been defined for the type).

Required Objects	Description
CentraSiteVirtualType	The CentraSiteVirtualType Concept that identifies the VirtualType of the Service object.
Native service	The XML Service object from which the Virtual XML Service was generated.
Supporting Documents	ExternalLinks that point to files in the supporting document library plus the files themselves.

Note:

The list of supporting documents that the collector returns includes:

- documents that have been attached to the service using any of the predefined File attributes defined in the Virtual XML Service type
- documents that have been attached to the virtual service using a custom File attribute
- any documents that have been attached to the virtual service using an ad-hoc ExternalLink

Virtual REST Service

List all nonshared components, shared components, and required objects associated with the predefined composite type Virtual REST Service.

Nonshared Components	Description
<i>The set of nonshared components defined for the REST Service type.</i>	See nonshared components in REST Service .
WS-Policy	The WS-Policy object associated with the REST service.
Processing Steps	The policy objects that represent the processing steps for the Virtual REST Service.
Extrinsic object	An internal copy of the WSDL20 that is maintained for the Virtual REST Service.
VSD	The Virtual REST Service's Virtual Service Descriptor (VSD).

Shared Components	Description
<i>The set of shared components defined for the REST Service type.</i>	See shared components in REST Service .

Required Objects	Description
Service type	The ObjectType Concept that defines the structure of a Service object in this registry (including all user-defined profiles that have been defined for the type).
CentraSiteVirtualType	The CentraSiteVirtualType Concept that identifies the VirtualType of the Service object.
Native service	The XML Service object from which the Virtual REST Service was generated.
Supporting Documents	ExternalLinks that point to files in the supporting document library plus the files themselves.

Note:

The list of supporting documents that the collector returns includes:

- documents that have been attached to the service using any of the predefined File attributes defined in the Virtual XML Service type
- documents that have been attached to the virtual service using a custom File attribute
- any documents that have been attached to the virtual service using an ad-hoc ExternalLink

Virtual OData Service

List all nonshared components, shared components, and required objects associated with the predefined composite type Virtual OData Service.

Nonshared Components	Description
<i>The set of nonshared components defined for the OData Service type.</i>	See nonshared components in OData Service .
Processing Steps	The policy objects that represent the processing steps for the Virtual OData Service.

Shared Objects	Description
None	n/a

Required Objects	Description
Service type	The Type object that defines the structure of an OData Service object in this registry (including all user-defined profiles that have been defined for the type).
CentraSiteVirtualType	The CentraSiteVirtualType Concept that identifies the VirtualType of the OData Service object.
Native service	The OData Service object from which the Virtual OData Service was generated.

Managing Types through CentraSite Business UI

This section describes operations you can perform to manage types through CentraSite Business UI.

Adding an Asset Type

Pre-requisites:

To add a user-defined custom asset type definition, you must have the Manage Asset Types permission.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

You might want to create a custom asset type, if you have an asset that is not represented by any of the predefined types shipped with CentraSite.

When you create a custom asset type, keep the following points in mind:

- To include a classification attribute or relationship attribute in your custom asset type, make sure that the corresponding taxonomy exists before you begin creating the custom asset type. To add a classification attribute or relationship attribute to the custom asset type, you must specify the taxonomy with which the attribute is associated. You cannot do this unless the taxonomy already exists in CentraSite.
- Consider using classification attributes and relationship attributes instead of ordinary String attributes whenever possible. Among other benefits, these attribute types enable users to more easily discover assets and understand the relationships that an asset has with other objects in the registry.
- In general, use a classification attribute or an enumerated String attribute instead of an ordinary String attribute when you want the attribute to be more functional.
- Instead of defining multiple asset types to represent variants of the same basic type, consider creating one basic type and using a classification attribute to differentiate them. For example, instead of creating separate asset types for different kinds of Web services (for example, business

services, technical services, security services), use the one basic Web service asset type and use a classification attribute to classify its variations.

- When you are creating a custom asset type, think about the design/change-time policies that you may want to apply to assets of that type. If you need to apply different policies to different sub-sets of the asset type, use a classification attribute to differentiate the sub-sets.
- If you do not want users to be able to assign ad hoc classifiers and associations to instances of a particular type of asset, omit the Classifications and Associations profiles from that asset's type.
- To provide different views of an asset to different users or groups, divide the attributes among profiles in a way that enables you to use profile permissions to selectively show or hide the appropriate set of attributes to different users or groups.

➤ **To add a custom asset type**

1. In the CentraSite Business UI activity bar, click **Asset Types**.
2. Click **Add Asset Type**.
3. In the Create New Asset Type page, provide the required information for each of the displayed data fields.

Field	Description
Name	Type a name for the new asset type. Specify a name that your users will recognize and understand. For example, use <code>BPEL Process Document</code> , NOT <code>bpdoc</code> . The name you assign to the asset type can contain any character, including spaces.
Description	<i>Optional.</i> Type a short description for the new asset type. This description appears when a user displays the list of types in the Asset Types page.
Options	Specify the options that are required to be set specific to the type. The available options are: Enable lifecycle management Allow a lifecycle model to be applied to instances of this type. Enable reports Allow reports to be generated against instances of this type. Enable versioning Allow users to generate versions of instances of this type.

Note:

When this option is not selected, CentraSite disables the **New Version** action from instances of this type.

Field	Description
Policies can be applied	<p>Allow user-defined design/change-time policies to be created and enforced for instances of this type.</p> <p>Note: When this option is not selected, CentraSite does not apply any user-defined design/change-time policies to instances of this type, even in cases where the policy is designed to execute against <i>all</i> asset types.</p> <p>Note: This option does not apply to system policies. CentraSite will apply system policies to instances of this type, irrespective of the option.</p>
Require consumer registration	Require users to register an application when they submit consumer registration requests for instances of this type.
Top level type	<p>Allow users to create instances of this type from scratch. When this option is selected, users are allowed to create instances of this type using the Create New Asset page in CentraSite Business UI.</p> <p>Generally, you will not select this option for types that are constituents of other assets. For example, the Operation type is used to represent an operation that belongs to a Web service. Operations are derived automatically from the service WSDL. They are not intended to be manually defined by users. Therefore, the Operation type is not designated as a top level type.</p>
Visible in asset browse	Allow instances of this type to be displayed in the Browse list. When this option is selected, CentraSite includes the type in the Choose Asset Types box (go to Browse > Additional Search Criteria > Asset Types > Choose). When this option is not selected, CentraSite omits the type from the Choose Asset Types box so users cannot browse for instances of this type. Including a type in the Choose Asset Types box enables users to define queries that select on that specific type.
Visible in search	Allow users to define a type-ahead Search for instances of this particular type. When this option is selected, CentraSite includes the type in the Scopes box next to the Search text box). When this option is not selected, CentraSite omits the type from the Scopes box so users cannot search for instances of this type.

4. Click **Save**.

The user-defined asset type is added to the CentraSite registry repository.

Adding Attribute to an Asset Type

Pre-requisites:

To add a user-defined attribute to the asset type definition, you must have the Manage Asset Types permission.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

> To add an attribute to a new asset type

1. In the CentraSite Business UI activity bar, click **Asset Types**.

A list of defined asset types is displayed in the Asset Types page.

2. Click the asset type to that you want to add an attribute.

The Asset Type details page is displayed. Also, the Actions bar displays a set of actions that are available for working with the displayed type.

3. On the actions bar of the Asset Type details page, click **Add Attribute**.
4. In the **Add Attribute** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Name	Type a display name for the new attribute. This is the attribute name that CentraSite displays in instances of this type in CentraSite Business UI. Ensure the display name is meaningful. The display name can contain any combination of characters, including spaces. Note: If you are defining a relationship attribute, by default the attribute's name is derived from the name of the association type that you assign to the attribute. You can, however, assign a custom name to the relationship attribute by specifying the Name attribute.
Description	<i>Optional.</i> Type a short description for the new attribute.
Data Type	Select the data type for this attribute.

Field	Description
	<p>Select the data type carefully. You cannot change the data type after a type is added to the CentraSite registry.</p> <p>Additional type-specific fields or check boxes are displayed based on the selected data type.</p>
Required	<p>Select the Required check box to restrict users from saving an instance of this type without first assigning a value to this attribute.</p> <p>In CentraSite Business UI, a required attribute field with an empty value is indicated with a warning icon.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: An attribute can be a required attribute and have a default value. If you do not supply a value for an attribute that is required and has a default value, the default value is automatically assigned to this attribute.</p> </div> <p>Consider the following guidelines, when modifying an existing type:</p> <ul style="list-style-type: none"> ■ When there are no instances of the type in the registry, you can add a required attribute to an existing type. ■ When there are instances of the type in the registry, you can add a required attribute of type <code>slot</code> or <code>classification</code>. However, you cannot add a required attribute of type <code>relationship</code> or <code>file</code>. ■ When you add a new required attribute, no automatic update of existing instances takes place. This prevents the potential degradation of performance that could arise from the automatic update of a large number of instances. ■ You can select the Required check box for an existing attribute, even if there are empty instances of that attribute. However, in this case a default value must be provided for the attribute. ■ You can clear the Required check box for an attribute at any time (even if there are instances of the attribute existing in the registry).
Multi Value	<p>Select the Multi Value check box to allow an attribute to hold multiple values (that is, an array of values).</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: The Multi Value check box is not available for Boolean data type.</p> </div> <p>Consider the following guidelines, when modifying an existing type:</p> <ul style="list-style-type: none"> ■ You can switch on an attribute to Multi Value at any time (even if there are instances of the attribute existing in the registry).

Field	Description
	<ul style="list-style-type: none">■ You can switch off an attribute from Multi Value only if instances of the type exist in the registry, and each instance has at most one value assigned to this attribute (that is, no instances exist wherein this attribute has multiple values).

Default Value *Optional.* Type a value you want to assign to the attribute by default.

Note:

The **Default Value** option is not available for all attribute types.

Consider the following guidelines, when modifying an existing type:

- You can change a default value, assign a default value, or remove a default value from an attribute at any time. Changing the attribute's default value *does not* immediately affect any existing instances of the attribute.
- If you add a default value to an attribute that did not previously have one, and the registry contains empty instances of that attribute, the default value is assigned to those assets the next time that they are saved to the registry.
- If you add a new attribute to an existing type and you assign a default value to that attribute, the default value is assigned to the existing instances of that type the next time those instances are saved to the registry.

If an attribute has the **Required** check box, then the following conditions apply:

- When creating a new type definition with one or more required attributes, you do not need to provide a default value for the required attributes.
- If instances of a type exist, and you update the type definition in any manner, regardless of whether or not you modify the required attributes, you must provide a default value for each of the required attribute in the type definition. Required attributes that have no value is set to the default value the next time an instance of the particular type is updated in the registry.
- If instances with missing required attributes are viewed in CentraSite Business UI, these attributes are simulated and displayed with the default value. But the default value does not be added to the instance until the next update in the registry.

5. Click **OK**.

The user-defined attribute is added to the type.

Adding Profile to an Asset Type

Pre-requisites:

To add a user-defined profile to the asset type definition, you must have the Manage Asset Types permission.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

You define custom profiles for logical grouping of attributes in a type definition.

You add a custom profile to the type definition in two ways:

- **Using the available attributes in an asset type.** To define a profile manually, you select attributes from the list of available attributes and assign them to the profile.
- **Using a Java plug-in that contains a computed profile.** A computed profile is a user-defined profile that is implemented as a Java plug-in. You add the computed profile to the asset type by importing the computed profile's definition from an archive file. The plug-in specifies the attributes that are contained in the profile. The plug-in also has the sole responsibility for rendering the layout representation within the profile. After a computed profile has been defined for an asset type, the computed profile is treated in the same way as any other profile; for example, permissions for computed profiles can be granted in the same way as for standard profiles, and the ordering of profiles within a type definition is the same for computed profiles as for standard profiles.

> To add a profile to an asset type

1. In the CentraSite Business UI activity bar, click **Asset Types**.

A list of defined asset types is displayed in the Asset Types page.

2. Click the asset type to that you want to add a profile.

The Asset Type details page is displayed. Also, the Actions bar displays a set of actions that are available for working with the displayed type.

3. On the actions bar of the Asset Type details page, click **Add Profile**.

4. In the **Add Profile** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Profile Name	Type a display name for the new profile.

Field	Description
	<p>This is the profile name that CentraSite displays in instances of this type in CentraSite Business UI. Ensure the display name is meaningful (for example, Technical Notes, not tn).</p> <p>The display name can contain any combination of characters, including spaces.</p>
Computed Profile	<p>Select the Computed Profile check box to add a computed profile, and then select the implementation language from the drop-down list box.</p> <p>When this check box is selected, CentraSite renders an additional input field Profile Implementation Archive.</p>
Profile Implementation Archive	<p>Click Choose to browse and select an archive file that contains the computed profile's definition, and upload the computed profile definition to CentraSite.</p> <p>For information on how to implement a computed profile for CentraSite Business UI, see <i>CentraSite Developer's Guide</i>.</p>

5. Click **OK**.

The user-defined profile is added to the type.

Viewing the Asset Type List

You can view the list of available asset types, add a custom asset type, and delete an existing asset type in the **Asset Types** page.

> To view the list of asset types

1. In the CentraSite Business UI activity bar, click **Asset Types**.

A list of defined asset types is displayed in the Asset Types page.

The Asset Types page provides the following information about each asset type:

Column	Description
Name	The name of the asset type.
Description	A short description about the asset type.
Last Updated	The date on which the asset type was last modified.

You can adjust the view to show or hide any of the available attributes by opening the drop-down list labeled **View** and selecting the attributes that you want to include in the view, and clearing the attributes that you do not want to view.

You can change the order in which the attributes are displayed by opening the drop-down list labeled **Sort by**. The list displays all the attributes that are selected in the **View** drop-down list. The order in which the attributes appear in the **Sort by** drop-down list is the order in which the attributes appear for each displayed user. To change the order in which any given attribute is displayed, select the attribute in the drop-down list **Sort by** and use the arrows to move the attribute to the required position.

In the Asset Types page, an action bar displays one or more icons that you can use to perform various tasks on the selected types.

Action	Description
Add Asset Type	Create a new custom asset type in CentraSite.
Delete	Remove an existing asset type from CentraSite.

Modifying Asset Type Details

Pre-requisites:

To modify the details of a user-defined asset type, you must have the Manage Asset Types permission. Besides allowing you to modify user-defined asset types, this permission allows you to modify certain predefined types installed by CentraSite.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

When you modify the details of a custom asset type, keep the following points in mind:

- With respect to attributes:
 - You can add new optional attributes to the asset type at any time.
 - You can add new required attributes if there are no existing instances of the type in the registry. If there are existing instances of the type, you can add a new required attribute of type `slot` or `classification`, but not of type `relationship` or `file`.
 - If you add a new attribute, regardless of whether it is optional or required, no automatic update of existing instances takes place. This prevents the potential degradation of performance that could arise from the automatic update of a large number of instances.
 - You cannot modify the data type for an attribute, but you can modify many of an attribute's other options. Be aware that certain attributes are not permitted if assigned (that is, non-empty) instances of the attribute are present in the registry.
 - You cannot delete an attribute from a type if an instance of the type exists with a value assigned to the attribute. In such a situation, you must first remove the attribute's value from each such instance before you can delete the attribute.

Note that the command line tool `CentraSite command remove Attribute` provides support for removing an attribute, in cases where existing instances contain a value for the attribute. For information on the usage of `CentraSite command`, see [“Removing Attribute from Asset Type Definition” on page 274](#).

- You can add, modify, delete, rename, and reorganize the profiles associated with a type at any time.
- Beginning with version 9.9, `CentraSite` provides the ability to define consumable asset types for a type. If you are upgrading to `CentraSite 9.9` from a version of `CentraSite` that did not support the definition of consumable types, `CentraSite` assigns each Virtual type (Virtual Service, Virtual REST Service, Virtual OData Service, and Virtual XML Service) a default consumable type for the User, Group, and Application types.
- You cannot remove a consumable type from a type definition if an instance of the consumable type is already registered as a consumer of an instance of the type. In such a situation, you must first unregister each such consuming instance before you can remove the consumable type.

Important:

If you are modifying one of the predefined asset types installed with `CentraSite`, review the information in [“Introduction to Types” on page 182](#) before you begin. It explains the kinds of modifications that you can make to the predefined types.

Important:

If you are using `CentraSite` in conjunction with other software products, for example, the products of the `webMethods Product Suite` or a third-party product, those products can add their own asset types to `CentraSite`. Be aware that `CentraSite` treats these types as user-defined custom types, which can be modified by an administrator with the appropriate permissions (just like any other custom type). Modifying or deleting these types in `CentraSite` can lead to inconsistencies or errors in the product that uses the type. For example, if you modify or delete a type that is used by the `webMethods Product Suite`, components such as the `webMethods Integration Server` may no longer be able to publish assets to `CentraSite`. To prevent these types of errors, *do not modify or delete any asset type on which other Software AG components or third-party products depend.*

➤ To modify the details of an asset type

1. In the `CentraSite Business UI` activity bar, click **Asset Types**.

A list of defined asset types is displayed in the `Asset Types` page.

2. Click the asset type whose details you want to modify.

The `Asset Type` details page is displayed. Also, the `Actions` bar displays a set of actions that are available for working with the displayed type.

3. In the `Basic Information` section, examine or modify the type's name, description, and its specific options as required.

4. In the Attributes section, examine or modify the type's attribute information as required.
5. In the Profiles section, examine or modify the type's profile information as required.
6. After you have made the required changes, click **Save** to update the asset type definition.

Modifying Profile Attribute List of an Asset Type

Pre-requisites:

To modify the attribute list of a profile in the asset type definition, you must have the Manage Asset Types permission.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

You might want to modify the attribute list of a profile when you want to:

- Add one or more attributes to the profile.
- Remove one or more attributes from the profile.
- Rearrange a set of attributes within the profile.

➤ To modify the attribute list of a profile

1. In the CentraSite Business UI activity bar, click **Asset Types**.

A list of defined asset types is displayed in the Asset Types page.

2. Click the asset type that contains the required profile.

The Asset Type details page is displayed. The Profiles section contains the list of available profiles for the type.

3. Click the chevron next to the profile whose attribute list you want to modify.
4. To add an attribute to the profile, use the **Drag and Drop** icon to drag an attribute from the attribute list in the Attributes section and drop it over to the required profile in the Profiles section.
5. To remove an attribute from the profile, follow these steps:
 - a. Hover over the attribute you want to remove.

This displays icons for one or more actions that you can perform on the attribute.

- b. Click **Delete**.

6. To change the display order of the attributes that are displayed in the profile, use the **Move up** and **Move down** arrow icons.
7. After you have made the required changes, click **Save** to update the profile information.

Modifying Consuming Types of an Asset Type

Pre-requisites:

To modify the consuming types of a user-defined asset type, you must have the Manage Asset Types permission.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

You might want to modify the consuming types of an asset type to change the consumer registration list of the type.

➤ To modify the consuming types of an asset type

1. In the CentraSite Business UI activity bar, click **Asset Types**.

A list of defined asset types is displayed in the Asset Types page.
2. Click the asset type whose consuming types you want to modify.

The Asset Type details page is displayed. Also, the Actions bar displays a set of actions that are available for working with the displayed type.
3. On the actions bar of the Asset Type details page, click **Edit Consuming Types**.
4. In the **Consuming Type** box, select the types whose instances you want to define as consumers to instances of this type. In the instance details of this type, only the instances of the defined consuming types will be listed for consumer registration.
5. Click **OK**.

The asset type definition is updated.

Deleting Asset Types

Pre-requisites:

To delete a user-defined custom asset type definition, you must have the Manage Asset Types permission.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

When you delete an asset type, keep the following points in mind: :

- You can delete a type *only if there are no instances of that type in the registry*.
- The core asset types that belong to CentraSite are non-deletable. CentraSite does not allow you to delete these types, even if there are no instances of the selected type in the registry.

Important:

If you have selected several asset types where one or more of them are predefined types, you can use the **Delete** button to delete the types. However, as you are not allowed to delete predefined asset types, only types you have permission for is deleted. The same applies to any other types for which you do not have the required permission.

Important:

If you are using CentraSite in conjunction with other software products, for example, the products of the webMethods Product Suite or a third-party product, those products can add their own asset types to CentraSite. Be aware that CentraSite treats these types as user-defined custom types, which can be deleted by an administrator with the appropriate permissions (like any custom type). Deleting these types in CentraSite can lead to inconsistencies or errors in the product that uses the type. For example, if you delete a type that is used by the webMethods Product Suite, components such as the webMethods Integration Server may no longer be able to publish assets to CentraSite. To prevent these types of errors, *do not delete any asset type on which other Software AG components or third-party products depend*.

➤ **To delete types from the asset types page**

1. In the CentraSite Business UI activity bar, click **Asset Types**.

A list of defined asset types is displayed in the Asset Types page.

2. In the Asset Types page, select one or multiple types.
3. On the Actions bar of the Asset Types page, click **Delete**.
4. Click **OK** in the confirmation dialog box.

Each selected type is permanently removed from the CentraSite registry repository.

Managing Types through CentraSite Control

This section describes operations you can perform to manage types through CentraSite Control.

Adding an Asset Type

To create a new type definition, you must have the Manage Asset Types permission.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

If you have an asset that is not represented by one of the predefined types provided by CentraSite, you must create a custom asset type for it. CentraSite provides a wizard for creating custom asset types. If you want users to be able to generate the asset type from an input file, you must also create a custom importer for that type and register the importer in CentraSite.

Follow these general guidelines when creating a custom asset type in CentraSite:

- An asset type has two names: a *display name* and a *schema name*.
 - The display name is the name that CentraSite uses when it refers to the type in the user interface. (This is the name that appears in the catalog browser, for example.) The display name can contain any character, including spaces.
 - The schema name is the name that is given to the underlying schema that contains the type definition. The schema name must conform to the naming requirements for an NCName data type, which does not permit names with spaces or most special characters.

By default, CentraSite derives the schema name from the display name that you specify for the type. If your display name includes special characters, then you must explicitly specify a schema name that is NCName conformant in the type's **Advanced Settings** dialog box.

- To include a Classification attribute or Relationship attribute in your custom asset type, make sure that the corresponding taxonomy exists before you begin creating the custom asset type. To add a Classification attribute or Relationship attribute to the asset type, you must specify the taxonomy with which the attribute is associated. You cannot do this unless the taxonomy already exists in CentraSite.
- Consider using Classification attributes and Relationship attributes instead of ordinary String attributes whenever possible. Among other benefits, these attribute types enable users to more easily discover assets and understand the relationships that an asset has with other objects in the registry.
- In general, use a Classification attribute or an enumerated String instead of an ordinary String attribute when you want the attribute to be more strongly typed.
- Instead of defining multiple asset types to represent variants of the same basic type, consider creating one basic type and using a classification attribute to differentiate them. For example, instead of creating separate asset types for different kinds of Web services (for example, business services, technical services, security services), use the one basic Web service asset type and use a Classification attribute to classify its variations.
- When you are creating a custom asset type, think about the design/change-time policies that you may want to apply to assets of that type. If you need to apply different policies to different sub-sets of the asset type, use a Classification attribute to differentiate the sub-sets.
- If you do not want users to be able to assign ad hoc classifiers and associations to instances of a particular type of asset, omit the Classifications and Associations profiles from that asset's type.

- To provide different views of an asset to different users or groups, divide the attributes among profiles in a way that enables you to use profile permissions to selectively show or hide the appropriate set of attributes to different users or groups.

1. In CentraSite Control, go to **Administration > Types**.
2. In the **Types** tab, click **Add Asset Type**.

This opens the **Add Asset Type** wizard.

3. In panel 1 of the **Add Asset Type** wizard, provide the required information for each of the displayed data fields.

Field	Description
Name	<p>Type a name for the new asset type.</p> <p>Specify a name that your users will recognize and understand. For example, use BPEL Process Document, not bpdoc.</p> <p>The name you assign to the asset type can contain any character, including spaces. However, if you specify a name that does not conform to the NCName type, you <i>must</i> click Advanced Settings and specify a name that is NCName conformant in the Schema Name field.</p> <div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p>Note: If the name that you assign to the asset type is NCName conformant, except that it includes spaces, it is not necessary to explicitly specify the type's schema name. CentraSite automatically replaces space characters with the _ character when it generates the schema name for an asset type.</p> </div>
Description	<p><i>Optional.</i> Type a short description for the new asset type. This description appears when a user displays the list of types in the Type Management page.</p>

4. Click **Advanced Settings**.
5. In the **Add Asset Type - Advanced Settings** dialog box, provide the required information for each of the displayed data fields.
 - a. Verify that the schema name and the namespace name that were generated by CentraSite are valid.

Field	Description
Schema Name	<p>Verify that the schema name that CentraSite generated for this asset type is NCName conformant.</p>

Field**Description**

CentraSite automatically populates this field with a name that is derived from the asset type name that you specified in the previous step. For example, if your asset type name is *My Asset Name*, CentraSite automatically populates this field with *My_Asset_Name*.

If the schema name generated by CentraSite does not meet the following criteria, you must specify a name that does.

- The name must begin with a letter or the underscore character (`_`).
- The remainder of the name can contain any combination of letters, digits, or the following characters: `.` `-` `_` (that is, period, dash, or underscore). It can also contain combining characters and extender characters (for example, diacriticals).
- The name cannot contain any spaces.

Additionally, the type's fully qualified schema name must be unique among all types in the registry.

Note:

Modify the schema name carefully. You cannot change the schema name later.

Namespace

Modify the namespace that CentraSite has proposed for the type if necessary. By default, CentraSite generates the namespace in the following form:

```
http://namespaces.OrganizationName.com/Schema
```

Where, *OrganizationName* is the name of your organization.

The **Namespace** value is used to qualify the name specified in **Schema Name**. Together, the **Schema Name** and **Namespace** values produce the type's fully qualified name. This name must be unique within the registry.

You generally do not need to modify the namespace that CentraSite proposes for a type. In most cases, the proposed namespace is adequate. However, you might modify the namespace if you want it to include the name of a different organization or if you need to resolve a naming conflict between this type and an existing type.

Note:

Modify the namespace value carefully. You cannot change the namespace value later.

- b. To use a custom icon to represent this type in the user interface, upload large and small versions of the icon.

Note:

If you do not specify a custom icon, CentraSite assigns the default icon to the type.

Field	Description
Large Icon	<p><i>Optional.</i> Specify the large icon that is to be used to represent this type. CentraSite Control and CentraSite Business UI use this icon when it displays the details for an instance of the type.</p> <p>To use a custom icon, click Browse to search for a particular image file, and upload the file containing the large version of the icon to CentraSite. This icon must be in GIF format. To ensure proper alignment when it is displayed in the user interface, the icon must be 64 x 64 pixels in size.</p>
Small Icon	<p><i>Optional.</i> Specify the small icon that is to be used to represent this type. CentraSite Control displays this icon when instances of the type appear in lists or summary tables.</p> <p>To use a custom icon, click Browse to search for a particular image file, and upload the file containing the small version of the icon to CentraSite. This icon must be in GIF format. To ensure proper alignment when it is displayed in the user interface, the icon must be 16 x 16 pixels in size.</p>

c. Select one or more type-specific options:

Enable this option...	To...
Visible in Asset Browse	<p>Allow instances of this type to be displayed in the catalog browser. When you enable this option, CentraSite Control includes the type in the Asset Types pane on the Asset Catalog > Browse page. When you disable this option, CentraSite Control omits the type from the Asset Types pane so users cannot browse for instances of the type.</p>
Enable reports	<p>Allow reports to be generated against instances of this type.</p>
Policies can be applied	<p>Allow user-defined design/change-time policies to be created and enforced for instances of this type.</p>

Note:

If you disable this option, CentraSite does not apply any user-defined design/change-time policies to instances of the type, even in cases where the policy is designed to execute against *all* asset types.

Note:

This option does not apply to system policies. CentraSite will apply system policies to instances of the type whether this option is enabled or not.

Enable this option...	To...
Require Consumer Registration	Require users to register an application when they submit consumer registration requests for assets of this type.
Enable versioning	Allow users to generate versions of instances of this type. When you disable this option, CentraSite disables the Add New Versions command and omits the Versions profile from instances of this type.
Top Level type	<p>Allow users to create instances of this asset type from scratch. When you enable this option, users are allowed to create instances of the type using the Add Asset button in CentraSite Control.</p> <p>Generally you disable this option for types that are constituents of other assets, or for types that are only meant to be added to the registry by an importer. For example, the Operation type is used to represent an operation that belongs to a Web service. Operations are derived automatically from the service WSDL. They are not intended to be manually defined by users. Therefore, the Operation type is not designated as a "top level type".</p>
Enable Lifecycle Management	Allow a lifecycle model to be applied to assets of this type.
Visible in Search	<p>Allow users to define a search for this particular type. When you enable this option, CentraSite Control includes the type in the Types list on the Advanced Search page. Including a type in this list enables users to define queries that select on that specific type.</p> <p>Note: If you change the state of this option and then do an advanced search, you may need to refresh the Advanced Search page to see the change reflected in the Types list.</p>

- d. To specify asset types that can consume instances of this type, specify the consumable asset types:
 - a. Click **Consumer Registration Settings**.

This opens the **Consumer Registration Settings** dialog box.
 - b. In the field labeled **Consumable Type**, select an asset type to allow instances of the particular asset type to consume instances of this type.

Note:
If you do not specify a consumable asset type in the type definition (with the exception of Virtual Services, Virtual XML Services, and Virtual REST Services), the asset types Users, Groups, and Applications are available for registering as consumers of asset instances of this type.

- c. Click **OK**.
 - e. Click **OK** to complete the advanced settings of type definition.
6. Click **Finish**.

Adding an Attribute to Asset Type

Pre-requisites:

To add a new user-defined attribute to the asset type definition, you must have the Manage Asset Types permission.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

> To add an attribute to an asset type

1. In CentraSite Control, go to **Administration > Types**.
2. In the **Types** tab, click **Add Asset Type**.
This opens the **Add Asset Type** wizard.
3. In panel 2 of the **Add Asset Type** wizard, click **Add Attribute**.
This opens the **Add Attribute** dialog box.
4. In the **Add Attribute** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Name	Type a display name for the new attribute. This is the attribute name that CentraSite displays in instances of this type in CentraSite Control and CentraSite Business UI. Ensure the display name is meaningful. The display name can contain any combination of characters, including spaces. You can change an attribute's display name at any time.

Note:

If you are defining a Relationship attribute, by default the attribute's name is derived from the name of the association type that you assign to the attribute. You can, however, assign a custom name to the Relationship attribute by specifying the **Name** attribute.

Field	Description
Schema Name	<p>Type a schema name for the new attribute.</p> <p>This is the internal name that CentraSite assigns to the attribute. This name can contain only letters, numbers and the underscore character (_).</p> <p>If you do not enter a schema name, CentraSite automatically generates a schema name for the attribute based on the name you type in the Name field. However, if the name that CentraSite generates includes invalid characters, you is prompted to provide a valid schema name when you save the attribute.</p> <p>Note: select the schema name carefully. You cannot change the schema name after a type is added to CentraSite registry.</p> <p>Note: Attributes of the data type, Relationship, File, and Classification do not require a schema name.</p>
Description	<p><i>Optional.</i> Type a short description for the new attribute.</p>
Required	<p>Select the Required check box to restrict users from saving an asset instance of this type without first assigning a value to this attribute.</p> <p>In CentraSite Control and CentraSite Business UI, a required attribute is displayed with an asterisk (*).</p> <p>Note: An attribute can be a required attribute and have a default value. If you do not supply a value for an attribute that is required and has a default value, the default value is automatically assigned to this attribute.</p> <p>Consider the following guidelines, when modifying an existing type:</p> <ul style="list-style-type: none">■ When there are no instances of the type in the registry, you can add a required attribute to an existing type.■ When there are instances of the type in the registry, you can add a required attribute of type <code>slot</code> or <code>classification</code>. However, you cannot add a required attribute of type <code>relationship</code> or <code>file</code>.■ When you add a new required attribute, no automatic update of existing instances takes place. This prevents the potential degradation of performance that could arise from the automatic update of a large number of instances.■ You can select the Required check box for an existing attribute, even if there are empty instances of that attribute. However, in this case a default value must be provided for the attribute.

Field	Description
	<ul style="list-style-type: none"> ■ You can clear the Required check box for an attribute at any time (even if there are instances of the attribute existing in the registry).
Read-only	<p>Select the Read-only check box to restrict users from modifying the value of an attribute after an asset is added to the registry.</p> <p>You can clear the Read-only check box for an attribute at any time (even if there are instances of the attribute existing in the registry).</p>
Multiplicity	<p>Click the Single Value or Multi Value option to allow an attribute to hold just a single value or multiple values (that is, an array of values).</p> <p>Note: The Multiplicity option is non-editable and grayed out for Boolean type.</p> <p>Consider the following guidelines, when modifying an existing type:</p> <ul style="list-style-type: none"> ■ You can switch an attribute's Multiplicity to Multi Value at any time (even if there are instances of the attribute existing in the registry). ■ You can switch an attribute's Multiplicity to Single Value only if instances of the type exist in the registry, and each instance has at most one value assigned to this attribute. (that is, no instances exist wherein this attribute has multiple values).
Data Type	<p>Select the attribute's data type.</p> <p>Select the data type carefully. You cannot change the data type after a type is added to the CentraSite registry.</p> <p>Additional type-specific fields or check boxes are displayed based on the selected data type.</p>
Default Value	<p><i>Optional.</i> Type a value you want to assign to the attribute by default.</p> <p>Note: The Default Value option is not available for all attribute types.</p> <p>Consider the following guidelines, when modifying an existing type:</p> <ul style="list-style-type: none"> ■ You can change a default value, assign a default value, or remove a default value from an attribute at any time. Changing the attribute's default value <i>does not</i> immediately affect any existing instances of the attribute. ■ If you add a default value to an attribute that did not previously have one, and the registry contains empty instances of that attribute, the default value is assigned to those assets the next time that they are saved to the registry. ■ If you add a new attribute to an existing type and you assign a default value to that attribute, the default value is assigned to the existing

Field	Description
	instances of that type the next time those instances are saved to the registry.
	If an attribute has the Required check box, then the following conditions apply:
	<ul style="list-style-type: none">■ When creating a new type definition with one or more required attributes, you do not need to provide a default value for the required attributes.■ If instances of a type exist, and you update the type definition in any manner, regardless of whether or not you modify the required attributes, you must provide a default value for each of the required attribute in the type definition. Required attributes that have no value is set to the default value the next time an instance of the particular type is updated in the registry.■ If instances with missing required attributes are viewed in CentraSite Control, these attributes are simulated and displayed with the default value. But the default value does not be added to the instance until the next update in the registry.

Adding Profile to an Asset Type

You can define the custom profiles for logical grouping of attributes in the type definition by using the **Add Profile** dialog box in CentraSite Control.

To add a new profile to the type definition, you must have the Manage Asset Types permission.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

You can add a profile in two ways:

- **Manually, using the asset type's available attributes.** To define a profile manually, you select attributes from the list of available attributes and assign them to the profile.
- **Using a Java plug-in that contains a computed profile.** A computed profile is a user-defined profile that is implemented as a Java plug-in. You add the computed profile to the asset type by importing the computed profile's definition from an archive file. The plug-in specifies the attributes that are contained in the profile. The plug-in also has the sole responsibility for rendering the layout representation within the profile. After a computed profile has been defined for an asset type, the computed profile is treated in the same way as any other profile; for example, permissions for computed profiles can be granted in the same way as for standard profiles, and the ordering of profiles within a type definition is the same for computed profiles as for standard profiles.

➤ **To add a profile to a new asset type**

1. In CentraSite Control, go to **Administration > Types**.

2. In the **Types** tab, click **Add Asset Type**.

This opens the **Add Asset Type** wizard.

3. In panel 3 of the **Add Asset Type** wizard, click **Add Profile**.

This opens the **Add Profile** dialog box.

4. In the **Add Profile** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Profile Name	Type a display name for the new profile. This is the profile name that CentraSite displays in instances of this type in CentraSite Control and CentraSite Business UI. Ensure the display name is meaningful (for example, <code>Technical Notes</code> , not <code>tn</code>). The display name can contain any combination of characters, including spaces. You can change an profile's display name at any time.
Computed Profile	Select the Computed Profile check box to add a computed profile, and then select the implementation language from the drop-down list box. When you select this check box, in the Add Profile dialog box, the list of available attributes is replaced by the input field Profile Implementation Archive .
Profile Implementation Archive	Click Choose File to search for a particular archive file that contains the computed profile's definition, and upload the computed profile definition to CentraSite. For information on how to implement a computed profile for CentraSite Control, see <i>CentraSite Developer's Guide</i> .

5. *Non-computed profile*. To define the set of attributes that should be assigned to the new profile, and to rearrange the order of displaying the assigned attributes within the profile, select the attributes from the list shown in the box **Available Attributes**, then use the arrow buttons to copy, remove, and rearrange the attributes in the box **Assigned Attributes**. Click **OK**.

Inheriting Base Type Profiles, Lifecycle Models, and Policies

You can customize the type definition to inherit its base type's properties by using the **Inherit** options the **Edit Asset Type - Advanced Settings** dialog box in CentraSite Control.

To define inheritance in the type definition, you must have the Manage Asset Types permission.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

CentraSite provides the ability to inherit the base type's profiles, lifecycle models, and policies, and thus ensure that the base type's properties apply to all instances of the (derived) sub type.

When an asset is an instance of a sub type, the set of profiles that CentraSite applies to the asset instance depends on the sub type's **Inherit base type profiles** setting.

When the **Inherit base type profiles** option is enabled for a sub type, CentraSite inherits the profiles of the base type in addition to the profiles of the sub type to the asset instance. For example, when you enable this option for the sub type REST Service, an asset instance of this type will include both the profiles that are defined for that sub type REST Service and the profiles that are defined for its base type Service.

When this option is disabled for a sub type, CentraSite applies the profiles that are defined for that sub type; but it does not inherit the profiles of its base type. For example, when you disable this option for the sub type REST Service, asset instances of the REST Service type will simply include profiles that are defined for that type. Asset instances does not inherit the profiles that are defined for its base type Service.

The following table summarizes how the set of profiles that CentraSite applies for a type is affected by the state of the **Inherit base type profiles** option.

If the type's "Inherit Base Type Profiles" option is ...	Instances of the type have profiles of ...
ENABLED	Base Type — AND — Sub Type
DISABLED	Sub Type

By default, the **Inherit base type profiles** option is disabled for predefined asset types.

> To inherit base type profiles in sub types

1. In CentraSite Control, go to **Administration > Types**.
2. In the **Types** tab, right-click a (sub) type whose base type profiles you want to inherit, and click **Details**.
3. In the Asset Type Details page, click the **Edit** button.

This opens the **Edit Asset Type** wizard.

4. In panel 1 of the **Edit Asset Type** wizard, click **Advanced Settings**.

- In the **Edit Asset Type - Advanced Settings** dialog box, select the **Inherit ...** check box(es) as required.

Option	Description
Inherit basetype profiles	Inherits the profiles that are defined in the asset type definition.
Inherit basetype policies	Inherits the policies that are defined for the asset type definition.
Inherit basetype LCM	Inherits the lifecycle models (LCMs) that are defined for the asset type definition.

- Click **OK** to leave the **Edit Asset Type - Advanced Settings** dialog box.
- Click **Finish** to save the updated type definition.

After you have inherited the base type profiles, policies, and LCMs in the sub type definition, you can view the inherited profiles, policies, and LCMs. However, you cannot customize their contents.

Cloning Base Type Profiles

To clone base type profiles in the type definition, you must have the Manage Asset Types permission.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

CentraSite provides the ability to clone the base type's profiles, and thus ensure that the base type's profiles apply to all instances of the (derived) sub type. In addition to cloning the base type's profiles, it is also possible to create, edit, and delete any number of attributes and profiles in a sub type, without affecting its base type definition.

When the **Inherit base type profiles** check box is not already enabled in the **Edit Asset Type - Advanced Settings** dialog box of a sub type, CentraSite presents the **Clone Base Type Profiles** dialog box that prompts for cloning the base type profiles. After confirming the cloning functionality, the appropriate base type profiles is cloned in the sub type definition.

The **Clone Base Type Profiles** functionality lets you make a copy of the base type profiles in the sub type, and save it to the user defined profiles to enable certain permissions (view, modify, and delete) for one or more users on an instance of the particular type.

In addition, the profile cloning functionality enables you to customize the predefined profiles that are shipped with the base type.

➤ To clone base type profiles in sub types

- In CentraSite Control, go to **Administration > Types**.

2. In the **Types** tab, right-click a (sub) type whose base type profiles you want to inherit, and then click **Details**.
3. In the Asset Type Details page, click the **Edit** button.

This opens the **Edit Asset Type** wizard.
4. In panel 1 of the **Edit Asset Type** wizard, click **Advanced Settings**.
5. In the **Edit Asset Type - Advanced Settings** dialog box, clear the **Inherit base type profiles** check box.
6. Click **OK** to leave the **Edit Asset Type - Advanced Settings** dialog box.
7. In the **Clone Base Type Profiles** dialog box, click **Yes** to make a copy of the base type profiles in the sub type.
8. Click **Finish** to save the updated type definition.

After you have cloned the base type profiles in the sub type, you can customize the cloned profiles and its attributes as required.

Excluding Sub Types from CentraSite Business UI Search

You can customize the (base) type definition to exclude its sub types by using the **Exclude sub types** option in the **Edit Asset Type - Advanced Settings** dialog box in CentraSite Control.

To exclude sub types in the (base) type definition, you must have the Manage Asset Types permission.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

The **Exclude sub types** option for a base type determines whether instances of its (derived) sub types should be displayed in the CentraSite Business UI. You can enable or disable this option for each base type. When you enable this option, Business UI omits instances of its sub types from the Search Results page so users cannot search for such instances. When you disable this option, Business UI includes the instances of its sub types in the Search Results page.

➤ To exclude sub types from CentraSite Business UI search

1. In CentraSite Control, go to **Administration > Types**.
2. In the **Types** tab, right-click a (base) type whose sub types you want to exclude from the search in CentraSite Business UI, and click **Details**.

3. In the Asset Type Details page, click the **Edit** button.

This opens the **Edit Asset Type** wizard.

4. In the **Edit Asset Type - Advanced Settings** dialog box, select the **Exclude sub types from CentraSite Business UI search** check box.
5. Click **OK** to leave the **Edit Asset Type - Advanced Settings** dialog box.
6. Click **Finish** to save the updated type definition.

Viewing Asset Type List

You can view the list of types defined in CentraSite by using the **Type Management** page in CentraSite Control.

> To view the list of asset types

1. In CentraSite Control, go to **Administration > Types**.
2. To filter the list to see just a subset of the available asset types, type a partial string in the **Search** field.

CentraSite applies the filter to the **Name** column. The **Search** field is a type-ahead field, so as soon as you type any characters, the display is updated to show only those types whose name contains the specified characters. The wildcard character % is supported.

If you type...	CentraSite Displays
b	Names that contain b
bar	Names that contain bar
%	All names

The Type Management page provides the following information about each asset type:

Column	Description
Name	The name of the asset type.
Description	A short description about the asset type.

You can adjust the view to show or hide the individual column by using the **Select Columns** icon that is located in the upper-right corner of the Type Management page.

The shortcut menu of a particular asset type displays one or more actions that you can perform on that type.

Action	Description
Details	Displays the details page of the asset type.
Delete	Deletes the asset type.
Export	Exports the asset type from the registry to an archive file on the file system.
Impact Analysis	Helps to easily visualize the associations that exist between the asset type and registry objects.
Add to List	Adds the asset type to a list in My Favorites .
Add to Favorites	Adds a shortcut to the asset type you want to use routinely or otherwise keep close at hand.

Modifying Asset Type Details

To modify the details of a user-defined type, you must have the Manage Asset Types permission. Besides allowing you to modify user-defined types, this permission allows you to modify certain predefined types installed by CentraSite.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

Follow these general guidelines when modifying the details of an asset type definition:

- You can modify the type's display name, description, icons and advanced options. You cannot modify the type's **Schema Name** or its **Namespace** property. These two properties are set when the type is created and cannot be changed thereafter.
- With respect to attributes:
 - You can add new optional attributes to the type at any time.
 - You can add new required attributes if there are no existing instances of the type in the registry. If there are existing instances of a type, you can add a new required attribute of type `slot` or `classification`, but not of type `relationship` or `file`.
 - If you add a new attribute, regardless of whether it is optional or required, no automatic update of existing instances takes place. This prevents the potential degradation of performance that could arise from the automatic update of a large number of instances.
 - You cannot modify the data type for an attribute, but you can modify many of an attribute's other options. Be aware that certain attributes are not permitted if assigned (that is, non-empty) instances of the attribute are present in the registry.
 - You cannot delete an attribute from a type if an instance of the type exists with a value assigned to the attribute. In such a situation, you must first remove the attribute's value from each such instance before you can delete the attribute.

Note that the command line tool `CentraSiteCommand` provides support for removing an attribute, in cases where existing instances contain a value for the attribute.

- A sub type inherits all of its attributes from its base type. To add new attributes to a sub type, do one of the following:
 - Add the attributes to profiles of the base type and inherit the profiles in sub type, OR
 - Clone the profiles of the base type and then add the attributes to the cloned profiles in sub type.

You can selectively display these attributes on the profiles that you have defined in or cloned from the base type.

- You cannot delete attributes or edit the properties of attributes within the inherited profiles in the sub type. However, you can delete and edit attributes within cloned profiles in the sub type.
- You can add, modify, delete, rename, and reorganize the profiles associated with a base type at any time.
- You can modify, delete, rename, and reorganize the base type profiles associated with a sub type at any time, if the profiles are cloned from the base type.
- You cannot modify, delete or rename the base type profiles associated with a sub type, if the profiles are inherited from the base type.
- Modifying the cloned profiles and attributes in a sub type does not affect the profiles and attributes in the base type.
- When recloning the base type profiles within a sub type, any new attribute that is added to the base type, and any attribute that is already deleted from the sub type but still available in the base type, is cloned in the sub type.
- Recloning the base type profiles within a sub type does not affect the existing attributes in the sub type.
- Beginning with version 9.9, CentraSite provides the ability to define consumable asset types for an asset type. If you are upgrading to CentraSite 9.9 from a version of CentraSite that did not support the definition of consumable types, CentraSite assigns each virtual asset type (Virtual Service, Virtual XML Service, Virtual REST Service, Virtual OData Service) a default consumable type definition for the User, Group, and Application asset types.
- You cannot remove a consumable type from an asset type definition if an instance of the consumable type is already registered as consumer of an instance of the selected type. In such a situation, you must first unregister each such consuming instance before you can remove the consumable type.
- You can add custom attributes to the User and Organization types that are installed with CentraSite. The custom attributes enables you to include additional metadata about the organizations or users at your site. For example, if the organizations within your enterprise belong to specific affiliates, you might want the Organization object to include an attribute

that identifies the affiliate to which an organization belongs. The custom attributes are displayed in the **Attributes** profile of the Organization or User details page in the CentraSite Control.

Important:

- If you are modifying one of the predefined asset types installed with CentraSite, review the information in [“Introduction to Types” on page 182](#) before you begin. It explains the kinds of modifications that you can make to the predefined types.
- If you are using CentraSite in conjunction with other software products, for example, the products of the webMethods Product Suite or a third-party product, those products can add their own asset types to CentraSite. Be aware that CentraSite treats these types as user-defined custom types, which can be modified by an administrator with the appropriate permissions (just like any other custom type). Modifying or deleting these types in CentraSite can lead to inconsistencies or errors in the product that uses the type. For example, if you modify or delete a type that is used by the webMethods Product Suite, components such as the webMethods Integration Server may no longer be able to publish assets to CentraSite. To prevent these types of errors, *do not modify or delete any asset type on which other Software AG components or third-party products depend.*

> To modify the details of an asset type

1. In CentraSite Control, go to **Administration > Types**.
2. In the **Types** tab, right-click a type whose details you want to modify, and then click **Details**.
3. In the Asset Type Details page, click the **Edit** button.

This opens the **Edit Asset Type** wizard.

4. In panel 1 of the **Edit Asset Type** wizard, examine or modify the type's basic information, and its advanced settings as required.
5. In panel 2 of the **Edit Asset Type** wizard, examine or modify the type's attribute information as required.
6. In panel 3 of the **Edit Asset Type** wizard, examine or modify the type's profile information as required.
7. In panel 4 of the **Edit Asset Type** wizard, select one or more generic profiles you want to display in the details page of an asset instance of this type.
8. In panel 5 of the **Edit Asset Type** wizard, rearrange the order in which you want to display the generic profiles in the details page of an asset instance of this type.
9. Click **Finish** to save the updated type definition.

Deleting Asset Types

To delete a user-defined asset type, you must have the Manage Asset Types permission.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

Follow these general guidelines when deleting asset types in CentraSite:

- You can delete a type *only if there are no instances of that type in the registry*.
- The core asset types that belong to CentraSite are non-deletable. CentraSite does not allow you to delete these types, even if there are no instances of the selected type in the registry.

Important:

If you have selected several asset types where one or more of them are predefined types, you can use the **Delete** button to delete the types. However, as you are not allowed to delete predefined asset types, only types you have permission for is deleted. The same applies to any other types for which you do not have the required permission.

Important:

If you are using CentraSite in conjunction with other software products, for example, the products of the webMethods Product Suite or a third-party product, those products can add their own asset types to CentraSite. Be aware that CentraSite treats these types as user-defined custom types, which can be deleted by an administrator with the appropriate permissions (like any custom type). Deleting these types in CentraSite can lead to inconsistencies or errors in the product that uses the type. For example, if you delete a type that is used by the webMethods Product Suite, components such as the webMethods Integration Server may no longer be able to publish assets to CentraSite. To prevent these types of errors, *do not delete any asset type on which other Software AG components or third-party products depend*.

➤ To delete types

1. In CentraSite Control, go to **Administration > Types**.
2. In the **Types** tab, right-click a type you want to delete, and click **Delete**.

You can also select multiple types, click the **Actions** menu, and click **Delete**.

3. Click **OK** in the confirmation dialog box.

Adding an Association Type

To create a user-defined association type, you must have the Manage Asset Types permission.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

➤ To add an association type

1. In CentraSite Control, go to **Administration > Types**.
2. Click the **Association Types** tab.
3. In the **Association Types** tab, click **Add Association Type**.

This opens the **Add Association Type** wizard.

4. In the **Add Association Type** wizard, type the appropriate information for each of the displayed data fields.

In this field...	Do the following...
Name	Type a name for the new association. Be aware that this is the name that is given to attributes that use this association. Therefore, the name should be meaningful when used as an attribute name. For example, use an association name such as <i>Developed By</i> , not <i>developer association</i> . <ul style="list-style-type: none"> ■ An association name does not need to be unique within CentraSite. However, to reduce ambiguity, you should avoid giving multiple associations the same name. ■ An association name can contain any character (including spaces).
Forward Label	Specify the relationship of the source asset (the one in which the Relationship attribute resides) to one or more specified targets. If you are not specifying a name for the forward label, then CentraSite will treat the association name as the forward label.
Reverse Label	<i>Optional.</i> Specify the relationship of the specified targets to the source asset.

5. Click **OK**.

Modifying Association Type Details

To modify the details of a user-defined association type, you must have the Manage Asset Types permission. Besides allowing you to modify user-defined association types, this permission allows you to modify certain predefined association types installed by CentraSite.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

Follow these general guidelines when modifying the details of an association type:

- You can modify the association type's name at any time.
- You can modify an association type for relationship properties *only if there is no relationship attribute defined with it in the catalog*. After an association type has been assigned to an attribute, it can no longer be edited.

➤ To modify the details of an association type

1. In CentraSite Control, go to **Administration > Types**.
2. Click the **Association Types** tab.
3. Click an association type whose details you want to modify.
4. In the **Edit Association Type** dialog box, examine or modify the fields as required.
5. Click **OK** to save the updated type definition.

Deleting Association Types

To delete a user-defined association type, you must have the Manage Asset Types permission.

Note:

By default, users with the CentraSite Administrator role and Asset Type Administrator role have this permission.

Follow these general guidelines when deleting association types in CentraSite:

- You cannot delete an association type that CentraSite provides out-of-the-box (not even if you belong to a role with the Manage Asset Types permission).
- You can delete an association type *only if there is no relationship attribute defined with it in the catalog*. After an association type has been assigned to a relationship attribute, it can no longer be deleted.
- You cannot delete the stand-alone association type that is in use for defining an association in the catalog.

➤ To delete association types

1. In CentraSite Control, go to **Administration > Types**.
2. Click the **Association Types** tab.

3. In the **Association Types** tab, select one or more association types, and click **Delete**.
4. Click **OK** in the confirmation dialog.

Managing Types through Command Line Interface

This section describes operations you can perform to manage types through Command Line Interface.

Removing Attribute from Asset Type Definition

Pre-requisites:

To remove an attribute from a type definition through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

A user-defined attribute cannot be deleted from an asset type definition when there are asset instances of that type containing a value for the attribute.

CentraSite provides a command `remove Attribute` for this purpose. This command helps to automatically remove the attribute's value from all of the existing asset instances, before removing it from the type definition.

Follow these general guidelines when removing an attribute from the asset type definition:

- If you have one or more existing asset instances of this type, the remove operation of an attribute from the type definition can take some time to complete.
- You cannot remove a predefined attribute from any of the predefined asset types. You can, however, remove a custom (that is, user-defined) attribute from any predefined asset type.

➤ To remove an attribute from a type definition

- Run the command `remove Attribute`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd remove Attribute [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -assetType <ASSET-TYPE> [-attributeKind <ATTRIBUTE-KIND>] -attributeName <ATTRIBUTE-NAME>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .

Parameter	Description
PASSWORD	The password for the CentraSite user identified by the parameter USER-ID.
ASSET-TYPE	The name of the asset type, in the format "{<namespace of the asset type>}SchemaName".
ATTRIBUTE-KIND	A one-character code representing the type of the attribute you wish to remove. Allowable values are "C" for Classification, "R" for Relationship, "F" for File and "S" for all other attribute types. The use of the <code>attributeKind</code> parameter is optional.
ATTRIBUTE-NAME	The name of the attribute schema for attributes whose <i>AttributeKind</i> is "S". For attributes with an <i>AttributeKind</i> other than "S" it is the name of the attribute itself.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd remove Attribute -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-assetType {http://namespaces.CentraSite.com/Schema}XMLSchema -attributeKind S
-attributeName test_String_Attribute
```

The response to this command could be:

```
Executing the command : remove Attribute
Successfully executed the command : remove Attribute
```

Checking Sequence Order for Asset Profiles

Pre-requisites:

To check the sequence order of asset profiles through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

In the CentraSite Control, the sequence number plays a vital role in the profile display order and the associated instance-level profile permissions. In general, the sequence numbers assigned to the predefined profiles and computed profiles are odd numbers and even numbers are assigned to the user-defined profiles. But, some predefined profiles are designated with an even sequence number.

When a user creates a new profile, consider the system assigns an even sequence number that matches with one of the predefined profiles. When the user sets the profile-level permissions for the new profile, the same permissions are also assigned to the predefined profile that has the same even sequence number within the asset type. As a result, the user might get permissions to more profiles than intended.

CentraSite provides a Java tool named `FixProfileSequenceNumber.jar` for this purpose.

➤ To check the sequence order of asset profiles

- Run the Java tool `FixProfileSequenceNumber.jar`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd FixProfileSequenceNumber.jar <CentraSite URL> <admin user id> <password>`

The input parameters are:

Parameter	Description
CentraSite URL	The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
admin user id	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
password	The password for the CentraSite user identified by the parameter <code>admin user id</code> .

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd FixProfileSequenceNumber.jar http://localhost:53307/CentraSite/CentraSite Administrator manage
```

Cloning an Asset Type

Pre-requisites:

To clone an existing base type and create a new asset type through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a Java tool named `AssetTypeManager.jar` for this purpose.

> To clone an asset type

- Run the Java tool `AssetTypeManager.jar`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd|sh AssetTypeManager.jar <CentraSite URL> <admin user id> <password> clone <base type name> <clone base type name> <clone base type namespace>`

The input parameters are:

Parameter	Description
CentraSite URL	The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
admin user id	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .

Parameter	Description
password	The password for the CentraSite user identified by the parameter <code>admin user id</code> .
base type name	The name of an existing base type you want to use to clone a new asset type.
clone base type name	The name of the new asset type to be cloned.
clone base type namespace	The namespace of the new asset type to be cloned. The namespace must be in the following format: <code>http://namespaces.AssetName.com</code> Where, <code>AssetTypeName</code> is the name of the new asset type to be cloned.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd AssetTypeManager.jar
http://localhost:53307/CentraSite/CentraSite Administrator manage clone Service
ClonedService http://namespaces.ClonedService.com
```

Creating a Virtual Asset Type

Pre-requisites:

To create a new virtual asset type through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

The new virtual asset type will be created with the following type options:

- Visible in Asset Browse
- Visible in Search
- Inherit Base Type Profiles
- Inherit Base Type Policies
- Inherit Base Type LCM

CentraSite provides a Java tool named `AssetTypeManager.jar` for this purpose.

> To create a virtual asset type

- Run the Java tool `AssetTypeManager.jar`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd AssetTypeManager.jar <CentraSite URL> <admin user id> <password> create <virtual type name> <base type name>`

The input parameters are:

Parameter	Description
CentraSite URL	The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
admin user id	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
password	The password for the CentraSite user identified by the parameter <code>admin user id</code> .
virtual type name	Name of the new virtual asset type you want to create.
base type name	The fully qualified name of the existing base asset type.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd AssetTypeManager.jar
http://localhost:53307/CentraSite/CentraSite Administrator manage create
MyVirtualRESTService RESTService
```

Deleting a Virtual Asset Type

Pre-requisites:

To delete an existing virtual asset type through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a Java tool named `AssetTypeManager.jar` for this purpose.

> To delete virtual asset type

- Run the Java tool `AssetTypeManager.jar`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd AssetTypeManager.jar <CentraSite URL> <admin user id> <password> delete <virtual type name>`

The input parameters are:

Parameter	Description
CentraSite URL	The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
admin user id	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .

Parameter	Description
password	The password for the CentraSite user identified by the parameter admin user id.
virtual type name	Name of the virtual asset type you want to delete.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd AssetTypeManager.jar
http://localhost:53307/CentraSite/CentraSite Administrator manage delete
MyVirtualRESTService
```

Changing an Asset's Type

Pre-requisites:

To change the asset type of an existing asset through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a Java tool named `AssetManager.jar` for this purpose. The tool changes the type of an existing asset or a list of assets (saved search) to a specified user-defined virtual asset type or back to the specified base type.

➤ To change the type of an existing asset

- Run the Java tool `AssetManager.jar`.

The syntax is of the format:

- `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd AssetManager.jar <CentraSite URL> <admin user id> <password> set type <objectId> <type>`
- `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd AssetManager.jar <CentraSite URL> <admin user id> <password> set type <saved search> <type>`

The input parameters are:

Parameter	Description
CentraSite URL	The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
admin user id	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
password	The password for the CentraSite user identified by the parameter admin user id.
objectId	The ID of the asset whose type you want to change. You can specify the UDDI key of the asset.

Parameter	Description
	For example, <code>uddi:1e5aff10-f3e3-11df-86fc-a6e2fa0ea483e</code>
<code>saved search</code>	The fully qualified name of the saved search. If the saved search name contains white spaces, enclose the name within "".
<code>type</code>	The fully qualified name of the user-defined virtual type or the base type. If the type name contains white spaces, enclose the name within "".

Examples (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd AssetManager.jar  
http://localhost:53307/CentraSite/CentraSite Administrator manage set type  
uddi:1e5aff10-f3e3-11df-86fc-a6e2fa0ea483e Service
```

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd AssetManager.jar  
http://localhost:53307/CentraSite/CentraSite Administrator manage set type  
uddi:1e5aff10-f3e3-11df-86fc-a6e2fa0ea483e "My Saved Search" Service
```

7 Taxonomy Management

■ Introduction to Taxonomies	282
■ Managing Taxonomies through CentraSite Business UI	286
■ Managing Taxonomies through CentraSite Control	292

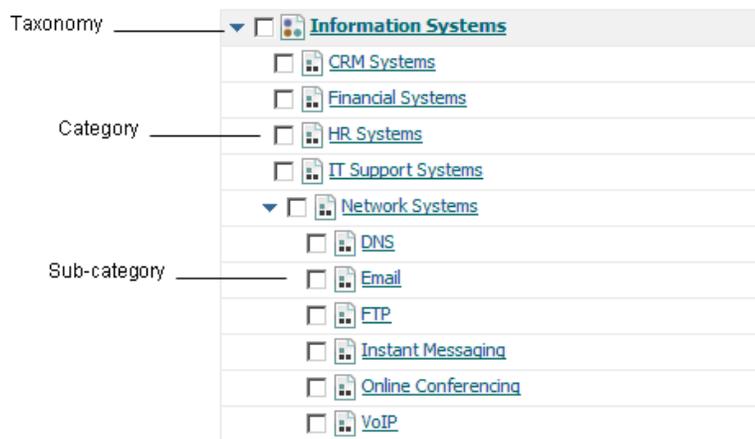
Introduction to Taxonomies

Note:

Beginning with version 9.10, managing taxonomies through CentraSite Control has been deprecated and will be removed in a future release. Software AG recommends that you use the newly introduced CRUD interface of CentraSite Business UI for creating and managing taxonomies.

A taxonomy is a hierarchical classification scheme. In CentraSite, you use taxonomies to classify objects in the registry. Taxonomies enable you to filter, group, and sort the contents of the registry.

A taxonomy consists of a name and zero or more *categories*. A category represents a classification within the taxonomy. A category can have multiple levels of sub-categories.



A *taxonomy* categorizes assets in CentraSite so that a consumer can search for assets within a particular category. CentraSite provides several predefined taxonomies, which are available to all users. Additionally, you can create custom taxonomies to suit your business needs and subdivide them by creating *categories*. Categories help consumers locate assets more easily. For example, if you are offering assets to help your consumers better manage their finances, classifying your assets under `personal banking` or `money management` helps them locate your assets easily.

To create and manage a taxonomy, you must belong to a role that has the Manage Taxonomies system-level permission. By default, users with the CentraSite Administrator or Asset Type Administrator role have this permission, and can assign this permission to other roles. The Manage Taxonomies permission enables you to manage (create, view, edit, and delete) any user-defined taxonomy (the predefined taxonomies provided by CentraSite can be modified in certain ways, but cannot be deleted).

Classifying Assets Using Taxonomies

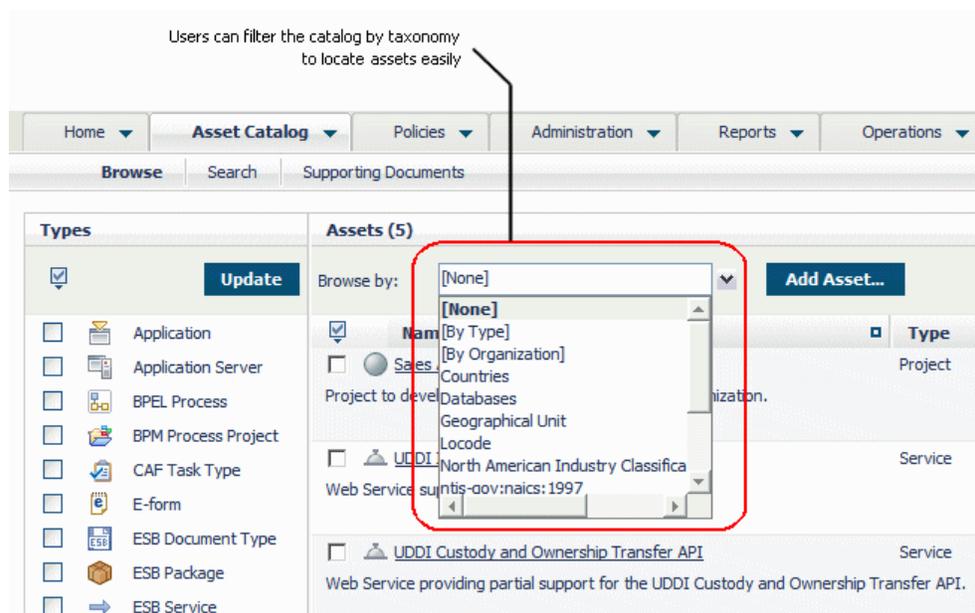
While publishing assets to CentraSite, they can be classified in two ways:

- By directly assigning values to an asset's *Classification attributes*. If an asset's type includes one or more Classification attributes, the asset can be classified by setting these attributes.

- By assigning ad hoc classifiers to an asset's *Classifications profile*. This profile enables the classification of an asset by any available taxonomy defined in CentraSite. It allows assigning classifiers to an asset in cases where the asset itself does not include any explicit Classification attributes or does not include the needed type of Classification attribute.

Using Taxonomies to Locate Assets

Classified assets are easier to locate because CentraSite includes convenient tools for filtering, reporting, and querying the registry by taxonomy. For example, the Browse page enables users to browse the asset catalog according to a specified taxonomy. The advanced search feature enables users to query the registry for assets that are classified in a particular way.



Note:

You must select the `Taxonomy is browsable` property to browse the asset catalog by a taxonomy. If this property is not selected, it does not appear in the **Browse by** catalog browser's drop-down list.

Using Taxonomies to Target the Execution of Design/Change-Time Policies

Design/Change-Time policies execute when events within the policy's scope occur in the registry. The scope of a policy specifies to which type of registry objects the policy applies (for example, Service objects, Policy objects, User objects) and during which type of events the policy is triggered (for example, a PreCreate event, a PostCreate event, a PreStateChange event).

Classifying assets helps you create highly targeted design/change-time policies, because the scope of a policy can be additionally constrained to objects that are classified in a specified way. For example, instead of applying a particular policy to all Application Server assets, you may restrict the policy to just the Application Server assets that are classified by the APAC category from the Domains taxonomy.

When you define a custom asset type, decide whether you need to apply different design/change-time policies to specific subsets of that type. If so, ensure that the asset type includes a Classification attribute that can be used to distinguish those subsets. (Consider making this a required attribute to ensure that users do not forget to classify assets of this type.)

The Scope of a Taxonomy

Like types, taxonomies are system-wide objects (that is, all organizations have access to the same global set of taxonomies). You cannot restrict a taxonomy to a specific organization.

Taxonomies are also visible to all users. All users (including guest users) can view the taxonomies defined within an instance of CentraSite.

Predefined Taxonomies

CentraSite installs a number of standard taxonomies that you can use to classify assets.

These include:

- ISO 3166 Country Codes
- North American Industry Classification System 2002 (NAICS)
- ThomasNet Supplier Registry
- Product and Service Category System: United Nations Standard Products and Services Code (UNSPSC)

CentraSite also includes a number of special-purpose taxonomies to support the use of CentraSite by products such as the webMethods product suite. These taxonomies belong to other products, which expect the taxonomy definitions to remain unchanged.

You cannot delete any of the predefined taxonomies installed with CentraSite or modify their category structure. Deleting these taxonomies in CentraSite can lead to inconsistencies or errors in the product that uses the taxonomy. For example, if you delete a taxonomy that is used by the webMethods product suite, components such as the webMethods API Portal may not be able to capture the original asset metadata from CentraSite. You can, however, modify certain attributes and properties for these taxonomies. Additionally, you can suppress them in the user interface. For example, if your users will never use the NAICS taxonomies that CentraSite provides, you can remove these taxonomies from the user interface.

Custom Taxonomies

In addition to using the taxonomies that CentraSite provides, you can create your own custom taxonomies.

When you include a Classification attribute in a type, you usually need to create a corresponding taxonomy for the attribute (unless the required taxonomy already exists in the registry). For example, let's say you decide that you want to classify your Application Server assets according to the domain in which they reside. To do this you would first create a custom taxonomy that identifies the various domains in your environment. Then, after the taxonomy exists, you would

customize the Application Server asset type and add a Classification attribute to it that enables users to classify application server assets by the Domain taxonomy.

Visibility of Taxonomies in CentraSite

All users have implicit (and irrevocable) view permission on taxonomies. This permission enables any user to access any taxonomy for the purpose of classifying or filtering registry objects. However, taxonomies have the following additional properties that control whether they are visible to users within the CentraSite Control and CentraSite Business UI.

Property	Description
Taxonomy is Browsable	<p>Determines whether the taxonomy appears in the taxonomy lists that CentraSite Control and CentraSite Business UI display to users for classifying or filtering objects. If this property is selected for a taxonomy, CentraSite Control and CentraSite Business UI include the taxonomy in the following lists:</p> <ul style="list-style-type: none"> ■ <i>Specific to CentraSite Business UI.</i> The Choose Taxonomies list in the Advanced Search panel (go to Browse > Narrow Your Results > Additional Search Criteria > Taxonomies > Choose). ■ <i>Specific to CentraSite Control.</i> The Browse by list in the Asset Catalog > Browse page. ■ The Add Classification dialog box displayed from the Classification profile. <p>The Taxonomy is Browsable check box is by default selected when you create a taxonomy. You might clear this checkbox, for example, if a taxonomy is intended to be used to classify objects programmatically (for example, using a policy). Doing this prevents end users from using the taxonomy to classify assets using the Classification profile in CentraSite Control and CentraSite Business UI.</p> <p>You can also use this property to suppress the display of the predefined taxonomies that CentraSite installs. For example, if your site has no need to use the NAICS taxonomies provided by CentraSite, you can eliminate them from the taxonomy lists displayed in the Taxonomies page by disabling their Taxonomy is Browsable property.</p>
Applicable to Object Types	<p><i>Specific to CentraSite Control.</i> When a taxonomy is browsable (that is, when its Taxonomy is Browsable checkbox is selected), the Applicable to Object Types property determines for which object types the taxonomy is displayed. For example, if you specify that a taxonomy is not applicable to XML Schemas, CentraSite Control does not include the taxonomy in the taxonomy lists that users see when they add classifiers to XML Schemas.</p>

Property	Description
Internal	<p><i>Specific to CentraSite Control.</i> Indicates whether a taxonomy is meant for use by end users. Taxonomies that are <i>internal</i> are designed to support CentraSite's own internal processes and are, therefore, suppressed from most taxonomy lists displayed in CentraSite Control.</p> <p>A taxonomy's Internal property cannot be assigned or viewed through CentraSite Control. It can only be accessed through the API.</p>

Managing Taxonomies through CentraSite Business UI

This section describes operations you can perform to manage taxonomies through CentraSite Business UI.

Adding Taxonomy

You can create a new taxonomy by using the Taxonomies page. You can access this page only if you have the Manage Taxonomies permission in CentraSite.

To create a new taxonomy in CentraSite, follow these steps:

> To add a taxonomy

1. In the CentraSite Business UI activity bar, click **Taxonomies**.

A list of defined taxonomies is displayed in the Taxonomies page. The Actions bar displays a set of actions that are available for working with taxonomies.
2. On the Actions bar of the Taxonomies page, click **Add Taxonomy**.
3. In the Create New Taxonomy page, provide the required information for each of the displayed data fields.

Field	Description
Taxonomy Name	<p>Name of the taxonomy.</p> <p>This is the name that users will see when they search for taxonomies in CentraSite.</p> <p>A taxonomy name can contain any characters (including spaces), and must be unique within the CentraSite registry.</p>
Description	<p><i>Optional.</i> The description for the role.</p> <p>This description appears when a user displays the list of taxonomies in the Taxonomies page.</p>

Field	Description
Taxonomy is browsable	<i>Optional.</i> Select the Taxonomy is browsable check box if you want the taxonomy to be visible to users for filtering and classification purposes.

4. Click **Save**.

The newly created taxonomy is added to the CentraSite registry or repository.

Viewing the Taxonomy List

You can view the list of taxonomies in CentraSite by using the Taxonomies page. You can access this page only if you have the Use the Administration UI permission.

The Taxonomies page displays all the browsable taxonomies.

> To view the list of taxonomies

1. In the CentraSite Business UI activity bar, click **Taxonomies**.

A list of defined taxonomies is displayed in the Taxonomies page.

For each taxonomy, the list includes various attributes of the taxonomy. The attributes include:

Column	Description
Name	Name of the taxonomy.
Description	The description for the taxonomy.
Last Updated	The date and time the taxonomy was last modified.

You can adjust the view to show or hide any of the available attributes by opening the drop-down list labeled **View** and selecting the attributes that you want to include in the view, and clearing the attributes that you do not want to view.

You can change the order in which the attributes are displayed by opening the drop-down list labeled **Sort by**. The list displays all the attributes that are selected in the **View** drop-down list. The order in which the attributes appear in the **Sort by** drop-down list is the order in which the attributes appear for each displayed user. To change the order in which any given attribute is displayed, select the attribute in the drop-down list **Sort by** and use the arrows to move the attribute to the required position.

In the Taxonomies page, an action bar displays one or more icons that you can use to perform various tasks on the selected taxonomies.

Action	Description
Add Taxonomy	Creates a new taxonomy in the CentraSite registry.
Delete	Deletes one taxonomy or multiple taxonomies in a single operation.

Viewing List of Assets Classified by Taxonomies from the Search Results Page

You can view a list of assets classified by taxonomies of a particular classification by using the Search Results page.

➤ To view the list of assets classified by taxonomies from the Search Results page

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link that is located in the upper-left corner of the menu bar.
 - Click the **Search** icon that is located next to the **Scope** list. The default search scope is **Assets**.

2. In the **Additional Search Criteria** list, select **Taxonomies**.

3. To search for the list of taxonomies, click **Choose**.

This opens the **Choose Taxonomies** dialog box.

4. In the **Choose Taxonomies** dialog box, follow these steps:

- a. Click the chevron next to taxonomy option button as required.

A list of defined classifications for the particular taxonomy is displayed.

- b. In the list of taxonomy classifications, select the check box for one classification, or select the check boxes for multiple classifications.

- c. Click **OK**.

A list of assets classified by taxonomies is displayed in the Search Results page.

5. To filter the list to see just a subset of the available taxonomies, type a partial string in the **Keyword** text box. Add the specified keyword to the Search Recipe, by clicking the plus symbol next to the text box, or press Enter.

The Search Results page provides the following information about each taxonomy:

Column	Description
Name	Name of the taxonomy.
Description	The description for the taxonomy.
Last Updated	The date on which the taxonomy was last modified.

You can adjust the view to show or hide any of the available attributes by opening the drop-down list labeled **View** and selecting the attributes that you want to include in the view, and clearing the attributes that you do not want to view.

You can change the order in which the attributes are displayed by opening the drop-down list labeled **Sort by**. The list displays all the attributes that are selected in the **View** drop-down list. The order in which the attributes appear in the **Sort by** drop-down list is the order in which the attributes appear for each displayed user. To change the order in which any given attribute is displayed, select the attribute in the drop-down list **Sort by** and use the arrows to move the attribute to the required position.

Viewing Taxonomy Details

You can view the detailed taxonomy information by using the Taxonomy Details page. You can open this page only if you have the Use the Administration UI permission.

➤ To view the details of a taxonomy

1. In the CentraSite Business UI activity bar, click **Taxonomies**.

A list of defined taxonomies is displayed in the Taxonomies page.

2. In the list of taxonomies, click a taxonomy for which you want to display the details.

The Taxonomy Details page is displayed.

3. To expand or collapse the categories, click the chevron next to the category name.

Modifying Taxonomy Details

You can modify an existing taxonomy information by using the Taxonomy Details page. You can access this page only if you have the Use the Administration UI permission.

You can modify taxonomies only if you have Manage Taxonomies permission. The modification is broken up across different profiles in the Taxonomy Details page, which means that modifications done in each profile are independent of each other and must be saved individually. The modifications you can perform in each profile is outlined in the subsequent sections.

➤ To modify the details of a taxonomy

1. In the CentraSite Business UI activity bar, click **Taxonomies**.

A list of defined taxonomies is displayed in the Taxonomies page.

2. Click a taxonomy for which you want to modify the attributes.

The Taxonomy Details page is displayed. Also, the Actions bar displays a set of actions that are available for working with the displayed taxonomy.

3. In the Taxonomy Details page, change values of the attributes in the respective fields as required.

4. To add categories and subcategories to the taxonomy, do the following:

- a. On the Actions bar of the taxonomy, click the **Add Category** icon.

This opens the **Add Category** dialog box.

- b. In the **Add Category** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Category Name	The name of the category. This is the name that users will see in the taxonomy hierarchy when they view the list of assets using the Choose Taxonomies list. The name must be meaningful (for example, <i>Business Application Systems</i> , not <i>BAS</i>).

Note:

A category name does not need to be unique within the taxonomy. However, to reduce ambiguity, you should avoid giving multiple categories the same name.

Description *Optional.* The description for the category.

- c. Click **OK**.
 - d. To expand or collapse the categories, click the chevron next to the category name.
5. To further add sub-categories, modify, or delete categories that are displayed in the **Categories** section, do the following:

■ **Add a Sub-Category**

1. In the list of categories, hover over the category to add a sub-category.

This displays icons for one or more actions that you can perform on the category.

2. Click the **Add** icon.

3. In the **Add Category** dialog box, provide the required information for the displayed data fields.
4. Click **OK**.

Repeat for each category you want to add.

■ **Modify a Category**

1. In the list of categories, hover over the category whose details you want to modify.
This displays icons for one or more actions that you can perform on the category.
2. Click the **Edit** icon.
3. In the **Edit Category** dialog box, provide the required information for the displayed data fields.
4. Click **OK**.

Repeat for each category you want to modify.

■ **Delete a Category**

1. In the list of categories, hover over the category that you want to delete.
This displays icons for one or more actions that you can perform on the category.
2. Click **Delete**.
3. Click **OK** in the confirmation dialog box.

Repeat for each category you want to delete.

6. After you have made the required changes, click **Save** to update the taxonomy information.

Deleting Taxonomies

Pre-requisites:

You can access this page only if you have the Use the Administration UI permission. Also, you can delete taxonomies only if you have the Manage Taxonomies permission.

You cannot delete predefined taxonomies and taxonomies with categories.

➤ **To delete taxonomies from the taxonomies page**

1. In the CentraSite Business UI activity bar, click **Taxonomies**.
A list of defined taxonomies is displayed in the Taxonomies page.
2. In the Taxonomies page, select one or multiple taxonomies.

3. On the Actions bar of the Taxonomies page, click **Delete**.

You can also delete a taxonomy from its details page.

4. Click **OK** in the confirmation dialog box.

Each selected taxonomy is permanently removed from the CentraSite registry repository.

Managing Taxonomies through CentraSite Control

This section describes operations you can perform to manage taxonomies through CentraSite Control.

Adding Taxonomy

You can access this page only if you have the Use the Administration UI permission in CentraSite. Also, you can create a taxonomy only if you have the Manage Taxonomies permission.

> To add a taxonomy

1. In CentraSite Control, go to **Administration > Taxonomies**.
2. In the **Taxonomies** page, click **Add Taxonomy**.
3. In the area labeled **Taxonomy Information**, provide the required information for each of the displayed data fields.

Field	Description
Name	<p>Name of the taxonomy.</p> <p>This is the name that users see when they search for assets using the Browse catalog page. The name must be meaningful (for example, Software AG ASIA-PAC and not SAG-AP).</p> <p>A taxonomy name can contain any character (including spaces), and must be unique within the CentraSite registry.</p>
Description	<p><i>Optional.</i> The description for the taxonomy.</p> <p>This description appears when a user displays the list of taxonomies in the Taxonomies page.</p>
Documentation	<p><i>Optional.</i> Name of the file that contains additional external documentation for the taxonomy.</p> <p>You can use Browse option to locate the file.</p>

Field	Description
Taxonomy is browsable	<i>Optional.</i> Select this check box to allow the taxonomy to be visible to users for filtering and classification purposes in CentraSite Control.
Icon	<i>Optional.</i> The location of a bitmap file that contains an icon that is displayed in the user interface to identify this taxonomy. The bitmap format must be either JPG, GIF, or PNG and it must be 16x16 pixels, to match the size of the existing taxonomies.

4. To specify the object types that are allowed to use the taxonomy for association purposes, click **Applicable to Object Types** and perform the following:
 - a. Click **Applicable to Object Types**.
 - b. In the **Applicable to Object Types** dialog box, select the check box for one object type, or select the check boxes for multiple object types that you want to associate with the taxonomy.
 - c. Click **OK**.
5. To enable other users to view, modify, and delete a taxonomy that you have created, you must modify the taxonomy's permission settings. To do this, click the **Permissions** tab and perform the following:
 - a. Click **Add Users/Groups**.
 - b. In the **Add Users/Groups** dialog box, select one or more users and groups to which you want to assign permissions.
 - c. To filter the list of users and groups, type a partial string in the **Search** field.

CentraSite applies the filter to the **Name** column.
 - d. Click **OK**.
 - e. In the **Permissions** tab, select the **View**, **Modify**, and **Full** check boxes to assign appropriate permissions to each user and group in the **Users/Groups** list.
 - f. Click **OK**.
6. Click **Save**.

The newly created taxonomy is added to the CentraSite registry or repository.

Viewing the Taxonomy List

You can access this page only if you have the Use the Administration UI permission in CentraSite. Also, you can view a list of taxonomy only if you have the Manage Taxonomies permission.

By default, users in the CentraSite Administrator or Asset Type Administrator role have this permission, although an administrator can grant this permission to other roles.

The Taxonomies page, by default, displays all the browsable taxonomies in CentraSite. To display all the available taxonomies in CentraSite, select the **Show all Taxonomies** button.

> To view the list of taxonomies

- In CentraSite Control, go to **Administration > Taxonomies**.

A list of the defined taxonomies is displayed in the Taxonomies page.

The Taxonomies page provides the following information about each taxonomy:

Column	Description
Name	Name of the taxonomy.
Description	The description for the taxonomy.

You can adjust the view to show or hide the individual column by using the **Select Columns** icon that is located in the upper-right corner of the Taxonomies page.

The shortcut menu of a particular taxonomy displays one or more actions that you can perform on that taxonomy.

Action	Description
Details	Displays the details page of the taxonomy.
Delete	Deletes the taxonomy.
Add Category	Adds a category or a subcategory to the taxonomy.
Export	Exports the taxonomy from the registry to an archive file on the file system.
Impact Analysis	Helps to easily visualize the associations that exist between the taxonomy and registry objects.
Add to List	Adds the taxonomy to an existing list.
Add to Favorites	Adds a shortcut to the taxonomy you want to use routinely or otherwise keep close at hand.

Viewing Taxonomy Details

You can access this page only if you have the Use the Administration UI permission in CentraSite. Also, you can view taxonomy details only if you have the Manage Taxonomies permission.

➤ To view the details of a taxonomy

1. In CentraSite Control, go to **Administration > Taxonomies**.
2. Right-click a taxonomy for which you want to display the details, and click **Details**.

The Edit Taxonomy page is displayed.

The area labeled **Taxonomy Information** displays the generic attributes that includes details about basic taxonomy information - Name, Description, Organization, and so on.

The taxonomy details are displayed in the following tabs:

- **Applicable to Object Types:** Displays a list of object types that are associated with the taxonomy.

In the **Applicable to Object Types** tab, you can associate object types with the taxonomy and revoke the existing associations with the taxonomy.

- **Permissions:** Displays a list of permissions that are defined for the taxonomy.

In the **Permissions** tab, you can assign permissions to users or groups and remove permissions from users or groups for the taxonomy.

Modifying Taxonomy Details

You can access this page only if you have the Use the Administration UI permission in CentraSite. Also, you can modify taxonomy details only if you have the Manage Taxonomies permission.

You can perform the taxonomy modification tasks from the Edit Taxonomy page. The modification is broken up across the different tabs in the Edit Taxonomy page, which means that modifications done in each tab are independent of each other and must be saved individually. The modifications you can perform in each profile is outlined in the subsequent sections.

➤ To modify the details of a taxonomy

1. In CentraSite Control, go to **Administration > Taxonomies**.
2. Right-click a taxonomy whose details you want to modify, and click **Details**.

This opens the Edit Taxonomy page.

3. In the **Taxonomy Information** section, change values of the attributes in the respective data fields as required.

4. Right-click a taxonomy whose details you want to modify, and click **Details**.

This opens the Edit Taxonomy page.

5. To modify the permission assignments, click **Permissions**, and then follow the instructions in [“Setting Permissions on Taxonomy” on page 297](#).

6. To add categories and subcategories to the taxonomy, follow the instructions in [“Adding Taxonomy Category” on page 300](#).

7. To examine and modify the properties of categories and subcategories for the taxonomy, follow the instructions in [“Modifying Taxonomy Category Details” on page 301](#).

8. To delete categories and subcategories of the taxonomy, follow the instructions in [“Deleting Taxonomy Categories” on page 301](#).

9. After you have made the required changes, click **Save** to update the taxonomy information.

Associating Taxonomy to Object Types

You can access this page only if you have the Use the Administration UI permission in CentraSite. Also, you can associate taxonomy to object types only if you have the Manage Taxonomies permission.

Note:

This functionality applicable only to CentraSite Control has been marked as deprecated and is removed in a future release.

You can associate a taxonomy to multiple object types. When you do this, you can classify objects of the specified types with the associated taxonomy.

The following general guidelines apply when associating a taxonomy with multiple object types:

- You can either select the check boxes for the object types individually, or select them all by using the **All Object Types** check box.
- By default, CentraSite displays the **All Object Types** check box as selected. When this check box is selected, the check boxes for the list of object types are automatically displayed as selected and disabled.

This default selection does not associate the taxonomy with the selected object types. Instead, it only allows the taxonomy to be applied to the selected object types (that is, CentraSite allows you to classify the selected object types using the taxonomy).

- If you clear the **All Object Types** check box and select ALL of the object types as selected, then there does not exist an association and the taxonomy is applicable to all of the selected object types.
- If you clear the **All Object Types** check box and the list of ALL the object types, then there does not exist an association and the taxonomy is still applicable to all the object types.
- Therefore, to associate a taxonomy with one or more object types, you must:
 1. Clear the **All Object Types** check box.
 2. Select the check box of the object types that you want to associate with the taxonomy.

➤ **To associate a taxonomy with object types**

1. In CentraSite Control, go to **Administration > Taxonomies**.
2. Right-click a taxonomy that you want to associate with the object types, and click **Details**.
This opens the Edit Taxonomy page.
3. Click the **Applicable to Object Types** tab.
4. In the **Applicable to Object Types** tab, select one or multiple object types to associate with this taxonomy.

Note:

To associate a taxonomy with all of the asset types (including new asset types that might be added to the registry in the future), select the **All Object Types** option. Note that this option associates the taxonomy with asset types only. It does not associate a taxonomy with non-asset types, such as Organizations, and Users, that can also be used with the taxonomy.

5. Click **Save**.

Setting Permissions on Taxonomy

You can access this page only if you have the Use the Administration UI permission in CentraSite. Also, you can view a list of taxonomy only if you have the Manage Taxonomies permission.

Note:

This functionality applicable only to CentraSite Control has been marked as deprecated and is removed in a future release.

All CentraSite users are permitted to view the taxonomies that you create. However, only you (as the owner of the taxonomy) and users who belong to a role with the Manage Taxonomies permission are allowed to modify or delete these taxonomies. To enable other users to modify and delete a taxonomy that you have created, you must modify the taxonomy's instance-level permission settings.

The following general guidelines apply when setting permissions on taxonomies:

- You can assign permissions to any individual user or group defined in CentraSite.
- The groups to which you can assign permissions include the following system-defined groups:

Group Name	Description
Users	All users within a specified organization.
Members	All users within a specified organization and its child organizations.
Everyone	All users of all organizations and child organizations, <i>including guest users</i> (if your CentraSite permits access by guests).

- If a user is affected by multiple permission assignments, the user receives the union of all the assignments. For example, if group ABC has Modify permission on a taxonomy and group XYZ has Full permission on the same taxonomy, users that belong to both groups will, in effect, receive Full permission on that taxonomy.
- If you have assigned users the Modify or Full permission on a taxonomy and you want them to be able to manage (modify or delete) the template in CentraSite Control, ensure that the users have Use the Administration UI permission.

➤ To assign permissions to a taxonomy

1. In CentraSite Control, go to **Administration > Taxonomies**.
2. Right-click a taxonomy you want to assign the user and group permissions, and click **Details**.
3. In the Edit Taxonomy page, click the **Permissions** tab.
4. In the **Permissions** tab, click **Add Users/Groups**.
5. In the **Add Users/Groups** dialog box, select one or more users and groups to which you want to assign permissions.
6. To filter the list, type a partial string in the **Search** text box.
7. Click **OK**.

CentraSite applies the filter to the **Name** column.

Examples

String	Description
b	Displays names that contain b.

String	Description
bar	Displays names that contain bar.
%	Displays all users and groups.

- In the **Permissions** tab, assign specific permissions to each user and group in the **Users/Groups** list as follows:

Permission	Allows the selected user or group to...
View	View the taxonomy. Note: Disabling this permission does not prevent a user from accessing the taxonomy. CentraSite implicitly grants users view permission on all taxonomies. The implicit permission granted by CentraSite is not revoked by disabling the View permission on this tab.
Modify	View and edit the taxonomy.
Full	View, edit, and delete the taxonomy. This permission also allows the selected user or group to assign instance-level permissions to the taxonomy.

- Click **Save**.

Deleting Taxonomies

You can access this page only if you have the Use the Administration UI permission in CentraSite.

You cannot delete predefined taxonomies.

➤ To delete taxonomies

- In CentraSite Control, go to **Administration > Taxonomies**.
- Right-click a taxonomy that you want to delete, and click **Delete**.

You can also select multiple taxonomies, click the **Actions** menu, and click **Delete**.

- Click **OK** in the confirmation dialog box.

Each selected taxonomy is permanently removed from the CentraSite registry or repository.

Adding Taxonomy Category

You can access this page only if you have the Use the Administration UI permission in CentraSite.

Note:

This functionality applicable only to CentraSite Control has been deprecated and will be removed in a future release.

You can subdivide a taxonomy by creating categories and subcategories.

The following general guidelines apply when adding categories to a taxonomy:

- The **Add Category** item is visible only if the following conditions are met:
 - If the taxonomy has already been categorized with one or more child taxonomies (categories).
 - If the taxonomy does not have a child category and has never been classified for any of its applicable object types.
- The **Add Category** item is not visible if the following conditions are met:
 - If the taxonomy does not include a child category and is classified for at least one of its applicable object types.
 - If the taxonomy and its category are classified for at least one applicable object type.
 - If the taxonomy is related to any one of the lifecycle model.

> To add a category to a taxonomy

1. In CentraSite Control, go to **Administration > Taxonomies**.
2. Right-click a taxonomy for that you want to add a category, and click **Add Category**.
3. In the **Add Category** dialog, provide the required information for each of the displayed data fields.

Field	Description
Name	<p>Name of the category.</p> <p>This is the name that users will see in the taxonomy hierarchy when they view assets using the Browse page. The name must be meaningful (for example, Software AG ASIA-PAC, not SAG-AP).</p> <p>A category name can contain any characters (including spaces), and must be unique within a taxonomy hierarchy.</p>
Description	<p><i>Optional.</i> The description for the category.</p>

Field	Description
Icon	<i>Optional.</i> The location of a bitmap file that contains an icon that is displayed in the user interface to identify this new category. The bitmap format must be either JPG, GIF or PNG; it must be 16x16 pixels, to match the size of the existing category.

4. Click **OK**.

The newly created category is added to the taxonomy.

5. To expand or collapse the categories, click the chevron next to the taxonomy name.

Modifying Taxonomy Category Details

You can access this page only if you have the Use the Administration UI permission in CentraSite.

Note:

This functionality applicable only to CentraSite Control has been deprecated and will be removed in a future release.

> To modify the details of a taxonomy category

1. In CentraSite Control, go to **Administration > Taxonomies**.
2. In the list of taxonomies, click the chevron next to the taxonomy name to expand the list of categories.
3. Right-click a category whose details you want to modify, and click **Details**.
4. In the **Edit Category** dialog box, change values of the attributes in the respective data fields as required.
5. Click **OK** to save the updated taxonomy category information.

Deleting Taxonomy Categories

You can access this page only if you have the Use the Administration UI permission in CentraSite.

Note:

This functionality applicable only to CentraSite Control has been deprecated and will be removed in a future release.

The following general guidelines apply when deleting categories of a taxonomy:

- To delete a category, you must first delete all of the sub-categories underneath it.

- You cannot delete a category if objects are currently classified by the category.
- You cannot delete categories from the predefined taxonomies provided by CentraSite (not even if you belong to a role with Manage Taxonomies permission).

➤ **To delete categories**

1. In CentraSite Control, go to **Administration > Taxonomies**.
2. In the list of taxonomies, click the chevron next to the taxonomy name to expand the list of categories.
3. Right-click a category that you want to delete, and click **Delete**.

You can also select multiple categories, and click **Delete**

4. Click **OK** in the confirmation dialog box.

Each selected category is removed from the taxonomy.

8 Lifecycle Management

■ Introduction to Lifecycle Management	304
■ Lifecycle Model for Lifecycle Models (LCM for LCMs)	307
■ Predefined Lifecycle Models	308
■ Customizing Lifecycle Models	308
■ Updating Assets that are Under Lifecycle Management	314
■ Reverting an Asset That is Under Lifecycle Management to a Previous State	315
■ Managing Lifecycle Models through CentraSite Business UI	316
■ Managing Lifecycle Models through CentraSite Control	335

Introduction to Lifecycle Management

Lifecycle management is of importance to every enterprise that wants to implement a process-driven SOA with emphasis on adaptability, service reuse, and improvement. The lifecycle management (LCM) system for CentraSite helps to:

- Assess change impact and manageability across all service consumers.
- Ensure service quality through an integrated lifecycle approval process.
- Enable a single viewpoint for service stages and their artifacts.

Lifecycle Models

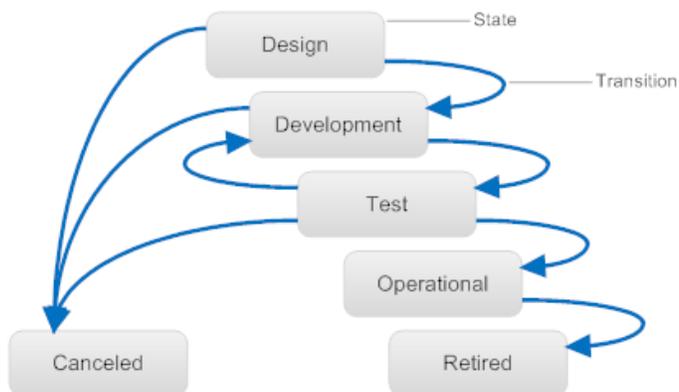
A *lifecycle model* describes the distinct steps through which a particular type of SOA asset passes from conception to retirement. In other words, a lifecycle model enables you to classify assets according to the state they have reached in their lifecycle. It also provides the basis for CentraSite's lifecycle governance capabilities. Using these capabilities, you can steer assets through the different steps of their lifecycle and apply governance controls at significant stages of the lifecycle process.

A lifecycle model is composed of *states* and *transitions*.

A *state* represents a distinct step through which an asset passes on its way from conception to retirement. A very simple lifecycle model for an asset might include states such as Design, Development, Test, Operational, and Retired. It might also include the Canceled state for assets whose lifecycle is terminated prior to completion.

A *transition* represents the act of switching an asset from one state to another. When you define a lifecycle model, you specify both the states that make up the lifecycle and the transitions that can occur from each state. For example, the Test state might have possible transitions to the Operational, Development, and Canceled states. If these are the only three transitions that you define for the Test state, then these are the only states to which CentraSite allows an asset in the Test state to be switched.

Lifecycle models are composed of states and transitions



Lifecycle Models Helps You Organize Your Assets

When you apply a lifecycle model to an asset type, the lifecycle model itself is treated as a taxonomy by many of CentraSite's browse and search tools. For example, using the catalog browser in CentraSite Business UI and CentraSite Control, you can view the contents of your catalog according to a specified lifecycle model. This view enables you to quickly ascertain which assets are in a particular phase of their development lifecycle.

CentraSite's search tools and reporting features also allow you to query assets by lifecycle state. You can use the advanced search feature, for example, to filter assets by their lifecycle state. You can also use CentraSite's reporting facility to examine the lifecycle status of the assets in your catalog.

Lifecycle Models Help You Govern Your Assets

Lifecycle models enable you to govern your assets more effectively by allowing you to enforce governance controls at various points in an asset's lifecycle.

When you associate a lifecycle model with an asset type, you make it possible to impose design/change-time policies at each step of the asset's lifecycle. These policies enable you to control the transition of an asset from one step of its lifecycle to another by triggering review and approval processes, issuing email notifications, updating permission settings and generally verifying that an asset meets the requirements necessary to enter the next step in its lifecycle.

System-wide Lifecycle Models

System-wide models, also called *global* models, apply to objects that can be owned by any organization. Global lifecycle models do not belong to any organization, but apply to all organizations. Global lifecycle models may have organization-specific policies attached to them.

System-wide models have precedence over organization-specific models, when a system-wide model for an object type (set of types) exists, it takes priority over all other models. However, if an organization-specific model already exists and a system-wide model is added, then the organization-specific model still exists and the objects that are in this model completes their lifecycle without effects. Only new objects is assigned to the new system-wide model.

Organization-specific Lifecycle Models

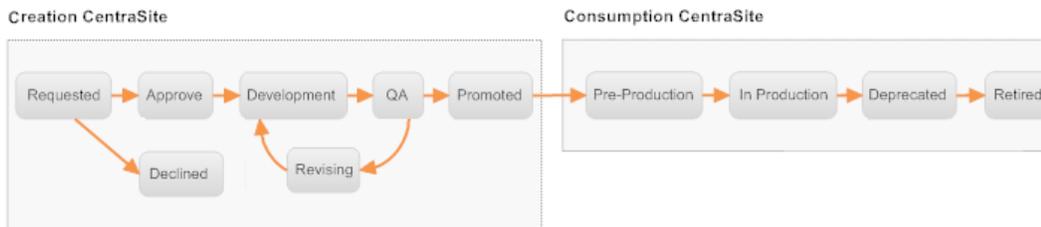
Organization-specific models, also called *organizational* models, apply only to objects that are owned by a specific organization. Each organizational lifecycle model belongs to an organization. Organizational lifecycle models are technically hierarchical, as organizations may contain sub-organizations.

For a given object type (say Service), each organization can define and activate its own lifecycle model, so that there are several models that can control the Service type. However, per organization only one lifecycle model can be active. When a new Service instance is created, then the user creating the instance belongs to an organization and that organization's active lifecycle model is used to control the particular Service instance.

It is possible to define a lifecycle model which consists of multiple nodes (registries) to make up one overall model. Each node only knows the model for the current node. It also knows the nodes that make up the complete lifecycle model and where they are.

Lifecycle Stages

Sometimes an asset's overall lifecycle is split across two or more registries. The most common example of this occurs when assets that are in the development and test phases of their lifecycle are maintained in one registry (the creation CentraSite) and assets that are deployed (that is, in production) are maintained in a separate registry (the consumption CentraSite).



When a site splits an asset's lifecycle across multiple registries, each participating registry is referred to as a *stage*. Each stage knows about the other participating stages, but does not know the details of the lifecycle that takes place in those stages (that is, the registries that participate in the overall lifecycle are not aware of the specific states and transitions that occur in the other registries).

To model a lifecycle that extends across multiple registries, you must create a separate lifecycle model on each participating registry. Each model describes just the segment of the lifecycle that occurs within its own registry. For example, in the multi-stage lifecycle depicted above, the lifecycle model on the creation registry would consist of the Requested, Declined, Approve, Development, Revising, QA, and Promoted states. The lifecycle model on the consumption registry would consist of the Pre-Production, In Production, Deprecated, and Retired states.

To indicate that a lifecycle ends on one registry and continues on another, the state that represents the end of an asset's lifecycle on a particular registry includes a pointer to the registry that hosts the next stage of the lifecycle.

Important:

Only an end state in a lifecycle model can have a pointer to another stage.

When an asset reaches the end of its lifecycle on one registry, you promote the asset to the next stage of its lifecycle by exporting the asset from the current registry and importing it to the next.

When you import the asset into the registry that hosts the next stage of its lifecycle, CentraSite verifies that the asset is being imported into the correct stage. It does this by checking the address specified in the stage parameter that is included in the archive file with the exported asset. If the address identified in the stage parameter in the archive file matches the registry's own address, CentraSite allows the asset to be imported.

Lifecycle States

The registry stages are broken down into states for lifecycle-enabled objects like services. This way the lifecycle process can be split into smaller steps, each supported and controlled by its responsible stage registry. States are connected to each other by allowable transitions. An approval process can define the conditions and activities to set a service from one state to another. CentraSite comes with the following:

- A set of object states, associated to each registry stage.
- A state-transition proposal to control the lifecycle.
- An example for collaboration management by roles, rights and notifications.

By using stages and states, you ensure that the registry object always has one defined state, even if multiple registries exist. State transitions are restricted, this ensures that the registry object had passed the lifecycle process correctly and achieved the required quality.

When moving from one state to the next, the following situations can exist:

- The required next state is in the same stage as the current state. In this case, the state change can be performed directly on the registry object from the user interface.
- The required next state is not in the same stage as the current state. In this case, the object must be exported (where necessary, along with its associated objects). In the exported archive, the current state remains unchanged. On importing the archive to the new stage, the state is set automatically to the appropriate new state.

Lifecycle Model for Lifecycle Models (LCM for LCMs)

CentraSite provides a lifecycle model for lifecycle models, which consists of three states: *New*, *Productive*, and *Retired*. Predefined policies associated with the LCM for LCMs activate and deactivate a lifecycle model, as appropriate, when you switch the lifecycle model's lifecycle state. The model is also structured in a way that allows CentraSite to automatically deactivate an old version of a lifecycle model when a new one is activated.

Normally, there is no need to change the LCM for LCMs and you should only consider modifying it if you have a compelling reason to do so. Because of the complex nature of the LCM for LCMs and its associated policies, any changes you make should be limited to the ones described below.

You cannot change the original version of the predefined LCM for LCMs, but you can create a new version of the LCM for LCMs and modify the new version while it is still in the New lifecycle state. Then set the lifecycle state of the new version to Productive, which activates the new version and deactivates the old version.

Here are the changes you can make in a new version of the LCM for LCMs before you set its lifecycle state to Productive:

- You can edit the lifecycle model's name, description, and permission settings.
- You can rename the states in the lifecycle model.
- You can associate additional policies with the states in the lifecycle model. (Do not modify or delete the predefined policies that are associated with this lifecycle model.)

However, do not add states to the model and do not remove states from the model. Also, do not modify any of its state transitions.

Predefined Lifecycle Models

CentraSite installs a number of predefined lifecycle models that you can use to classify and govern assets.

These include:

Model Name	Applies to	Active upon Installation of CentraSite?	Customizable?
BPM Process Lifecycle	Process objects	No	Yes
Lifecycle Model for Lifecycles	Lifecycle Models objects	Yes	In limited ways.
Policy Lifecycle	Policy objects (Both design/change-time policies and run-time policies)	Yes	In limited ways.
Service Lifecycle	Service assets	No	Yes

CentraSite uses some of these lifecycles to govern particular types of registry objects (Policy objects, for example). These lifecycle models have system policies associated with them. You can customize these lifecycle models in only limited ways. These lifecycle models are already active when CentraSite is installed.

Other lifecycle models are treated as user-defined lifecycle models. These lifecycle models are provided to you as a convenience. You can use them instead of creating your own models from scratch. You may customize these lifecycle models as required. These lifecycle models are not active when you install CentraSite. If you want to use them, you must activate them.

Customizing Lifecycle Models

For a given object type (say Service), each organization can define and activate its own lifecycle model, so that there are several models that can control the Service type. However, per organization only one lifecycle model can be active. When a new Service instance is created, then the user creating belongs to an organization and that organization's active lifecycle model is used to control the particular Service instance.

It is possible to define a lifecycle model which consists of multiple nodes (registries) to make up one overall model. Each node only knows the model for the current node. It also knows the nodes that make up the complete lifecycle model and where they are.

When a lifecycle model is stored in CentraSite, it must be a valid state machine. This specifically means that all of the following semantic rules must be checked by CentraSite:

- There is an initial state. All objects under the control of the lifecycle model are initially in this state.

- Each state must be reachable from at least one other state.
- A default next state can be defined for each state.
- There must be at least one end state. An end state is one that it does not have any next state.
- There may be a preferred transition from one state to the next (for the UIs to use).

Lifecycle models can be applied to:

- Assets
- Policies
- Lifecycle models

CentraSite does not require you to apply lifecycle models to the assets in your registry. You can create and maintain an asset catalog without them. However, most of CentraSite's policy-based governance controls (for example, enforcing approval processes, validating attribute settings) can only be applied to assets that have an associated lifecycle model. The use of lifecycle models also makes it easier to manage the assignment of permissions (for example, granting View permissions to additional organizations) as an asset moves through its lifecycle.

If you want to use design/change-time policies to impose governance controls on a particular type of asset or you want to automate routine management tasks at certain points in the asset's lifecycle (such as setting permission assignments), associate a lifecycle model with the asset type.

Note:

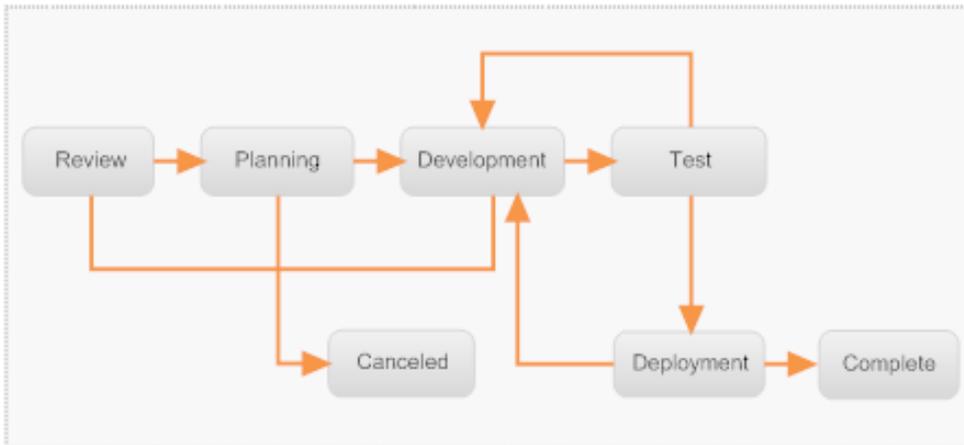
For your convenience, CentraSite provides predefined lifecycle models for several types of assets. You can use these lifecycle models as-is or customize them to suit your needs.

Applying Lifecycle Models to Asset Types

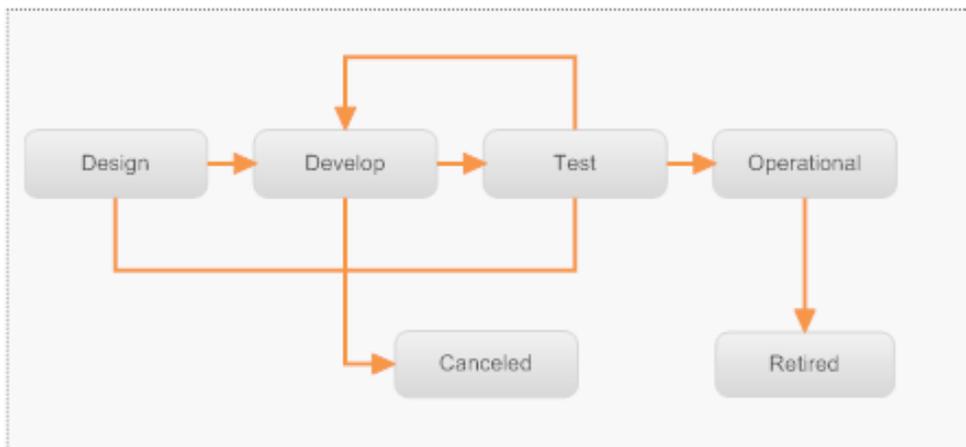
When you define a lifecycle model, you specify the asset types to which the model applies. Because different types of assets usually have different development paths, you generally create models that are specific to a single asset type (that is, one model for Service assets, one model for XML Schema assets, one model for Business Processes, and so on). However, if multiple asset types have the same lifecycle path, you can apply the same lifecycle model to them all.

You can model different lifecycle models for different assets types

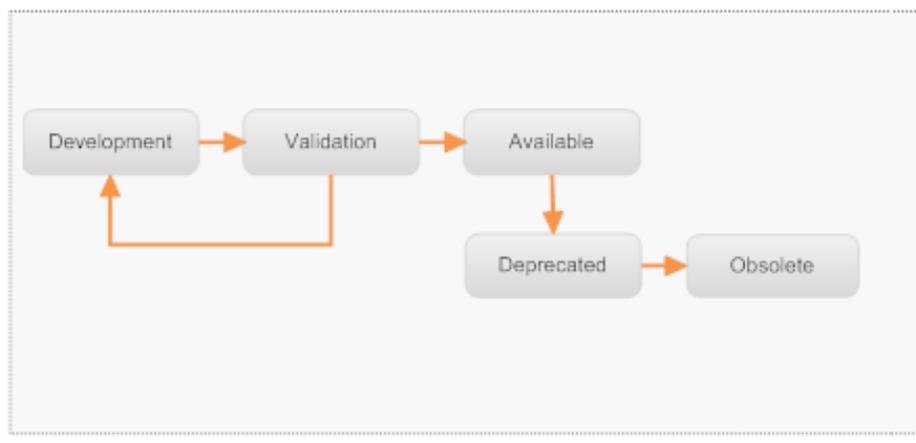
Lifecycle Model for Projects



Lifecycle Model for Business Processes



Lifecycle Model for XML Schemas and DTDs



When you apply the same lifecycle model to multiple asset types, you do not necessarily have to apply the same state-change policies to those types. You can trigger different policies depending

on the type of asset whose state is changed. If you were using the lifecycle model for XML Schema and DTDs shown in the figure above, you might create one policy that executes when an XML Schema switches to the Available state and another policy that executes when a DTD switches to the Available state.

If an active lifecycle model is already associated with a particular type within an organization, you cannot assign another lifecycle model to that type. In other words, a particular object type can be associated with one and only one lifecycle model within a particular organization. If you want to switch an object type to a different lifecycle model, you must disassociate it with the current model and then assign it to the new model.

Applying Lifecycle Models to Asset Types

You can associate a lifecycle model with virtual asset type. When you apply lifecycle model to a virtual asset type, you do not necessarily have to apply the same lifecycle model as its base type. You can create a lifecycle model specific to the virtual asset type.

You can configure a virtual type to follow the same lifecycle model as its base type or you can give the virtual type its own lifecycle model to follow. Whether a virtual type follows the lifecycle model of its base type is determined by the type's **Inherit Base Type LCM** setting.

When the **Inherit Base Type LCM** option is selected for a virtual type, the virtual type automatically inherits its lifecycle model from the base type if the virtual type does not have an assigned lifecycle model of its own. (In other words, the lifecycle model for the base type serves as the default lifecycle model for the virtual type. CentraSite applies the lifecycle model of the base type to the virtual type only when the virtual type has no other lifecycle model assigned to it.)

If the **Inherit Base Type LCM** option is disabled, the lifecycle of the virtual type is completely independent of the lifecycle of the base type. The virtual type will only have lifecycle model if you explicitly assign one to it.

The following table summarizes how lifecycle model is applied to a virtual type depending on the state of the **Inherit Base Type LCM** option.

If the Inherit Base Type LCM option is...	And the Base Type...	And the Virtual Type...	Instances of the Virtual Type...
ENABLED	HAS an assigned lifecycle model	DOES NOT HAVE an assigned lifecycle model	Follows the lifecycle model assigned to the Base Type
ENABLED	HAS an assigned lifecycle model	HAS an assigned lifecycle model	Follows the lifecycle model assigned to the Virtual Type
ENABLED	DOES NOT HAVE an assigned lifecycle model	DOES NOT HAVE an assigned lifecycle model	does not have an assigned lifecycle model
ENABLED	DOES NOT HAVE an assigned lifecycle model	HAS an assigned lifecycle model	Follows the lifecycle model assigned to the Virtual Type

If the Inherit Base Type LCM option is...	And the Base Type...	And the Virtual Type...	Instances of the Virtual Type...
DISABLED	HAS an assigned lifecycle model	DOES NOT HAVE an assigned lifecycle model	does not have an assigned lifecycle model
DISABLED	HAS an assigned lifecycle model	HAS an assigned lifecycle model	Follows the lifecycle model assigned to the Virtual Type
DISABLED	DOES NOT HAVE an assigned lifecycle model	DOES NOT HAVE an assigned lifecycle model	does not have an assigned lifecycle model
DISABLED	DOES NOT HAVE an assigned lifecycle model	HAS an assigned lifecycle model	Follows the lifecycle model assigned to the Virtual Type

Note:

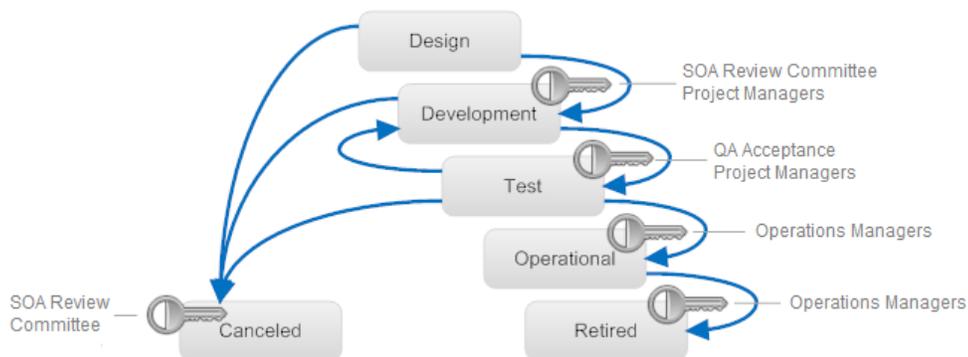
If a virtual type initially inherits its lifecycle model from the base type and later it is assigned its own lifecycle model, instances of the virtual type that were created while the type was following the lifecycle model of the base type continues following that model. Only instances of the virtual type that are created after the new lifecycle model is assigned to the virtual type complies with the virtual type's newly assigned lifecycle model.

Assigning Permissions to Lifecycle Model States

Each state that you define in a lifecycle model includes a set of optional *state permissions*. State permissions enable you to restrict who can transition assets to a specified state. You can assign state permissions to individual users or to groups.

If you do not explicitly assign permissions to a state, any user with Modify permission on an object can switch the object to that state.

You can optionally assign permissions to the states in a lifecycle



When you assign permissions to a state, two sets of users are allowed to switch an asset to that state: the set of users to which you explicitly grant state permission and users who have implicit

permission to switch lifecycle states. The set of users who have implicit permission to switch lifecycle states are:

- Users with Manage System-Wide Lifecycle Models permission (on objects managed by a system-wide lifecycle model).
- Users with Manage Lifecycle Models permission (on objects managed by an organization-specific lifecycle model).
- The owner of the Lifecycle Model.

Note that the group of users with implicit permission to switch states does not include the owner of the asset itself. If you want to give asset owners the ability to switch their assets to a particular state, you must explicitly include them using the state permission settings.

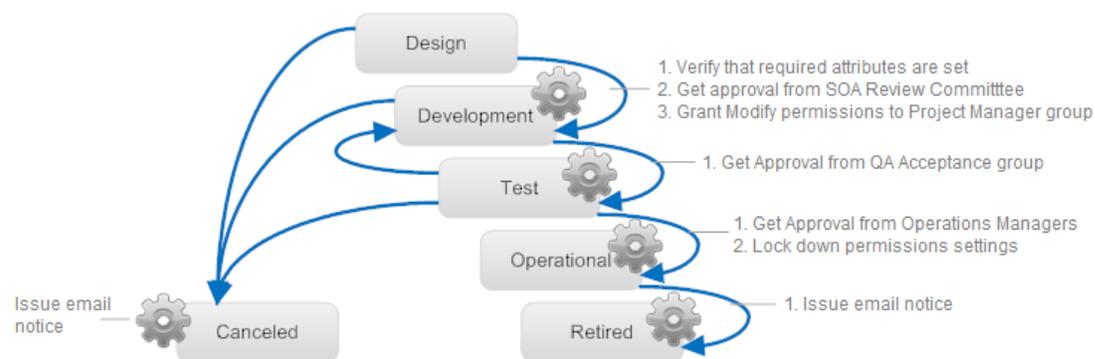
Also note that granting state permission to a user does not, in itself, give the user the ability to switch an asset to that state. The user must also have Modify permission on the asset itself. For example, let's say you give the Users group for organization ABC permission to switch assets to the Development state. Doing this does not mean that any user in organization ABC can switch the assets in organization ABC to the Development state. It means that any user in organization ABC *with Modify permission on an asset* can switch that asset to the Development state.

Note: CentraSite does not allow you to modify a lifecycle model, including its state permissions, after you activate the model. If you assign state permissions to a lifecycle model, consider assigning the permissions to groups instead of individual users. Doing this enables you to make simple adjustments to the permission settings by simply modifying the membership of the assigned groups. You does not have to deactivate the model to make these kinds of changes.

Triggering Policies during Lifecycle Model Transitions

You can configure design/change-time policies to execute during the transition points in an asset's lifecycle. For example, you might apply a policy that gives View permission to a specified group of users when an asset enters the Operational state, or you might apply a policy that obtains approvals from a review group before an asset enters the Development state.

CentraSite triggers polices when specified state transitions occur



When you create a policy that executes on a state change event, you specify whether the policy should execute immediately before CentraSite actually modifies the asset's state (which is called

a Pre-State Change event) or immediately after CentraSite modifies the asset's state (a Post-State Change event).

Generally, you execute policies that perform approval and validation actions on the Pre-State Change event. In other words, you use Pre-State Change policies to ensure that an asset satisfies the entry criteria for a given state.

On a Post-State Change event, you typically execute policies that update the asset (for example, granting instance-level permissions to the users who need to work with the asset in the next phase of its lifecycle) or issue notifications (for example, sending an email). In short, you use Post-State Change policies to execute actions that are to be carried out only if the asset's state is switched successfully.

There are, exceptions to the generalizations above. Under some circumstances you might want to set an asset's instance-level permissions or update its attributes in a Pre-State Change policy. That is you want to perform approval and validation actions in a Pre-State Change policy, and issue notifications and perform state-certain actions (that is, actions that should occur only after an object's state has been successfully switched) in a Post-State Change policy.

Updating Assets that are Under Lifecycle Management

If you need to update an asset that has reached the production phase of its lifecycle, you have a couple of choices. If you need to make a minor change, for example, you need to correct an attribute setting, add a classifier to the asset or modify the asset's description, an authorized user can simply make the change directly to the production version of the asset. If you are working in a multi-stage environment, you will need to manually apply the updates to the asset in each of the participating registries.

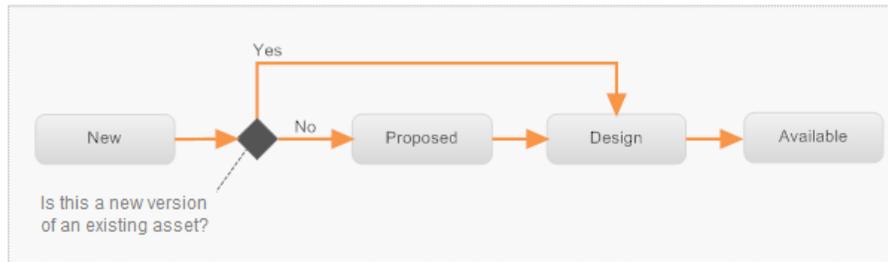
If the changes are substantive, in particular, if they involve changes to the structure of a schema or the definition of an interface, then you should create a new version of the asset. When you create a new version of an asset, the new version enters the initial lifecycle state, just as though it were a completely new asset. The new version of the asset will pass through the entire lifecycle just like any other new asset of its type.

If you are working in a multi-stage environment, you must create the new version on the registry that hosts the first stage of the lifecycle (that is, on the creation CentraSite). When the new version reaches the end state on that stage, you would promote the new version of the asset to the next stage just as you did with the previous version of the asset.

Creating a Different Lifecycle Path for a New Version of an Asset

For certain asset types, you might want to define separate lifecycle paths for new instances of an asset and new versions of an asset. For example, in a lifecycle for an XML schema like the one shown below, you might want new versions of existing schemas to bypass the Proposed state and go directly to the Design state.

Alternate Lifecycle Path for a New Version of an Asset



Creating an alternate path in a lifecycle requires the use of policies that conditionally change the state of an asset depending on the way in which the asset is classified. In the example shown above, this is achieved by doing the following:

- Defining an initial state (the New state) through which all schemas (new or versioned) pass.
- Creating policies that execute immediately after a schema enters the New state. These policies switch the schema to the Proposed state or the Design state depending on whether the schema is classified as New or Existing.

To implement a lifecycle like the one above, you must add to the XML Schema asset type a Classification attribute that can be used to classify a schema as either New or Existing. (You would need to create a custom taxonomy to support this attribute.)

You must also create two policies that execute after a schema enters the New state: one policy that executes when a New schema enters the New state (this policy will switch the schema to the Proposed state), and one policy that executes when an Existing schema enters the New state (this policy switches the schema to the Design state).

Note:

The example above describes how you can use policies to conditionally route an asset between two alternate paths when an asset enters the initial state of its lifecycle. However, you can use this same technique to establish alternate paths at any point in the asset's lifecycle. Its use is not limited to the initial state.

Reverting an Asset That is Under Lifecycle Management to a Previous State

When you switch an asset from one state to another, the asset exists in a pending state until the requested state change is complete. While an asset is in the pending state, it cannot be modified.

For most state switches, this is a very brief period of time. However, if the state change involves the execution of policies, it can be quite long (in the case of an approval policy, an asset might remain in the pending state for days).

An object remains in the pending state until the requested state change is complete



Name: OFX_Banking.xsd
Description: XML Schema file
Version: 1.0
Created: 2009-11-01 03:34 PM
Last Modified: 2009-11-01 03:35 PM
Lifecycle State: PENDING (Design)
Organization: MIS
Owner: MIS, CSA MIS

An asset can, on occasion, encounter conditions that cause it to become stuck in the pending state. To resolve the situation where an asset becomes stuck in the pending state, a user that has the CentraSite Administrator role can use the **Revert Pending State** command to return the asset to its prior state. After the asset is reverted and the issue that caused the asset to become stuck is corrected, an authorized user can switch the asset to its next lifecycle state again.

Note:

Reverting the lifecycle state of an asset does not undo any attribute changes that might have been made by policies that executed during the first state change event. It simply returns the asset's lifecycle property to its previous state. If other attribute changes occurred during the state change event, you need to undo those changes manually.

Managing Lifecycle Models through CentraSite Business UI

This section describes operations you can perform to manage lifecycle models through CentraSite Business UI.

Adding Lifecycle Model to Organization

Pre-requisites:

To create an organization-specific lifecycle model, you must have the *Manage Lifecycle Models* permission for the organization.

In addition, you can create a *system-wide* lifecycle model (that is, a lifecycle model that applies to all organizations), if you have the *Manage System-wide Lifecycle Models* permission.

The definition of a lifecycle model consists of specifying basic attributes of the model, the model's states and transitions, user permissions, and the associated asset types and policies.

A lifecycle model is valid if:

- it consists of at least one state and
- there is a possible transition path to all of the states except the initial state.
- there is at least one state, designated as the final state, that does not have a transition path to any other state.

You can create a new lifecycle model using the **Add Lifecycle Model** action.

➤ **To add a lifecycle model to an organization**

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. Click **Add Lifecycle Model**.

The Add Lifecycle Model page appears.

3. In the area labeled **Basic Information**, specify basic information for the lifecycle model.

Field	Description
Name	<p>Name of the lifecycle model.</p> <p>This is the name that users will see when they search for lifecycle models in CentraSite.</p> <p>A lifecycle model name can contain any characters (including spaces).</p>
Description	<p><i>Optional.</i> The description for the lifecycle model.</p> <p>This description appears when a user displays the list of lifecycle models in the Search Results page.</p>
Version	<p>A user-defined version number for the lifecycle model.</p>
Organization	<p>The organization to which you want to add the lifecycle model. (The Organization list only displays organizations for which you have the Manage Lifecycle Models permission.) If you select one of these names, the lifecycle model applies just to objects belonging to the selected organization.</p> <p>You can select All option from the drop-down list, implying that the lifecycle model is a system-wide lifecycle model, if you have the Manage System-wide Lifecycle Models permission.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Important: Select the organization carefully. If you choose the All option, you cannot change the organization assignment later.</p> </div>
Applicable Types	<p>The asset types associated with the lifecycle model.</p> <p>When a lifecycle model is assigned to an asset type and the lifecycle model is activated (that is, set to the state Productive), all existing assets of this type in the user's organization are assigned to the initial state of the lifecycle model. Also, each subsequently created asset of the given asset type in the user's organization is automatically associated with the lifecycle model and is initially set to the initial state of the lifecycle model. For more information about associating policies with lifecycle models, see “Associating Asset Types with Lifecycle Model” on page 327.</p> <p>If the selected asset type has virtual variants, then its virtual variants are automatically selected for the lifecycle model. Click here for a list of applicable types.</p>

4. In the canvas area, use the **Add Intermediate State** and **Add Final State** buttons to define states for the lifecycle model.

When you try to add a new lifecycle model to CentraSite, an **Initial State** is added to the lifecycle model by default, because it is mandatory for a lifecycle model to have at least one state. Also, if there is a state that does not have a transition path in the lifecycle model, CentraSite converts that state as the **Final State** of the model.

5. Click on a lifecycle state to display its detailed information. In the area labeled **Lifecycle State Properties**, modify the name and description of the lifecycle state, and its target state transition in the lifecycle model.

Field	Description
State Name	<p>Name of the state.</p> <p>This is the name that users will see when they define the state transitions for a lifecycle model.</p> <p>A state name can contain any characters (including spaces), and must be unique within a lifecycle model.</p>
Description	<p><i>Optional.</i> A comment that describes the purpose of the state.</p> <p>This description appears when a user hovers over its name in the state transition area.</p>
Target States	<p>A state that can be reached as a result of a transition from the current state.</p> <p>In the Target States list box, you can select the target state from a list of the states that you have defined so far. Alternatively, in the canvas area, you can select a single state, and then drag across the other states you want a transition from the selected state.</p> <div data-bbox="370 1291 1365 1465" style="background-color: #f0f0f0; padding: 10px;"><p>Note: The Target States list box is not displayed for a final state. This is because, the target state cannot have a transition to any other state within in the lifecycle model.</p></div> <p>If you wish to define more than one target state from the current state, click the plus icon next to the target state name. You may find it convenient to define all of the states before you start to define the state transitions.</p> <p>If you have more than one target state, select one of the target states as the default target state by choosing the radio button in the Default Target State.</p>
Promotion Stages	<p>A stage that can be the target of a transition from an end state of the current stage. An end state is a state that has no transitions to other states.</p> <p>The term promotion is used to describe the transition of an asset from one stage to the next stage.</p>

Field	Description
	If you wish to define more than one target stage from the selected state, click the plus icon.
Applied Policies	A list of the currently defined policies for the state. These policies will be triggered when the particular state in the lifecycle is entered.

- Click **Save**.

CentraSite validates the newly created lifecycle model.

If there are no validation errors, the lifecycle model is added to the CentraSite Registry Repository. If any part of the model is invalid, this is indicated by an appropriate error message.

Adding Lifecycle Stage

Pre-requisites:

To create a lifecycle stage, you must have the *Manage System-wide Lifecycle Models* permission.

A lifecycle model can contain one or more stages. Typically, stages are used to represent a clearly-defined deployment scenario within an object's development cycle. You could, for example, decide to use the lifecycle model during various stages of a product lifecycle such as development, test and production. The individual stages are also usually deployed on different physical machines, which is normal practice if for example you have a test environment and a production environment. Each stage you define contains all of the states of the model.

You can create a new lifecycle stage using the **Add Lifecycle Stage** action.

> To add a lifecycle stage

- In the CentraSite Business UI activity bar, click **Design Time**.
- Click **Add Lifecycle Stage**.
- In the **Create Lifecycle Stage** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Name	Name of the lifecycle stage.
Description	(Optional). The description for the lifecycle stage.
Endpoint	The URL of the host machine where the lifecycle stage is deployed on a CentraSite installation.
	The Endpoint URL has the following format:

Field	Description
	<code><scheme>://<host>:<port></code>
	The scheme is <code>http</code> or <code>https</code> . The host is the machine on which CAST is running, and port is the port on which CentraSite is listening.

4. Click **OK**.

The newly created lifecycle stage is added to the CentraSite Registry Repository.

Adding State to Lifecycle Model

Pre-requisites:

To define states for an organization-specific lifecycle model, you must have the *Manage Lifecycle Models* permission for the organization.

By default, users in the CentraSite Administrator or Organization Administrator role have this permission, although an administrator can grant this permission to other roles.

You define states for a lifecycle model using the Lifecycle Model details page.

➤ To add a new state to an existing lifecycle model

1. In the CentraSite Business UI activity bar, click **Design Time**.

A list of all lifecycle models appears.

2. Click the lifecycle model you want to add the state.

This opens the Lifecycle Model details page.

3. In the canvas area, click the **Add Intermediate State** and **Add Final State** buttons as required.

The State number is incremented by one each time you click the **Add Intermediate State** button (for example, State 1, State 2, State 3).

4. Click on a lifecycle state to display its detailed information. In the area labeled **Lifecycle State Properties**, modify the name and description of the lifecycle state, and its target state transition in the lifecycle model.
 - a. In the field labeled **State Name**, type the name of the state.
 - b. In the field labeled **Description**, type the description for the state.
 - c. In the field labeled **Target States**, select a state that can be reached as a result of a transition from the current state.

In the **Target States** list, you can select the target state from a list of the states that you have defined so far.

Note:

Alternatively, in the canvas area, you can select a single state, and then drag the mouse pointer across the other states you want a transition from the selected state.

If you wish to define more than one target state from the current state, click the plus icon next to the target state name.

If you have more than one target state, select one of the target states as the default target state by choosing the radio button in the **Default Target State**.

Note:

You may find it convenient to define all of the states before you start to define the state transitions.

- d. In the field labeled **Promotion Stages**, select a stage that can be the target of a transition from an end state of the current stage.

If you wish to define more than one target stage from the selected state, click the plus icon.

- e. In the field labeled **Applied Policies**, view a list of the currently defined policies for the state.

5. Click **Save**.

CentraSite validates the updated lifecycle model.

If there are no validation errors, the lifecycle model is updated in the CentraSite Registry Repository. If any part of the model is invalid, this is indicated by an appropriate error message.

Viewing the Lifecycle Model List and Details

Pre-requisites:

To view a list of all organization-specific lifecycle models, you must have the *Manage Lifecycle Models* permission for the organization. All users who log on to CentraSite Business UI can view the Lifecycle Model in the *read-only* mode.

By default, users in the CentraSite Administrator or Organization Administrator role have this permission, although an administrator can grant this permission to other roles.

You can view the list of available lifecycle models using the Search Results page. In addition, you can view the lifecycle model details, modify details, define states, state transitions, stages, and policies associated with a lifecycle model in the Lifecycle Model details page.

➤ To view the lifecycle model list and details

1. In the CentraSite Business UI activity bar, click **Design Time**.

A list of all lifecycle models appears.

2. To filter the list of available lifecycle models, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Applicable Scopes** list, select **Lifecycle Model**.
 - c. If you want to filter the list to see just a subset of the available lifecycle models, do the following:

To see...**Do this...**

A subset of the available lifecycle models

Type a partial text string in the **Keyword** text field. Click the plus button next to **Keyword** or press *Enter* to add the keyword to the search recipe.

The keyword search returns a list of lifecycle models that contain the specified keyword (that is, text string) in the model's string attributes (name and description).

A list of lifecycle models whose scope applies for a particular organization

- a. In the **Applicable Organizations** list, select an organization.
- b. Click the plus button next to the **Applicable Organizations** list or press *Enter* to add the selected organization to the search recipe.

A list of lifecycle models whose scope applies for all organizations

- a. In the **Applicable Organizations** list, select **All**.
- b. Click the plus button next to the **Applicable Organizations** list or press *Enter* to add all the organizations to the search recipe.

The Search Results page provides the following information about each model:

Column	Description
Name	Name of the lifecycle model.
Description	Descriptive information of the lifecycle model.
Organization	Name of the organization to which the lifecycle model belongs.
Owner	The user to which the lifecycle model belongs.
Version	The user-defined version number for the lifecycle model.
Created Date	The date on which the lifecycle model was added to the registry. CentraSite automatically sets this attribute when a user adds the lifecycle model to the registry. Once it is set, it cannot be modified.

Column	Description						
Last Updated	The date on which the lifecycle model was last modified.						
Lifecycle State	The current state of the lifecycle model (New, Productive, Retired).						
Asset Types	The object types to which the model applies for lifecycle path.						
Activation State (toggle button)	The status of the lifecycle model.						
	<table border="1"> <thead> <tr> <th>Icon</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>The lifecycle model is active (can be applied to asset types).</td> </tr> <tr> <td></td> <td>The lifecycle model is inactive, that is in the New or Retired state (cannot be applied to asset types).</td> </tr> </tbody> </table>	Icon	Description		The lifecycle model is active (can be applied to asset types).		The lifecycle model is inactive, that is in the New or Retired state (cannot be applied to asset types).
Icon	Description						
	The lifecycle model is active (can be applied to asset types).						
	The lifecycle model is inactive, that is in the New or Retired state (cannot be applied to asset types).						

The action bar displays one or more icons that you can use to perform various tasks on the lifecycle models.

Action	Description
Add Lifecycle Model	Creates a new lifecycle model in CentraSite.
Delete	Deletes the most recent version of a lifecycle model.

3. Select the lifecycle model to display its details page.

The lifecycle model details are displayed in the following sections:

- The area labeled **Basic Information** displays the basic lifecycle model information - Name, Description, Version, Organization, and Applicable Types. The **Applicable Types** field displays a list of asset types that are associated with the lifecycle model.
- The canvas area displays a list of states, state transitions, stages, and policies that are defined for the lifecycle model.
- The area labeled **Lifecycle State Properties** displays the lifecycle state information - State Name, Description, Target States, and Promotion Stages.

Viewing the Lifecycle Stage List

Pre-requisites:

To view a list of all lifecycle stages, you must have the *Manage System-wide Lifecycle Models* permission.

You can view the list of available lifecycle stages using the Search Results page. In addition, you can view and modify the lifecycle stage details in the Edit Lifecycle Stage dialog box.

➤ **To view the lifecycle stage list and details**

1. In the CentraSite Business UI activity bar, click **Design Time**.

A list of all lifecycle stages appears.

2. To filter the list of available lifecycle stages, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Applicable Scopes** list, select **Lifecycle Stage**.
 - c. To filter the list to see just a subset of the available lifecycle stages, type a partial string in the **Keyword** text field. Click the plus button next to **Keyword** or press Enter to add the keyword to the search recipe.

The Search Results page provides the following information about each stage:

Column	Description
Name	Name of the lifecycle stage.
Description	Descriptive information of the lifecycle stage.
Organization	Name of the organization to which the lifecycle stage belongs.
Owner	The user to which the lifecycle stage belongs.
Version	The system-defined version number for the lifecycle stage.
Created Date	The date on which the lifecycle stage was added to the registry. CentraSite automatically sets this attribute when a user adds the lifecycle stage to the registry. Once it is set, it cannot be modified.
Last Updated	The date on which the lifecycle stage was last modified.

The action bar displays one or more icons that you can use to perform various tasks on the lifecycle stages.

Action	Description
Add Lifecycle Stage	Creates a new lifecycle stage in CentraSite.
Delete	Deletes a lifecycle stage from CentraSite.

3. Select the lifecycle stage to display its details page.

The **Edit Lifecycle Stage** dialog box displays the lifecycle stage information - Name, Description, and Endpoint.

Modifying Lifecycle Model Details

Pre-requisites:

To modify an organization-specific lifecycle model information, you must have the *Manage Lifecycle Models* permission for the organization.

By default, users in the CentraSite Administrator or Organization Administrator role have this permission, although an administrator can grant this permission to other roles.

You can modify an existing lifecycle model information using the Lifecycle Model details page.

You cannot modify a lifecycle model that is currently in the Productive state. To modify a lifecycle model in the Productive state, you must create a new version of the lifecycle model.

➤ To modify the details of a lifecycle model

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. Select the lifecycle model that you want to update.

This opens the Lifecycle Model details page. On the Lifecycle Model details page, examine and modify the values of the attributes as required.

3. In the area labeled **Basic Information**, modify the values of the attributes in the respective data fields as required.
4. To modify the associated asset types, in the **Applicable Types** list box, assign one or multiple asset types to the lifecycle model, and remove one or multiple asset types from the lifecycle model, as required.
5. To modify the states and state transitions, stage transitions, and the associated policies of the lifecycle model, in the canvas area, modify the values of the attributes in the respective data fields as required.
6. To modify the permissions, click the **Permission** icon. Assign one or multiple users or groups with the state permissions, as required.
7. Click **Save**.

CentraSite validates the updated lifecycle model.

If there are no validation errors, the lifecycle model is updated in the CentraSite Registry Repository. If any part of the model is invalid, this is indicated by an appropriate error message.

Modifying Lifecycle Stage Details

Pre-requisites:

To modify a lifecycle stage information, you must have the *Manage System-wide Lifecycle Models* permission.

You can modify an existing lifecycle stage information using the **Edit Lifecycle Stage** dialog box.

➤ **To modify the details of a lifecycle stage**

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. Select the lifecycle stage that you want to update.
3. In the **Edit Lifecycle Stage** dialog box, modify the values of the attributes in the respective data fields as required.
4. Click **Save**.

Defining Stage Transitions for Lifecycle Model

Pre-requisites:

To define stage transition for an organization-specific lifecycle model, you must have the *Manage Lifecycle Models* permission for the organization.

By default, users in the CentraSite Administrator or Organization Administrator role have this permission, although an administrator can grant this permission to other roles.

You define the stage transition from an end state of the current stage to one or more new stages. An end state is a state that has no transitions to other states.

The term *promotion* is used to describe the transition of an asset from one stage to the next stage.

You can define stage transitions for a lifecycle model using the Lifecycle Model details page.

➤ **To define stage transitions for the lifecycle model**

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. Select the lifecycle model for which you want to define the state transition.

This opens the Lifecycle Model details page.
3. In the canvas area, click on a final state.

The stage details are displayed on the right side.
4. In the **Promotion Stages** list box, select one or more stages that can be the target of a transition from the selected state.

5. Click **OK**.
6. Click **Save**.

CentraSite validates the updated lifecycle model.

If there are no validation errors, the lifecycle model is updated in the CentraSite Registry Repository. If any part of the model is invalid, this is indicated by an appropriate error message.

Associating Asset Types with Lifecycle Model

Pre-requisites:

To associate asset types to an organization-specific lifecycle model, you must have the *Manage Lifecycle Models* permission for the organization.

By default, users in the CentraSite Administrator or Organization Administrator role have this permission, although an administrator can grant this permission to other roles.

When a lifecycle model is assigned to an asset type and the lifecycle model is activated (that is, set to the state Productive), all existing assets of this type in the user's organization are assigned to the first state of the lifecycle model. Also, each subsequently created asset of the given asset type in the user's organization is automatically associated with the lifecycle model and is initially set to the first state of the lifecycle model.

An asset type can only be associated with one lifecycle model within an organization. If a lifecycle model includes an asset type that is already associated with another lifecycle model in the same organization, an error message is displayed when you try to activate the lifecycle model.

If you need to add another asset type to a lifecycle model that is already in the Productive state, you must create a new version of the lifecycle model and add the asset type to the new version.

You can associate asset types to a lifecycle model using the Lifecycle Model details page.

➤ To associate a lifecycle model to an asset

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. Select the lifecycle model for which you want to associate asset types.

This opens the Lifecycle Model details page.
3. In the area labeled **Basic Information**, locate the **Applicable Types** list.
4. Click the chevron next to the **Applicable Types** list.

A list of available asset types that can be associated with the lifecycle model is displayed.

5. To define that a particular asset type must be associated with the lifecycle model, select the asset type from the **Applicable Types** list.

Note:

If you apply a lifecycle model to the **Service** asset type that has Virtual types associated with it, the model automatically applies to the asset instances of the Virtual types when both of the following conditions are met:

- Virtual type's **Inherit Base Type LCM** option is selected.
- Virtual type does not have a lifecycle model assigned to it.

Viewing Policies Associated with Lifecycle Model

Pre-requisites:

To view the policies that are associated with an organization-specific lifecycle model, you must have the *Manage Lifecycle Models* permission for the organization.

By default, users in the CentraSite Administrator or Organization Administrator role have this permission, although an administrator can grant this permission to other roles.

You can view the policies associated with a lifecycle model using the Lifecycle Model details page.

You can define one or multiple policies to be triggered when a state in the lifecycle is entered.

» To view policies associated with the lifecycle model

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. Select the lifecycle model for which you want to view the associated policies.

This opens the Lifecycle Model details page.

3. In the canvas area, click on the required state.

The stage details are displayed on the right side.

4. Click the **show applied policies for this state** link at the bottom of the details page.

A list of the defined policies for the state, and whether or not the policies are currently active or inactive is displayed.

A policy can be assigned to a particular state when you create a policy.

Setting Permissions on Lifecycle State

Pre-requisites:

To set state permissions for an organization-specific lifecycle model, you must have the *Manage Lifecycle Models* permission for the organization or at least have the instance-level *Full* permission for the lifecycle state.

By default, users in the CentraSite Administrator or Organization Administrator role have this permission, although an administrator can grant this permission to other roles.

You define the user and group permissions associated with each state of the lifecycle model. Each permission defines whether a user or group can move assets into a particular lifecycle state. Such permissions are referred to as state permissions.

If you leave the list of users and groups empty, CentraSite grants state permissions to all users and groups. If the list contains at least one user or group, permissions are denied for all other users and groups who are not in the list.

You can set state permissions for a lifecycle model using the Lifecycle Model details page.

➤ To assign permissions on a lifecycle state

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. Select the lifecycle model for which you want to assign the user and group permissions.
3. In the Lifecycle Model details page, click **Permissions**.

This opens the **Assign Lifecycle State Permissions** dialog box to assign the user and group permissions.

4. To add users or groups to the **User and Group Permissions** list, do one of the following:
 - a. To filter the list of users and groups, type a partial string in the **Add User or Group** text box.

CentraSite applies the filter to the **Name** column.

Examples

String	Description
b	Displays names that contain b.
bar	Displays names that contain bar.

- b. Select the user or group you want to grant or deny permission to change the lifecycle state of an asset.
- c. Click the plus button next to the **Add User or Group** text box to add the user or group to the **User and Group Permissions** list.
- d. Select one or more states into which the user or group is allowed to move an asset. You can either select the check boxes for the states individually, or select them all by selecting the corresponding check box in the **ALL** column. To cancel the selection of a state, clear the corresponding check box.

Note that there is no column corresponding to the initial state of an asset, because all assets under control of this lifecycle model are automatically put into the initial state.

- e. Click **OK**.

-OR-

- a. Click **Choose**. This opens the **Choose Users and Groups** dialog box.
- b. To filter the list of users and groups, type a partial string in the search text box.
- c. Click the **Search** icon.

CentraSite applies the filter to the **Name** column.

- d. Select the user or group you want to grant or deny permission to change the lifecycle state of an asset.
- e. Click **OK** to add the selected user or group.
- f. Select one or more states into which the user or group is allowed to move an asset. You can either select the check boxes for the states individually, or select them all by selecting the corresponding check box in the **ALL** column. To cancel the selection of a state, clear the corresponding check box.

Note that there is no column corresponding to the initial state of an asset, because all assets under control of this lifecycle model are automatically put into the initial state.

- g. Click **OK**.
5. Click **Save**.

CentraSite validates the updated lifecycle model.

If there are no validation errors, the lifecycle model is updated in the CentraSite registry or repository. If any part of the model is invalid, this is indicated by an appropriate error message.

Activating Lifecycle Model

Pre-requisites:

To activate an organization-specific lifecycle model information, you must have the *Manage Lifecycle Models* permission for the organization.

To activate a system-wide lifecycle model, you must have the *Modify Assets* permission for all organizations that own the lifecycle model's assigned asset types. This is because when a system-wide lifecycle model is activated, assets of its assigned asset types from all organizations is set to the initial state of the lifecycle model and this requires the modify permission for the assets.

Lifecycle models can be applied not only to assets but also to other lifecycle models. CentraSite provides a default lifecycle model that is applied automatically to all user-defined lifecycle models. This default lifecycle model defines the following states:

State	Description
New	A user-defined lifecycle model has been saved but is not yet been activated for use with its associated asset types.

State	Description
Productive	A user-defined lifecycle model has been activated for use with its associated asset types. The lifecycle status of assets of the associated asset types is visible in the detail view of the asset instances.
Retired	The lifecycle model is no longer in use and cannot be reactivated.

The following general guidelines apply when activating a lifecycle model:

- When you create a user-defined lifecycle model and assign it to one or more asset types, the lifecycle model is initially inactive (that is, in the New state).. CentraSite does not begin enforcing a new lifecycle model until you *activate* the model by switching it to the Productive state.
- After you activate a lifecycle model (that is, place it in the Productive state), that lifecycle model can no longer be modified. To make changes to the lifecycle model, you must create a new version of the model and make your changes to the new version.
- You cannot activate a lifecycle model that is currently in the Retired state. To activate a lifecycle model in the Retired state, you must create a new version of the lifecycle model.

➤ To activate a lifecycle model

1. In the CentraSite Business UI activity bar, click **Design Time**.
 2. Locate the lifecycle model you want to activate.
 3. Click the activation toggle button.
- A confirmation dialog box appears.
4. Click **Yes** to confirm activation.

CentraSite sets the state of the lifecycle model to **Productive**.

Versioning Lifecycle Model

Pre-requisites:

To version an organization-specific lifecycle model information, you must have the *Manage Lifecycle Models* permission for the organization.

If you need to make changes to a lifecycle model after the model has been activated (that is, after you place it in the Productive state), you must either create a new version of the existing model or replace the existing model with a completely new model. You cannot modify a lifecycle model directly after it has been activated.

The easiest way to apply changes to a lifecycle model is to generate a new version of the model. This task involves the following basic steps:

1. Creating a new version of the model. During this step, CentraSite creates an exact copy of the existing lifecycle model.
2. Updating the new version of the lifecycle model as necessary (for example, adjusting its state permissions, inserting additional states, removing states, modifying transition paths, applying it to additional object types).
3. Activating the new version. This step automatically activates the new version and retires the old version.

When you activate a new version of a lifecycle, instances of assets that were created under the old lifecycle model will automatically switch to the new lifecycle model if they are in a state that exists in the new model. Otherwise, they will continue to follow the old lifecycle model until they are switched to a state that exists in both models. At that point, they will switch to the new lifecycle model.

You can also apply changes to a lifecycle by defining an entirely new lifecycle model. To put the new model into effect, you must retire the existing model and then activate the new model. When you change a lifecycle this way, the objects that were created using the old model will complete their lifecycles under the old model. Objects that are created after the new model is activated will follow the new model.

You can create a new version of a lifecycle model using the Lifecycle Model details page.

The following general guidelines apply when versioning a lifecycle model:

- You can only create a new version from the latest version of a lifecycle model. For example, if a lifecycle model already has versions 1.0, 2.0, and 3.0, CentraSite allows you to create a new version of the lifecycle model from version 3.0. It makes no difference whether the lifecycle model that you are versioning is active or inactive. You can version a lifecycle model in either mode.
- A lifecycle model can only be updated when it is in the New state.
- When a lifecycle model is in the Productive or Retired state, you cannot change the state of the current version of the model back to New; in this case, you can only reach the New state by creating a new version of the lifecycle model.
- CentraSite automatically establishes a relationship between the new version of the lifecycle model and the previous version. This relationship enables several capabilities and features in CentraSite that relate to versioned lifecycle models.

➤ To version a lifecycle model

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. Select the lifecycle model for which you want to create a new version.
3. In the Lifecycle Model details page, click **New Version**.

When you create a new version, it is not yet activated. The lifecycle state of the new version is **New**, and the lifecycle version of the previously used version is still **Productive**.

The new version of the lifecycle model is displayed in the Search Results page.

4. Click the new version of lifecycle model
5. In the Lifecycle Model details page, change values of the attributes in the respective data fields as required.
6. Click **Activate**.

CentraSite sets the state of the new version of the lifecycle model to **Productive**.

This automatically changes the state of the previously used version from **Productive** to **Retired**.

All existing assets of the asset types that use this lifecycle model are automatically set for use with the new version.

When you change a lifecycle model that is already in use, you must ensure that all of the states that were in use in the old model are also available in the new model. If the old state model contains states that no asset instance is currently using, these states do not need to be present in the new model. The state transitions in the new model do not depend on the state transitions in the old model; you can define the state transitions in the new model as you please.

Deleting State and State Transitions from Lifecycle Model

Pre-requisites:

To delete an organization-specific lifecycle state and state transition, you must have the *Manage Lifecycle Models* permission for the organization.

By default, users in the CentraSite Administrator or Organization Administrator role have this permission, although an administrator can grant this permission to other roles.

A state can only be deleted if there is currently no policy that is triggered when this state is entered.

➤ To delete a state and state transition from a lifecycle model

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. Select the lifecycle model whose state or state transition you want to delete.
This opens the Lifecycle Model details page.
3. In the canvas area, hover over the required state transition, and click the **X** icon.
4. In the canvas area, select the required state, and click the **x** icon.
5. Click **Yes** in the confirmation dialog.

6. Click **Save**.

The selected state is deleted from the lifecycle model. When a state is deleted, all the transitions to and from the deleted state are also deleted. You will not be able to delete the Initial state of an LCM.

Deleting Lifecycle Models

Pre-requisites:

To delete an organization-specific lifecycle model, you must have the *Manage Lifecycle Models* permission for the organization.

By default, users in the CentraSite Administrator or Organization Administrator role have this permission, although an administrator can grant this permission to other roles.

The following general guidelines apply when deleting lifecycle models in CentraSite:

- To delete a lifecycle model, you must first deactivate the lifecycle model. This means that you should change the lifecycle model's own lifecycle status from Productive to Retired.
- You can only delete the newest version of a lifecycle model. You cannot delete an older version of the lifecycle model.

Note:

The lifecycle model for lifecycles and the lifecycle model for policies cannot be deleted.

Important:

If you have selected several lifecycle models where one or more of them are predefined models, you can use **Delete** to delete them. However, as you are not allowed to delete predefined lifecycle models, only models you have permission for is deleted. The same applies to any other lifecycle models for which you do not have the required permission.

> To delete a lifecycle model

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. Select the lifecycle model that is currently in Productive state.
3. Click the deactivation toggle button.
4. Click **Yes** in the confirmation dialog.

CentraSite sets the state of the lifecycle model to **Retired**.

5. Select the required version of lifecycle model you want to delete, and click **Delete**.

You can also delete a lifecycle model from its details page.

6. Click **Yes** in the confirmation dialog.

The selected version of the lifecycle model is deleted. All older versions of the lifecycle model does not be affected.

Deleting Lifecycle Stages

Pre-requisites:

To delete a lifecycle stage, you must have the *Manage Lifecycle Models* permission for the organization.

By default, users in the CentraSite Administrator or Organization Administrator role have this permission, although an administrator can grant this permission to other roles.

A state can only be deleted if there is currently no policy that is triggered when this state is entered.

➤ To delete a state from a lifecycle model

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. Select the lifecycle stage you want to delete, and click **Delete**.

A confirmation dialog box appears.

3. Click **Yes** to confirm deletion.

The selected lifecycle stage is deleted.

Managing Lifecycle Models through CentraSite Control

This section describes operations you can perform to manage lifecycle models through CentraSite Control.

Adding Lifecycle Model to Organization

You can access this page only if you have the Manage Lifecycle Models permission for a particular organization in CentraSite.

In addition, you can create a *system-wide* lifecycle model (that is, a lifecycle model that applies to all organizations), if you have the Manage System-wide Lifecycle Models permission.

The definition of a lifecycle model consists of specifying basic attributes of the model, the model's states and transitions, user permissions, and the associated asset types and policies.

A lifecycle model is valid if:

- it consists of at least one state and

- there is a possible transition path to all of the states except the first state.

To create a new lifecycle model in CentraSite, follow these steps:

➤ **To add a lifecycle model to an organization**

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.
2. In the Lifecycle Models page, click **Add Model**.

This opens the Add Lifecycle Model page.

3. In the area labeled **Lifecycle Model Information**, provide the required information for each of the displayed data fields.

Field	Description
Name	<p>Name of the lifecycle model.</p> <p>This is the name that users will see when they search for lifecycle models in CentraSite.</p> <p>A lifecycle model name can contain any characters (including spaces), and must be unique within an organization.</p>
Description	<p>(Optional). The description for the lifecycle model.</p> <p>This description appears when a user displays the list of lifecycle models in the Lifecycle Models page.</p>
Version	<p>A user-defined version number for the model.</p>
Organization	<p>The organization to which you want to add the model. (The Organization list only displays organizations for which you have the Manage Lifecycle Models permission.) If you select one of these names, the lifecycle model applies just to objects belonging to the selected organization.</p> <p>You can select ALL option from the drop-down list, implying that the lifecycle model is a system-wide lifecycle model, if you have the Manage System-wide Lifecycle Models permission.</p>

Important:

Select the organization carefully. You cannot change the organization assignment later.

4. In the **States** tab, click **Add State**.
5. In the field labeled **State Name**, type the name of the state.
6. In the field labeled **Description**, type the description for the state.

- In the area labeled **Transitions**, select a state that can be reached as a result of a transition from the current state. In the **Target State** list, you can select the target state from a list of the states that you have defined so far.

Note:

You may find it convenient to define all of the states before you start to define the state transitions.

If you wish to define more than one target state from the current state, click the plus icon next to the target state name.

If you have more than one target state, select one of the target states as the default target state by choosing the radio button in the column labeled **Default**.

You can show or hide the transition area for a state by clicking the chevron in the header line of the **Transitions** area.

- To define additional states, use the **Add State** button as required.
- Click **Save**.

CentraSite validates the newly created lifecycle model.

If there are no validation errors, the lifecycle model is added to the CentraSite registry or repository. If any part of the model is invalid, this is indicated by an appropriate error message.

Adding Lifecycle Stage to Organization

You can access this page only if you have the Manage Lifecycle Models permission for a particular organization in CentraSite.

A lifecycle model can contain one or more stages. Typically, stages are used to represent a clearly-defined deployment scenario within an object's development cycle. You could, for example, decide to use the lifecycle model during various stages of a product lifecycle such as development, test and production. The individual stages are also usually deployed on different physical machines, which is normal practice if for example you have a test environment and a production environment. Each stage you define contains all of the states of the model.

> To add a lifecycle stage

- In CentraSite Control, click **Administration > Lifecycles > Stages**.
- Click **Add Stage**.
- In the **Add Stage** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Name	Name of the lifecycle stage.
Description	(Optional). The description for the lifecycle stage.
Host Name	Name of the host machine where the lifecycle stage is deployed on a CentraSite installation.
Host Port	The port number of the Application Server Tier on the host machine where the lifecycle stage is deployed.
Use SSL	Select the Use SSL check box if the communication with the specified host machine should use the SSL protocol for secure communication.

4. Click **OK**.

The newly created lifecycle stage is added to the CentraSite registry or repository.

Adding State to Lifecycle Model

You can access this page only if you have the Manage Lifecycle Models permission for a particular organization in CentraSite.

> To add a new state to an existing lifecycle model

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.
2. Right-click a lifecycle model you want to add the state, and click **Details**.

This opens the Edit Lifecycle Model page.

3. In the **States** tab, click **Add State**.
4. Provide the values for the fields as required.
 - a. In the field labeled **State Name**, type the name of the state.
 - b. In the field labeled **Description**, type the description for the state.
 - c. In the area labeled **Transitions**, select a state that can be reached as a result of a transition from the current state. In the **Target State** list, you can select the target state from a list of the states that you have defined so far.

Note:

You may find it convenient to define all of the states before you start to define the state transitions.

If you wish to define more than one target state from the current state, click the plus icon next to the target state name.

If you have more than one target state, select one of the target states as the default target state by choosing the radio button in the column labeled **Default**.

You can show or hide the transition area for a state by clicking the chevron in the header line of the **Transitions** area.

5. To define additional states, use the **Add State** button as required.
6. Click **Save**.

CentraSite validates the updated lifecycle model.

If there are no validation errors, the lifecycle model is updated in the CentraSite registry or repository. If any part of the model is invalid, this is indicated by an appropriate error message.

Viewing the Lifecycle Model List

You can view the list of lifecycle models by using the Lifecycle Models page. You can access this page only if you have the Manage Lifecycle Models permission for a particular organization in CentraSite.

By default, users in the CentraSite Administrator or Organization Administrator role have this permission, although an administrator can grant this permission to other roles.

> To view the list of lifecycle models

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.

A list of defined lifecycle models is displayed in the Lifecycle Models page.

2. To filter the list to see just a subset of the available lifecycle models, type a partial string in the **Search** field.

CentraSite applies the filter to the **Name** column. The **Search** field is a type-ahead field, so as soon as you type any characters, the display is updated to show only those lifecycle models whose name contains the specified characters. The wildcard character % is supported.

If you type...	CentraSite Displays
b	Names that contain b
bar	Names that contain bar
%	All names

The Lifecycle Models page provides the following information about each model:

Column	Description
Name	Name of the lifecycle model.
Organization	Name of the organization to which the lifecycle model belongs.
Version	The user-defined version number for the lifecycle model.
Assigned to Types	The object types to which the model applies for lifecycle path.
State	The current state of the lifecycle model (New, Productive, Retired).
Active	The status of the lifecycle model.
Icon	Description
	The lifecycle model is active (can be applied to object types).
	The lifecycle model is inactive (cannot be applied to object types).

You can adjust the view to show or hide the columns by using the **Select Columns** icon that is located in the upper-right corner of the Lifecycle Models page.

The shortcut menu of a particular lifecycle model displays one or more actions that you can perform on that model.

Action	Description
Details	Displays the details page of the lifecycle model.
Delete	Deletes the most recent version of a lifecycle model.
Purge	Deletes the required version and all older versions of a lifecycle model.
Change Lifecycle State	Changes the lifecycle model's own lifecycle status.
Create new version	Creates a new version of the lifecycle model.
Export	Exports the lifecycle model from the registry to an archive file on the file system.
Impact Analysis	Helps to easily visualize the associations that exist between the group and registry objects.
Notify me	Sends notifications when changes are made to the lifecycle model.
Add to List	Adds the lifecycle model to a list in My Favorites .
Add to Favorites	Adds a shortcut to the lifecycle model you want to use routinely or otherwise keep close at hand.

Viewing Lifecycle Model Details

You can view the detailed lifecycle model information by using the Edit Lifecycle Model page. You can open this page only if you have the Manage Lifecycle Models for a particular lifecycle model in CentraSite.

➤ To view the details of a lifecycle model

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.

A list of defined lifecycle models is displayed in the Lifecycle Models page.

2. Right-click a lifecycle model for which you want to display the details, and click **Details**.

The Edit Lifecycle Model page is displayed.

The area labeled **Lifecycle Model Information** displays the generic attributes that includes details about basic lifecycle model information - Name, Description, Version, State, Organization, Owner.

The lifecycle model details are displayed in the following tabs:

- **States:** Displays a list of states, state transitions, stages, and policies that are defined for the lifecycle model.

In the **States** tab, you can add states to the lifecycle model, remove states from the lifecycle model, define the state transitions, define the stages, and display the policies associated with the lifecycle model.
- **Associated Types:** Displays a list of asset types that are associated with the lifecycle model.

In the **Associated Types** tab, you can associate asset types with the lifecycle model and revoke the existing associations with the lifecycle model.
- **State Permissions:** Displays a list of permissions that are defined for the states of a lifecycle model.

In the **State Permissions** tab, you can assign permissions to users or groups and remove permissions from users or groups for the states of a lifecycle model.

Modifying Lifecycle Model Details

You can modify an existing lifecycle model information by using the Edit Lifecycle Model page. You can access this page only if you have the Manage Lifecycle Models permission for a particular organization in CentraSite.

You cannot modify a lifecycle model that is currently in the Productive state. To modify a lifecycle model in the Productive state, you must do one of the following:

- Set the state of the lifecycle model to New.

- Create a new version of the lifecycle model.

➤ To modify the details of a lifecycle model

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.
2. Right-click a lifecycle model for which you want to modify the attributes, and click **Details**.

This opens the Edit Lifecycle Model page.
3. In the area labeled **Lifecycle Model Information**, modify the values of the attributes in the respective data fields as required.
4. To modify the states and state transitions, stage transitions, and the associated policies of the lifecycle model, click **States**. Modify the values of the attributes in the respective data fields as required.
5. To modify the associated asset types, click the **Associated Types** tab. Assign one or multiple asset types to the lifecycle model, and remove one or multiple asset types from the lifecycle model, as required.
6. To modify the permissions, click the **State Permissions** tab. Assign one or multiple users or groups with the state permissions, as required.
7. Click **Save**.

CentraSite validates the updated lifecycle model.

If there are no validation errors, the lifecycle model is updated in the CentraSite registry or repository. If any part of the model is invalid, this is indicated by an appropriate error message.

Defining Stage Transitions for Lifecycle Model

To define a stage transition for the lifecycle model, you must have the Manage Lifecycle Models permission for the particular organization in CentraSite.

You can define a stage transition for the lifecycle model on the Edit Lifecycle Model page.

You define the stage transition from an end state of the current stage to one or more new stages. An end state is a state that has no transitions to other states.

The term *promotion* is used to describe the transition of an asset from one stage to the next stage.

➤ To define stage transitions for the lifecycle model

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.
2. Right-click a lifecycle model you want to define the state transition, and click **Details**.

This opens the Edit Lifecycle Model page.

3. In the **States** tab, click the **Stages** button for the required state.

A list of the defined stages is displayed in the **Select State** dialog box.

4. In the **Select State** dialog box, select one or multiple stages that can be the target of a transition from the selected state.
5. Click **OK**.
6. Click **Save**.

CentraSite validates the updated lifecycle model.

If there are no validation errors, the lifecycle model is updated in the CentraSite registry or repository. If any part of the model is invalid, this is indicated by an appropriate error message.

Associating an Asset Type with Lifecycle Model

When a lifecycle model is assigned to an asset type and the lifecycle model is activated (that is, set to the state Productive), all existing assets of this type in the user's organization are assigned to the first state of the lifecycle model. Also, each subsequently created asset of the given asset type in the user's organization is automatically associated with the lifecycle model and is initially set to the first state of the lifecycle model.

An asset type can only be associated with one lifecycle model at a time. If a lifecycle model includes an asset type that is already associated with another lifecycle model, an error message is displayed when you try to activate the lifecycle model.

If you need to add another asset type to a lifecycle model that is already in the Productive state, you must create a new version of the lifecycle model and add the asset type to the new version.

> To associate a lifecycle model to an asset

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.
2. Right-click a lifecycle model for that you want to associate asset types, and click **Details**.
3. In the Edit Lifecycle Model page, click the **Associated Types** tab.

A list of available asset types that can be associated with the lifecycle model is displayed.

4. To define that a particular asset type must be associated with the lifecycle model, select an appropriate asset type from the **Available Types** list, then use the arrow buttons to copy the asset type into the **Selected Types** list.

Asset types that are already associated with another lifecycle model are displayed with the name of the lifecycle model beside them. For example, if the asset type *Application* is already associated with the lifecycle model *MyModel*, this will appear as *Application (MyModel)*.

Asset types that are associated with a system-wide lifecycle model will only appear in the list of available asset types if you have the Manage System-wide Lifecycle Models permission.

Note:

If you apply a lifecycle model to an asset type that has Virtual Service types associated with it, the model will automatically be applied to those Virtual Service types when both of the following conditions are met:

- The Virtual Service type's **Inherit Base Type LCM** option is selected.
- The Virtual Service types does not have a lifecycle model assigned to it.

Viewing Policies Associated with Lifecycle Model

You can define policies for the lifecycle model by using the Edit Lifecycle Model page. You can access this page only if you have the Manage Lifecycle Models permission for a particular organization in CentraSite.

You can define one or multiple policies to be triggered when a state in the lifecycle is entered.

➤ To define policies associated with the lifecycle model

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.
2. Right-click a lifecycle model you want to define the policies, and click **Details**.

This opens the Edit Lifecycle Model page.

3. In the **States** tab, click the **Policies** button for the required state.

A list of the defined policies for the state, and whether or not the policies are currently active or inactive is displayed.

A policy can be assigned to a particular state when you create a policy.

If you want to see the details of any of the given policies, select the policy name. This opens the **Policy Details** dialog.

Setting Permissions on Lifecycle State

You can set permissions on each state of a lifecycle model by using the Edit Lifecycle Model page. You can access this page only if you have the Manage Lifecycle Models permission or at least have the instance-level Modify permission for a particular lifecycle state in CentraSite.

You define the user and group permissions associated with each state of the lifecycle model. Each permission defines whether a user or group can move assets into a particular lifecycle state. Such permissions are referred to as state permissions.

If you leave the list of users and groups empty, CentraSite grants state permissions to all users and groups. If the list contains at least one user or group, permissions are denied for all other users and groups who are not in the list.

➤ **To assign permissions on a lifecycle state**

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.
2. Right-click a lifecycle model you want to assign the user and group permissions, and click **Details**.
3. In the Edit Lifecycle Model page, click the **State Permissions** tab.
4. In the **State Permissions** tab, click the **Add Users/Groups** button.
5. In the **Add Users/Groups** dialog box, perform the following:
 - a. To filter the list of users and groups, type a partial string in the **Search** text box.
 - b. Select one or multiple users and groups for whom you wish to grant or deny permission to change the lifecycle state of an asset.
 - c. Click **OK**.

CentraSite applies the filter to the **Name** column.

Examples

String	Description
b	Displays names that contain b.
bar	Displays names that contain bar.
%	Displays all users and groups.

- d. Select one or more states into which the user or group is allowed to move an asset. You can either select the check boxes for the states individually, or select them all by selecting the corresponding check box in the **ALL** column. To cancel the selection of a state, clear the corresponding checkbox.

Note that there is no column corresponding to the initial state of an asset, because all assets under control of this lifecycle model are automatically put into the initial state.

6. Click **Save**.

CentraSite validates the updated lifecycle model.

If there are no validation errors, the lifecycle model is updated in the CentraSite registry or repository. If any part of the model is invalid, this is indicated by an appropriate error message.

Activating Lifecycle Model

To be able to activate a system-wide lifecycle model, you must have the Modify Assets permission for all organizations that own the lifecycle model's assigned asset types. This is because when a system-wide lifecycle model is activated, assets of its assigned asset types from all organizations is set to the initial state of the lifecycle model and this requires the modify permission for the assets.

Lifecycle models are themselves governed by a predefined lifecycle model. This lifecycle model defines three states: New, Productive, and Retired.

When you initially create a lifecycle model it enters the New state. CentraSite does not begin enforcing a new lifecycle model until you *activate* the model by switching it to the Productive state.

After you activate a lifecycle model (that is, place it in the Productive state), that lifecycle model can no longer be modified. To make changes to the lifecycle model, you must create a new version of the model and make your changes to the new version.

Note:

After you retire a lifecycle model, that model cannot be activated again. The Retired state is an end state for lifecycle models.

The following general guidelines apply when activating a lifecycle model:

- You cannot activate a lifecycle model that is currently in the Retired state. To activate a lifecycle model in the Retired state, you must create a new version of the lifecycle model.
- Lifecycle models can be applied not only to assets but also to other lifecycle models. CentraSite provides a default lifecycle model that is applied automatically to all user-defined lifecycle models. This default lifecycle model defines the following states:

State	Description
New	A user-defined lifecycle model has been saved but is not yet been activated for use with its associated asset types.
Productive	A user-defined lifecycle model has been activated for use with its associated asset types. The lifecycle status of assets of the associated asset types is visible in the detail view of the asset instances.
Retired	The lifecycle model is no longer in use and cannot be reactivated.

- When you create a user-defined lifecycle model and assign it to one or more asset types, the lifecycle model is initially inactive (that is, in the New state).

> To activate a lifecycle model

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.
2. Right-click a lifecycle model you want to activate, and click **Change State**.
3. In the **Change State** dialog box, select **Productive**.

CentraSite sets the state of the lifecycle model to Productive.

4. Click **OK**.

Versioning a Lifecycle Model

You can create a new version of a lifecycle model by using the Create New Version action in the Lifecycle Models page. You can access this page only if you have the Manage Lifecycle Models permission for a particular organization in CentraSite.

If you need to make changes to a lifecycle model after the model has been activated (that is, after you place it in the Productive state), you must either create a new version of the existing model or replace the existing model with a completely new model. You cannot modify a lifecycle model directly after it has been activated.

The easiest way to apply changes to a lifecycle model is to generate a new version of the model. This task involves the following basic steps:

1. Creating a new version of the model. During this step, CentraSite creates an exact copy of the existing lifecycle model.
2. Updating the new version of the lifecycle model as necessary (for example, adjusting its state permissions, inserting additional states, removing states, modifying transition paths, applying it to additional object types).
3. Activating the new version. This step automatically activates the new version and retires the old version.

When you activate a new version of a lifecycle, instances of assets that were created under the old lifecycle model will automatically switch to the new lifecycle model if they are in a state that exists in the new model. Otherwise, they will continue to follow the old lifecycle model until they are switched to a state that exists in both models. At that point, they will switch to the new lifecycle model.

You can also apply changes to a lifecycle by defining an entirely new lifecycle model. To put the new model into effect, you must retire the existing model and then activate the new model. When you change a lifecycle this way, the objects that were created using the old model will complete their lifecycles under the old model. Objects that are created after the new model is activated will follow the new model.

The following general guidelines apply when versioning a lifecycle model:

- You can only create a new version from the latest version of a lifecycle model. For example, if a lifecycle model already has versions 1.0, 2.0, and 3.0, CentraSite allows you to create a new version of the lifecycle model from version 3.0. It makes no difference whether the lifecycle

model that you are versioning is active or inactive. You can version a lifecycle model in either mode.

- A lifecycle model can only be updated when it is in the New state.
- When a lifecycle model is in the Productive or Retired state, you cannot change the state of the current version of the model back to New; in this case, you can only reach the New state by creating a new version of the lifecycle model.
- CentraSite automatically establishes a relationship between the new version of the lifecycle model and the previous version. This relationship enables several capabilities and features in CentraSite that relate to versioned lifecycle models.

> To version a lifecycle model

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.
2. Right-click a lifecycle model for which you want to create a new version, and click **Create new version**.

When you create a new version, it is not yet activated. The lifecycle state of the new version is New, and the lifecycle version of the previously used version is still Productive.

The new version of the lifecycle model is displayed in the Lifecycle Models page.

3. Right-click the new version of the lifecycle model, and click **Details**.
4. In the Edit Lifecycle Model page, change values of the attributes in the respective data fields as required.
5. In the field labeled **State**, click **Change State**.
6. In the **Change State** dialog box, select **Productive**.

CentraSite sets the state of the new version of the lifecycle model to Productive.

This automatically changes the state of the previously used version from Productive to Retired.

All existing assets of the asset types that use this lifecycle model are automatically set for use with the new version.

When you change a lifecycle model that is already in use, you must ensure that all of the states that were in use in the old model are also available in the new model. If the old state model contains states that no asset instance is currently using, these states do not need to be present in the new model. The state transitions in the new model do not depend on the state transitions in the old model; you can define the state transitions in the new model as you please.

Deleting State from Lifecycle Model

You can delete a lifecycle state by using the Edit Lifecycle Model page. You can access this page only if you have the Manage Lifecycle Models permission for a particular organization in CentraSite.

A state can only be deleted if there is currently no policy that is triggered when this state is entered.

➤ To delete a state from a lifecycle model

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.

A list of defined lifecycle models is displayed in the Lifecycle Models page.

2. To filter the list to see just a subset of the available lifecycle models, type a partial string in the **Search** field.

CentraSite applies the filter to the **Name** column. The **Search** field is a type-ahead field, so as soon as you enter any characters, the display is updated to show only those lifecycle models whose name contains the specified characters. The wildcard character % is supported.

3. Right-click a lifecycle model whose state you want to delete, and click **Details**.

This opens the Edit Lifecycle Model page.

4. In the **States** profile, select a state or multiple states, and click **Delete**.

5. Click **OK** in the confirmation dialog box.

6. In the other states of the model, remove all transitions to the deleted state.

7. Click **Save**.

Each selected state is removed from the lifecycle model.

Deleting Lifecycle Models

Pre-requisites:

You can delete an existing lifecycle model by using the Lifecycle Models page. You can access this page only if you have the Manage Lifecycle Models permission for a particular organization in CentraSite.

The following general guidelines apply when deleting lifecycle models in CentraSite:

- To delete a lifecycle model, you must first deactivate the lifecycle model. This means that you should change the lifecycle model's own lifecycle status from Productive to Retired.
- If several versions of the lifecycle model exist, CentraSite offers two commands for deleting versions, namely **Delete** and **Purge**. The Delete command deletes the newest version of the

lifecycle model; if you use the Delete command on an older version, the command is rejected. The Purge command deletes the requested version and all other versions that are older than the requested version.

Command	Description
---------	-------------

Delete	Deletes the newest version of a lifecycle model. This command cannot be used on an older version of the lifecycle model.
Purge	Deletes the selected version and all older versions of a lifecycle model. If the selected version is the newest version, this command removes the lifecycle model altogether from CentraSite.

- When you purge old versions of a lifecycle model, any existing newer versions does not be deleted. In this case, assets governed by this lifecycle model continues to be governed by this lifecycle model.
- If you have several versions of a lifecycle model and one of the older versions is still in the Productive state, you cannot purge any newer version, since this would automatically try to delete the older version that is in Productive state. This can happen, for example, if you have just created a new version of a lifecycle model but have not yet set the new version to the Productive state.
- If you purge the newest version of a lifecycle model, you automatically delete all of the older versions also, that is, you remove the lifecycle model altogether from CentraSite. In this case, all assets that were governed by the lifecycle model are now of course no longer governed by the lifecycle model. You do not need to adapt these assets in any way for non-lifecycle usage and they are treated by CentraSite as if they had never been governed by a lifecycle model. The only visible change is that when you display the detail view of such an asset, the lifecycle status of the asset is no longer displayed.

Note:

The lifecycle model for lifecycles and the lifecycle model for policies cannot be deleted. Also, if several versions of such a lifecycle model exist, none of the versions can be purged.

Important:

If you have selected several lifecycle models where one or more of them are predefined models, you can use **Delete** to delete them. However, as you are not allowed to delete predefined lifecycle models, only models you have permission for is deleted. The same applies to any other lifecycle models for which you do not have the required permission.

Deleting Most Recent Version of Lifecycle Model

> To delete the most recent version of a lifecycle model

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.

A list of defined lifecycle models is displayed in the Lifecycle Models page.

2. Right-click a lifecycle model that is currently in Productive state, and click **Change Lifecycle State**.

This opens the **Change State** dialog.

3. In the **Change State** dialog box, select the **Retired** option button.

CentraSite sets the state of the lifecycle model to Retired.

4. Click **OK**.

5. Right-click the required version of lifecycle model you want to delete, and click **Delete**.

6. Click **OK** in the confirmation dialog box.

The selected version of the lifecycle model is deleted. All older versions of the lifecycle model does not be affected.

Deleting a Version and All Older Versions of Lifecycle Model (Purging)

➤ To delete a version and all older versions of a lifecycle model

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.

2. To purge any version of the lifecycle model that is currently in Productive state, right-click the required version of the lifecycle model, and click **Change Lifecycle State**.

This opens the **Change State** dialog.

3. In the **Change State** dialog box, select the **Retired** option button.

CentraSite sets the state of the lifecycle model to Retired.

4. Right-click the required version of lifecycle model you want to delete, and click **Purge**.

5. Click **OK** in the confirmation dialog box.

The selected version and all older versions of the lifecycle model is deleted. Any newer versions of the lifecycle model does not be affected.

9 Asset Management

■ Introduction to Asset Catalog	354
■ Customizing Your Asset Catalog	355
■ Executable Design/Change-Time Actions on Your Asset Catalog	366
■ Managing Assets through CentraSite Business UI	379
■ Managing Assets through CentraSite Control	549
■ Managing Assets through Command Line Interface	616
■ RAML to CentraSite REST API Mappings	634
■ Swagger to CentraSite REST API Mappings	641
■ Asset Navigator	648

Introduction to Asset Catalog

The *Asset Catalog* enables CentraSite users to view and manage assets in the CentraSite Registry or Repository.

The asset catalog functions as the central registry of reusable assets within a development environment. When initially installed, CentraSite's asset catalog supports several types of assets, such as Web Services, REST Services, OData Services, XML Schemas. However, an administrator can configure the asset catalog to hold additional assets that are customized for your environment. For example, the catalog at your site might be configured to hold items such as reusable Java libraries or portlets in addition to the basic set of assets that CentraSite supports.

Not all operations are allowed for all users. A user's role and the instance-level permissions on an asset determine which assets a user is allowed to access and what operations the user is allowed to perform on the asset.

- Provider users use the asset catalog to view and manage their organization's assets, publish assets, modify asset details, and assign permissions to assets.
- Consumer and Guest users use the asset catalog to browse the catalog and search for assets.

Each entry in the asset catalog represents a single asset such as a Web Service, REST Service, OData Service, XML Schema. An entry is composed of a set of attributes, a set of profiles, and optionally one or more associated files.

An *attribute* represents a specified characteristic or property of an asset. All assets have the basic set of attributes. In addition to the basic set of attributes, an asset can also have any number of extended attributes that are specific to the asset's type. For example, a Web service asset includes attributes that identify the service's endpoints, provide links to additional documentation, and indicate whether the service is stateful or stateless. The specific set of extended attributes an asset has depends on how your administrator has configured the asset's type.

A *profile* is a visual grouping of a set of attributes. The set of profiles that display for an asset, as well as the specific attributes that appear within the profiles, vary by asset type.

You can browse or search for assets by using a keyword, or by performing an advanced search that sorts and filters the results.

Browse

CentraSite Business UI: You can obtain an alphabetical list of assets available in the CentraSite registry.

CentraSite Control: You can obtain a hierarchical view of assets, organized and filtered according to a selected taxonomy.

Search Using Keywords

CentraSite Business UI: You can search for assets whose attributes contain a certain keyword. Additionally, you can search for assets on the basis of classifiers (for example, Assets, Everything) that are defined in the customization file.

CentraSite Control: You can search for assets whose attributes contain a certain keyword.

Narrow Your Search Using Logical Operators and Additional Search Modifiers

CentraSite Business UI: You can search for assets on the basis of several search criteria (keywords, asset types, lifecycle models, taxonomies, and attributes) using logical ALL or ANY combinations.

CentraSite Control: You can search for assets and supporting documents on the basis of several search criteria using logical AND or OR combinations.

Customizing Your Asset Catalog

CentraSite's flexible and extensible registry structure enables you to model any kind of asset that you might want to include in your asset catalog. It supports a rich set of attribute types for defining the different properties and qualities of your assets. These types include attributes that you can use to classify an asset according to a predefined or custom taxonomy and attributes that you can use to associate the asset with other objects in the registry.

The three major aspects of your catalog that you can customize are: types, taxonomies, and associations.

Creating Custom Types

A *type* (also called an *object type*) describes a kind of object that the registry can store. Besides defining the set of attributes that make up an object, a type includes several system properties that determine, among other things, whether objects of the type are visible in the user interface, whether objects of the type can be used with reports and policies, and whether they can be versioned.

CentraSite includes many predefined types. You can customize many of these predefined types and also create custom types of your own.

Note:

Types are system-wide objects, which means that they apply to all organizations. Consequently, all organizations within a particular instance of CentraSite use (or have access to) the same global set of types.

Object Types and Asset Types

All items stored in the CentraSite registry are *objects* of a particular type. Users, policies, and taxonomies are examples of objects that are stored in the registry. An *asset* is a specific kind of object that represents an artifact in your SOA environment such as a Web service, a REST service, XML schema or a business process. In other words, all assets are objects, but not all objects are assets.

The asset catalog represents the set of all objects in your registry that are assets. Many features within CentraSite operate specifically on the contents of the asset catalog.

Any custom type that you add to CentraSite is considered to be an asset type. Consequently, all instances of a custom type are treated as assets.

Customizing the Predefined Asset Types

CentraSite is installed with a number of predefined asset types, including types that represent Web services, REST services, OData services, XML schemas and BPEL processes. Before using these types in your environment, you should examine their type definitions and customize them as necessary.

With respect to customizing the predefined asset types installed with CentraSite, you can:

- Add attributes to the type
- Group certain attributes to profiles
- Specify which profiles are to be displayed for the type
- Change the type's system-property settings (for example, specify whether the type supports versioning or can be used with design/change-time policies)

Creating Custom Asset Types

Besides customizing the predefined asset types that are installed with CentraSite, you can also define custom types of your own. For example, if you wanted to include items such as service requests, IT projects, and source code libraries in your registry, you would create a custom type for each of these entities.

Note:

Before creating a custom type, always check to see whether CentraSite provides a predefined type that you might be able to customize and use. Customizing one of CentraSite's predefined types saves time, especially if the type requires a file importer.

Before creating a custom type, you must first decide which aspects of an entity you want to model in the registry. If you were creating a type to represent IT projects, for example, you might want to capture characteristics such as the name of the project requester, the lines of business the project is expected to affect, the project plan, the project manager and the project's expected completion date. After you decide which specific characteristics and qualities you want to model, you can create a custom type that includes a corresponding attribute for each of those characteristics or qualities.

Project		
Release Date	DateAndTime	Current target date for release.
Business Owner	String	Functional unit for which the project is being performed.
Status Reports	File	Weekly reports on project status.
Project Plan	File	Current project plan.
Managed By	Relationship	Project Manager(s)

Attributes for asset type "Project"

Assigning Attributes to a Type

An attribute holds data about an asset. All asset types include a basic set of attributes for general information such as the asset's name, description, creation date, and owner. You define additional attributes to hold data that is specific to the type of asset that you want to store in the registry.

When you define an attribute, you specify:

- The type of data that the attribute holds (for example, String, Number, Boolean)

- Whether the attribute holds a single value or multiple values (that is, an array)
- Whether an attribute is required or optional
- Whether the attribute is read-only

Besides basic data types such as String, Number, and Boolean, CentraSite supports the following special types:

Attribute Type	Description
Classification	This type enables users to classify an asset according to a specified taxonomy.
Relationship	This type enables users to establish an association between an asset and another object in the registry.
File	This type enables users to link an asset to a file that resides in CentraSite's repository or exists at a URL-addressable location on the network.

The inclusion of these attribute types facilitate many of the advanced features in CentraSite. It is a good idea to make use of them whenever possible.

Rather than using a String attribute to identify the project manager in this type, you could use a Relationship attribute instead. A Relationship attribute not only identifies the individual who is serving as the project manager, but also provides the additional benefit of enabling users to obtain detailed information about the project manager (because the attribute itself will link users to the actual User object for that individual).

Assigning Attributes to Profiles

A profile defines a collection of attributes that are meant to be grouped together for presentation purposes.

The attributes associated with a profile are displayed on individual tabs in CentraSite Control

The screenshot shows the CentraSite interface for an asset named 'Sales Analyzer'. The top navigation bar includes 'Home', 'Asset Catalog', 'Policies', and 'Administration'. Below this are 'Browse', 'Search', and 'Supporting Documents' buttons. The asset details form shows:

- Name:** Sales Analyzer
- Description:** Project to develop data mining tool for the Sales organization.
- Project Version:** 1.0

Below the details is a tabbed interface with 'Project Info', 'Permissions', and 'Object-Specific Properties' tabs. The 'Project Info' tab is active, showing a table of attributes assigned to a profile. An annotation 'You assign attributes to a profile' points to the table.

Attribute	Value(s)
*Business_Owner:	Global Sales and Marketing
*Release_Date:	2010-09-08
Status_Reports:	Weekly status 20100901 Weekly status 20100908
Project_Plan:	Project Plan for Sales Analyzer
Project_Manager:	Lupica, Marie - User

When you define a new asset type, you specify on which profiles the type's attributes are to be displayed.

All asset types include several generic profiles. Among others, these include:

- **Audit Log profile:** Displays the history of changes to the asset (including changes in an asset's lifecycle state).
- **Consumers profile:** Displays the users and applications that are registered consumers of an asset.
- **Permissions profile:** Displays instance-level permissions for an asset.
- **Classifications profile:** Displays an asset's classifiers.
- **Associations profile:** Lists the registry objects to which the asset is related.

The information on the generic profiles is generated by CentraSite. You cannot customize the content of these profiles or add attributes to them. You can, however, select the profiles you want CentraSite to include when it displays an asset of a defined type.

To display the attributes that you define for an asset type, you create custom profiles and assign the attributes to them. CentraSite does not require an attribute to be assigned to a profile. However, if you do not assign an attribute to a profile, the attribute is not be visible in the user interface. You can assign an attribute to multiple profiles if you want it to appear on multiple profiles (tabs) in the user interface.

Note:

If you want to provide different views of an asset to different users or groups, divide the attributes among profiles in a way that enables you to use profile permissions to selectively show or hide the appropriate set of attributes to different users or groups.

Creating Custom Asset Types that can be Imported from an Input File

The CentraSite Control user interface enables users to add assets to the registry in the following ways:

- Users can create an asset from scratch, which means that they manually assign values to the asset's attributes in the CentraSite Control and CentraSite Business UI interfaces.
- Users can import an asset from an archive file (a file that contains objects that have been exported from an instance of CentraSite).
- Users can import the asset from an input file. To add an asset in this way, CentraSite must be configured with an importer that can read the input file and generate an instance of the specified asset type from it.

CentraSite includes importers for the following types of assets:

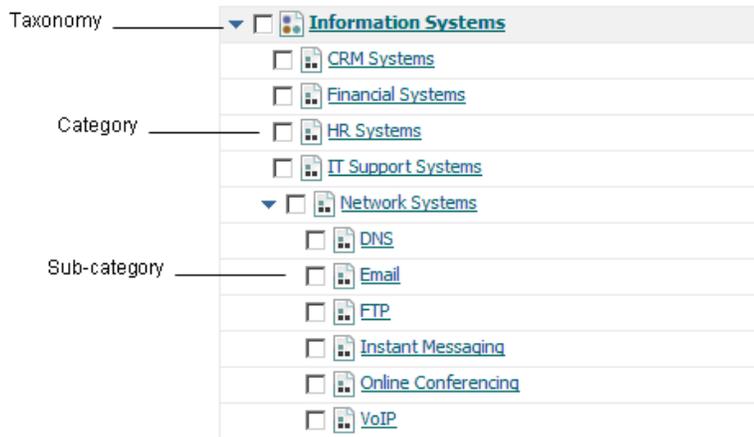
Type of Asset	Required Input File
Web Service	Web Services Description Language (WSDL) file
REST Service	<ul style="list-style-type: none"> ■ RESTful API Modeling Language (RAML) file ■ JavaScript Object Notation (JSON) file ■ Yet Another Markup Language (YAML) file
OData Service	Entity Data Model (EDMX) file
XML Schema	XML Schema Definition (XSD) file
Business Process	Business Process Execution Language (BPEL) file

If you want your users to be able to generate an instance of a custom asset type from an input file, you must build a custom importer and register it in CentraSite. You can find information about developing a custom importer in the *CentraSite Administrator's Guide*.

Defining and Using Taxonomies

A *taxonomy* is a hierarchical classification scheme. In CentraSite, you use taxonomies to classify objects in the registry. Taxonomies enable you to filter, group and sort the contents of the registry.

A taxonomy consists of a name and zero or more *categories*. A category represents a classification within the taxonomy. A category can have multiple levels of sub-categories.

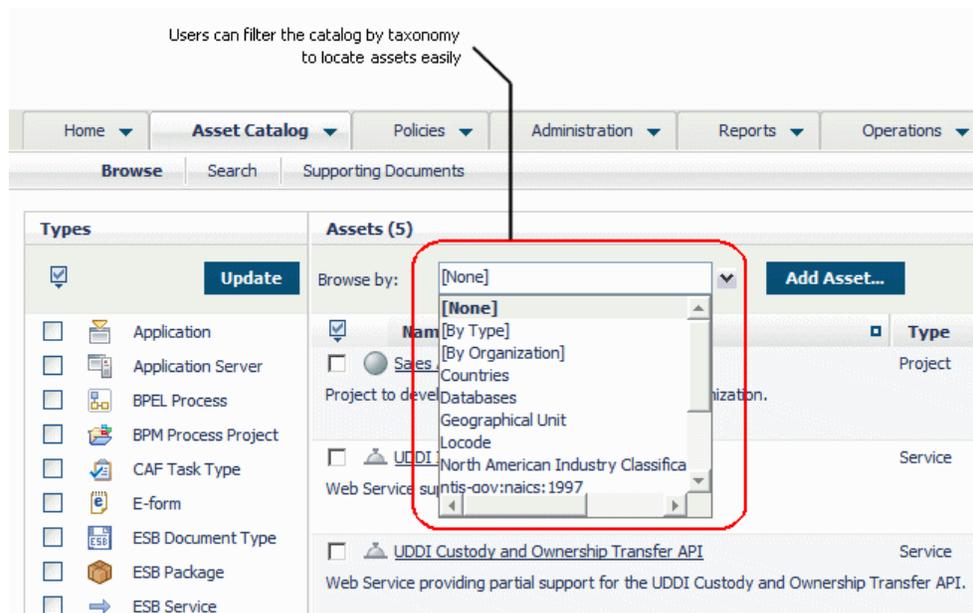


When users publish assets to CentraSite, they can classify the assets in two ways:

- By directly assigning values to an asset's *Classification attributes*. If an asset's type includes one or more Classification attributes, users can classify the asset by simply setting these attributes.
- By assigning ad hoc classifiers to an asset's *Classifications profile*. This profile enables users to classify an asset by any available taxonomy defined in CentraSite. It allows users to assign classifiers to an asset in cases where the asset itself does not include any explicit Classification attributes or does not include the needed type of Classification attribute.

Using Taxonomies to Locate Assets

Classified assets are easier for users to locate because CentraSite includes convenient tools for filtering, reporting, and querying the registry by taxonomy. For example, the Browse page in CentraSite Control enables users to browse the asset catalog according to a specified taxonomy. Additionally, the advanced search feature in CentraSite Control enables users to query the registry for assets that are classified a particular way. By classifying assets, you enable users to discover them using these tools.

**Note:**

If you want users to be able to browse the asset catalog by a taxonomy, you must enable the `Taxonomy is browsable` property. If this property is not enabled, it will not appear in the catalog browser's **Browse by** drop-down list.

Using Taxonomies to Target the Execution of Design/Change-Time Policies

Design/change-time policies execute when events within the policy's scope occur in the registry. The scope of a policy specifies to which type of registry objects the policy applies (for example, Service objects, Policy objects, User objects) and during which types of events the policy is triggered (for example, a PreCreate event, a PostCreate event, a PreStateChange event).

Classifying assets can help you create highly targeted design/change-time policies, because the scope of a policy can be additionally constrained to objects that are classified in a specified way. For example, instead of applying a particular policy to all Application Server assets, you might want to restrict the policy to just the Application Server assets that are classified by the APAC category from the Domains taxonomy.

When you define a custom asset type, think about whether you want to apply different design/change-time policies to specific subsets of that type. If so, make sure the asset type includes a Classification attribute that can be used to distinguish those subsets. (Consider making this a required attribute to ensure that you do not forget to classify assets of this type.)

The Scope of a Taxonomy

Like types, taxonomies are system-wide objects, which means that they apply to all organizations (that is, all organizations have access to the same global set of taxonomies). You cannot restrict a taxonomy to a specific organization.

Taxonomies are also visible to all users. You can give specific user's Modify or Full instance-level permissions on taxonomies, but you cannot revoke a user's View permission. All users (including guest users) can view the taxonomies defined within an instance of CentraSite.

The Predefined Taxonomies

CentraSite installs a number of standard taxonomies that you can use to classify assets.

These include:

- ISO 3166 Country Codes
- North American Industry Classification System 2002 (NAICS)
- ThomasNet Supplier Registry
- Product and Service Category System: United Nations Standard Products and Services Code (UNSPSC)

CentraSite also includes a number of special-purpose taxonomies that it uses for its own internal classification of registry objects.

You cannot delete any of the predefined taxonomies installed with CentraSite or modify their category structure. You can, however, modify certain attributes and properties for these taxonomies. Additionally, you can suppress them in the user interface. For example, if your users will never use the NAICS taxonomies that CentraSite provides, you can remove these taxonomies from the user interface.

Defining Custom Taxonomies

In addition to using the taxonomies that CentraSite provides, you can create your own custom taxonomies.

When you include a Classification attribute in a type, you usually need to create a corresponding taxonomy for the attribute (unless the required taxonomy already exists in the CentraSite registry). For example, let's say you decide that you want to classify your Application Server assets according to the domain in which they reside. To do this you would first create a custom taxonomy that identifies the various domains in your environment. Then, after the taxonomy exists, you would customize the Application Server asset type and add a Classification attribute to it that enables users to classify application server assets by the Domain taxonomy.

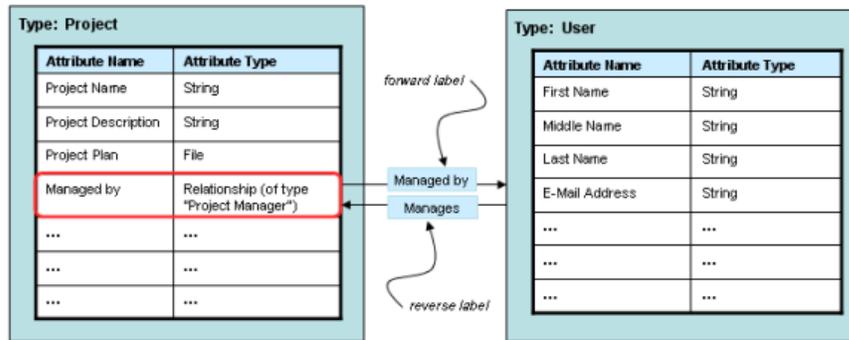
Creating Custom Association Types

An *association* type describes a type of relationship that can exist between objects in the registry.

An association type has a name, a forward label (which describes the relationship of the source object to a target object) and an optional reverse label (which describes the relationship of the target object to the source object).

You use association types to define Relationship attributes in an asset type. In the following example, a Relationship attribute called Managed By has been included in the Project asset type to associate a project asset with the user that manages the project.

You Use Association Types to Define Relationship Attributes in Object Types



Using Association Types to Relate Assets to Other Objects

When users publish assets to the registry, there are two ways in which they can relate an asset with other objects in the registry.

- **By establishing the relationship using an asset's Relationship attributes.** If an asset's type includes one or more Relationship attributes, users can relate an asset to other objects in the registry by simply setting these attributes.
- **By establishing an ad-hoc association using the asset's Associations profile.** If an asset's type includes the Associations profile, users can relate assets of that type with other objects on an "ad hoc" basis. Using this profile, users can relate an asset to virtually any other object in the registry (assuming they have View permission on the target object).

Using Association Types and Relationship Attributes to Support Impact Analysis

CentraSite's Asset Navigator feature helps to visualize the dependencies between assets. Users can determine the dependent assets using the Asset Dependency usecase.

When you include Relationship attributes in an asset type, you not only enable users to specify the objects to which an asset is related, you enable the relationships to be discovered and reported by the Asset Navigator feature.

Creating Custom Association Types

CentraSite provides numerous predefined association types for you to use to create Relationship attributes. However, you can also create custom association types as needed.

Like types and taxonomies, association types are system-wide objects. They apply to all organizations defined in the registry (that is, all organizations within an instance of CentraSite have access to the same global set of association types). You cannot restrict the use of an association type to a specific organization.

Working with Asset Types, Taxonomies and Association Types in a Multi-Stage Environment

If you are working in a multi-stage environment, it is important to master your custom asset types, taxonomies, and association types on one stage and then promote them to the other stages. *Do not* attempt to manually define these objects in each stage. Doing this will create objects that are equivalent, but not identical. That is, the objects will have the same attributes, but they will not

have the same Universally Unique Identifier (UUID). It is the UUID that uniquely distinguishes an object in the registry.

When objects are imported into CentraSite (either through an import process or a promotion process), CentraSite uses the UUID to determine whether an object that you are importing already exists on the target instance of CentraSite. If the UUID does not already exist, CentraSite adds the imported object to the registry. If an object with the same UUID exists in the target registry, and the object that you are importing has a more recent timestamp than it, CentraSite automatically replaces the object in the target registry with the one that you are importing.

Important:

Because the import process uses timestamps to determine whether an object in an archive file is more recent than the one that exists in the target registry, it is important that the system clocks on all of the participating stages are synchronized.

When you promote an asset from one stage to another, CentraSite also promotes the asset's type and the taxonomies that it uses. If you have manually defined these objects on the target instance of CentraSite, they will be duplicated, instead of replaced, during the promotion process. This will create a confusing situation wherein you have two instances of the same asset type and/or taxonomy on the target instance of CentraSite.

To avoid this condition, always create your custom asset types, taxonomies, and association types on the first stage of a multi-stage deployment and export those objects to the registries that host the subsequent stages of the lifecycle.

Note:

Association types that the asset uses are not automatically exported with an asset. If the asset uses custom association types, you must export the association types separately and import them on the other stage(s) before you import the asset itself.

Issues to Consider when Customizing Your Registry

CentraSite provides many ways for you to customize the asset catalog. After you install CentraSite, you should customize the predefined asset types provided by CentraSite (if necessary) and create new types, taxonomies and association types as required for your site.

Note:

Although you should do the initial customization after CentraSite is installed, you can always add additional asset types, taxonomies and association types as you develop the need for them.

When customizing the catalog for your site, keep the following points in mind:

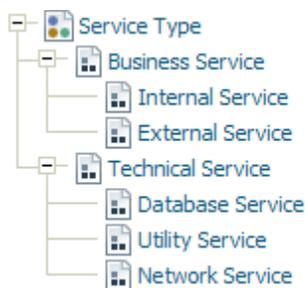
- You can customize any of the predefined asset types installed with CentraSite by adding attributes to them and/or modifying the content and organization of the profiles associated with the type.
- If you have an asset that is not represented by one of the predefined types provided by CentraSite, you must create a custom asset type for it. If you want users to be able to generate the asset type from an input file, you must also create a custom importer for that type and register the importer in CentraSite.

- Consider using Classification attributes and Relationship attributes instead of ordinary String attributes whenever possible. Among other benefits, these attribute types enable users to more easily discover assets and understand the relationships that an asset has with other objects in the registry.
- In general, use a Classification attribute or an enumerated String instead of an ordinary String attribute when you want the attribute to be more strongly typed.
- Instead of defining multiple asset types to represent variants of the same basic type, consider creating one basic type and using a classification attribute to differentiate them. For example, instead of creating separate asset types for different kinds of Web services (for example, business services, technical services, security services), use the one basic Web service asset type and use a Classification attribute to classify its variations.
- When you are designing a new asset type, think about the design/change-time policies that you might want to apply to assets of that type. If you need to apply different policies to different sub-sets of the asset type, use a Classification attribute to differentiate the sub-sets.
- If you do not want users to be able to assign ad hoc classifiers and/or associations to instances of a particular type of asset, omit the Classifications and Associations profiles from that asset's type.
- If a taxonomy is designed to be used with specific types of assets, specify those types in the taxonomy's **Applicable to Object Types** tab. This will prevent users from using the taxonomy to classify objects with which the taxonomy was not intended to be used.

Note:

This functionality applicable only to CentraSite Control has been marked as deprecated and will be removed in a future release.

- You can define taxonomies with multiple levels of sub-categories to create very fine-grain levels of classification. When you do this, users can search for assets that are classified by a specific category *or any of its sub-categories*. For example, the Service Type taxonomy shown in the figure below would enable users to locate a specific type of technical service (for example, a Utility or a Network service) or all technical services (that is, all services that are classified by the Technical Services category or by any of its sub-categories).



- If you are working in a multi-stage environment, always master your custom asset types, taxonomies and association types on one stage and export them to the other stages. *Do not attempt to define these objects manually on each stage.*

Executable Design/Change-Time Actions on Your Asset Catalog

This section describes various executable actions on assets.

Lifecycle State Changes

CentraSite provides the ability to define and track the lifecycle of an asset by using a state model. CentraSite allows the use of active policies to govern specific transitions in the lifecycle management process of an asset.

The lifecycle management (LCM) system for an asset helps to:

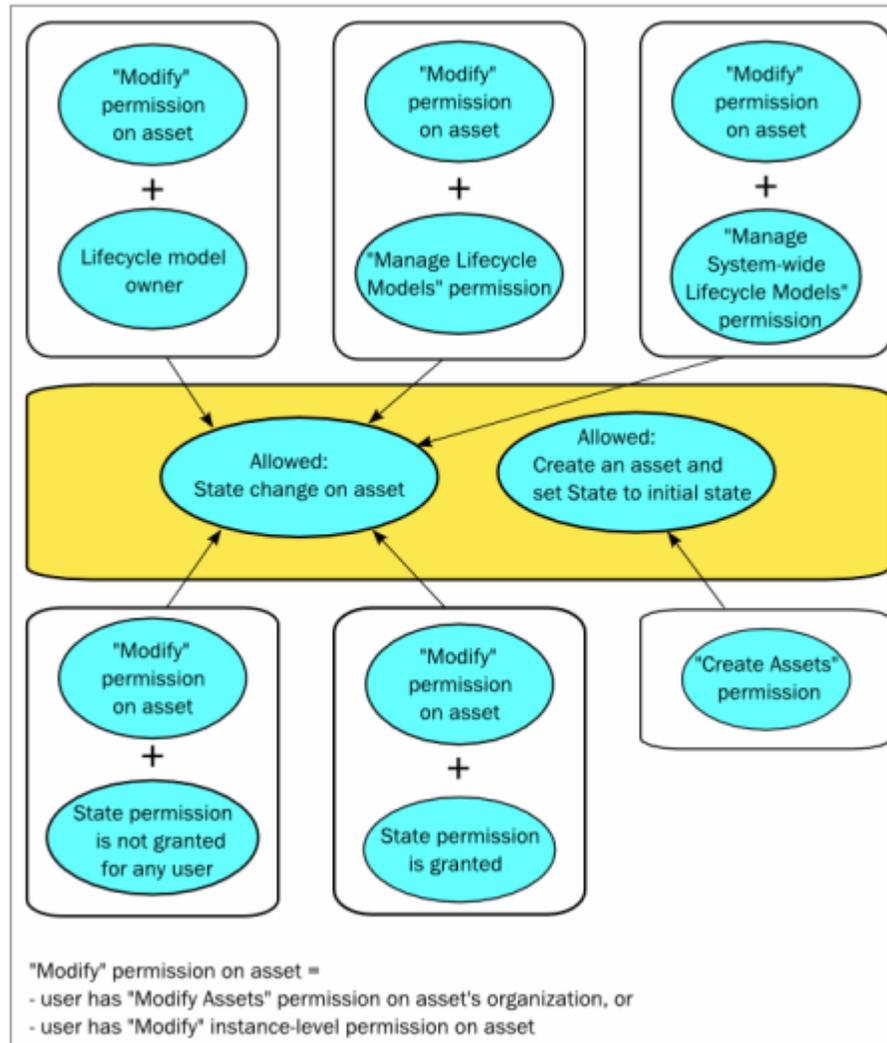
- Assess change impact and manageability across all service consumers
- Ensure service quality through an integrated lifecycle approval process
- Enable a single viewpoint for service stages and their artifacts

Typically, assets such as Web services pass through different states (designing, implementation, testing) before they can be used in a production environment. As the number of objects in a registry grows, it is necessary to introduce stages (Development, Test, Production) to provide adequate operational environments for all parties involved in the lifecycle. Furthermore, an organization may want to add conditions and rules for passing an object through the lifecycle. Therefore the registry should allow administrators to define roles and permissions and connect these to lifecycle steps.

Rules for changing Lifecycle state

For any given lifecycle model, a list of names of users and groups who are allowed to move assets to new states is maintained within the definition of the lifecycle model. For each user or group, the permission to move assets to new states can be restricted to a subset of the available states in the model. When the lifecycle model is assigned to an asset and a state has users or groups defined for it, only a user who is one of the defined users or groups can make the transition of the asset into that state. If no users or groups are defined for a particular state, any user who has Modify permission on the asset can change the lifecycle state for that asset.

Several rules determine who can change the lifecycle state of an asset. These rules are summarized in the following diagram.



For example: If you are the owner of a lifecycle model, you can assign any lifecycle state of this lifecycle to an asset whose asset type has this lifecycle model assigned, as long as you have the Modify Assets permission for the asset.

Any user who has the Create Assets permission can create an asset whose asset type has a lifecycle model assigned. When the asset is created, CentraSite automatically sets the lifecycle state of this asset to the initial state. However, to change the state from the initial state to another lifecycle state, the user requires the appropriate permissions as described above.

Note:

Manage Assets permission does NOT include the rights to change lifecycle states.

Ownership Changes

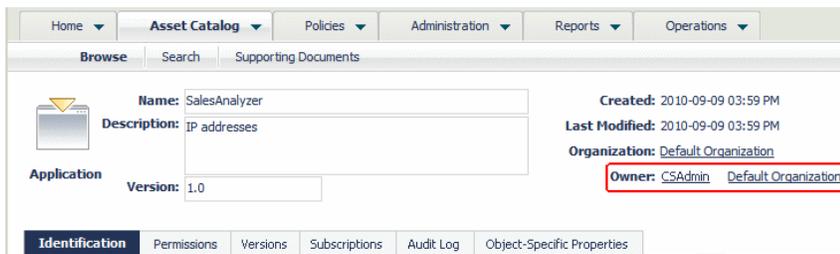
In CentraSite, there are two concepts of ownership. An asset belongs to a particular *user* (known as the asset's *owner*) and it also belongs to a particular *organization*. The owner of an asset has special access rights to the asset and serves as the asset's main point of contact. The asset's organization determines whose rules of governance apply to the asset.

After an asset is created, it is sometimes necessary to change its ownership. For example:

- You may want to transfer an asset *to another user* if the original owner leaves the company, transfers to another position, or is otherwise unable to continue serving as the owner of an asset.
- You might need to transfer ownership of an asset *to another organization* when the asset reaches a point in its lifecycle where it is managed by a different group of users. When a service moves into production, for example, you may want to transfer it to your operations organization.

User Ownership

The user who adds an asset to the catalog automatically becomes the asset's owner. User ownership is specified by the asset's **Owner** attribute, which appears on the asset's details page.



The owner of an asset automatically receives Full permission on the asset. The owner also participates in various processes and policies that affect the asset. For example, the owner of an asset is responsible for reviewing and approving all consumer-registration requests that users submit against the asset.

When you change ownership of an asset, you transfer all the permissions and responsibilities associated with ownership of the asset to another user.

Note:

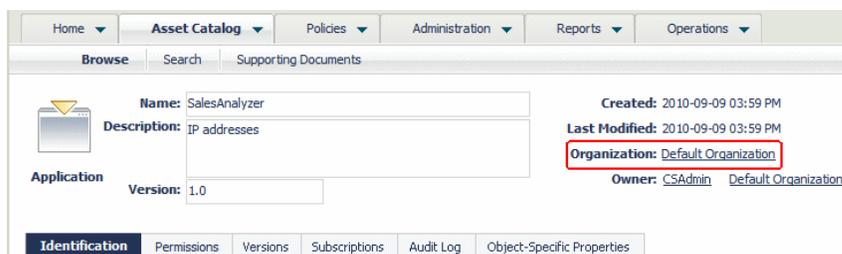
Certain predefined assets that are installed with CentraSite are owned by an internal user known as the *default user*. You cannot transfer assets to or from this user.

Organizational Ownership

The organizational ownership for an asset is specified by the asset's **Organization** attribute. The organization to which an asset belongs determines which policies apply to the asset, which lifecycle model it follows, and which group of users have implicit permission to view the asset. In other words, it determines whose rules of governance apply to the asset.

An asset's **Organization** attribute is specified when a user adds the asset to the catalog. Users can add assets to any organization for which they have Create Assets permission. (Most users only have permission to create assets in their own organization, so most assets in the registry belong to the same organization as their owner.)

The organization to which an asset belongs is shown in the **Organization** attribute on the asset's details page.



Note:

In some parts of the user interface and the CentraSite API for JAXR documentation, the organization to which an asset belongs is referred to as the *submitting organization*. This is simply another way of referring to the organization that is specified in the asset's **Organization** attribute.

Effect of Ownership Changes on Registry Objects

When you change the ownership of an asset, CentraSite modifies the asset's Owner and Organization attributes in the way you specify. Additionally, CentraSite:

- Records the ownership change in the audit log.
- Triggers pre- and post-update policies that exist for the asset.
- Sends a notification to the inbox of the asset's previous owner and the new owner. This behavior can be suppressed by modifying a parameter of the Default Move Handler action that is activated by the Default Move Handler.
- Updates the asset's instance level permissions (if the asset is transferred to a different user).
- Updates the asset's lifecycle state (if the asset is transferred to an organization that has its own lifecycle model for the asset's type).

Before transferring an asset to another user and/or organization, review this information so you understand how the asset will be affected.

Effect of an Ownership Change on Permission Assignments

When you transfer ownership to another user, the Full permission that CentraSite implicitly grants to the owner of an asset is transferred to the new owner and taken away from the previous owner (the previous owner retains existing explicit permissions on the asset). CentraSite makes no changes to the instance-level or role-based permissions currently associated with the asset. This means that:

- All instance-based permissions that are assigned to the asset will remain in effect after the transfer. For example, if group ABC currently has Modify permission on an asset, group ABC will continue to have Modify permission on the asset after the ownership change.
- All role-based permissions remain as is. If a user currently has access to the asset through an organization-level role-based permission, the user loses access to the asset if it is transferred to another organization. CentraSite Control makes no attempt to preserve a user's access to the asset by adjusting the user's role-based permissions. If you change the organizational

ownership of an asset, you should review the role-based permission settings in the receiving organization afterwards to ensure that the asset is available to all the users who need it.

Policies that are Triggered During an Ownership Change

CentraSite treats an ownership change as an *update to the asset*. Thus, changing the ownership of an asset triggers the execution of any pre-update and post-update policies that apply to the asset. If a pre-update policy fails, the ownership of the asset is not changed.

Note:

When you transfer an asset to a different organization, CentraSite applies the policies of the *receiving organization* to the asset.

Effect of an Ownership Change on Objects Associated with the Asset

If you transfer a composite asset to another user or organization, CentraSite automatically changes the ownership of all the asset's nonshared components.

Other than changing the nonshared components for a composite type, CentraSite does not change the ownership of any objects, assets, or repository artifacts that are associated with the asset. For example, if an Application asset has a Uses relationship with a Service asset, changing the ownership of one asset in this relationship does not change the ownership of the other.

After you transfer an asset to a new owner, review the asset and ensure that the new owner has permission to access the objects with which it is associated. Adjust the permission settings on those assets as necessary to ensure the new owner has access to them.

Effect of an Ownership Change on Other Versions of the Asset

Changing the ownership of an asset that is versioned does not affect any previous or later versions of the asset. When you transfer the ownership of a particular version of an asset, CentraSite transfers just that version. Other versions of the asset are not affected.

Effect of an Ownership Change on the Asset's Lifecycle State

If you transfer an asset to a different user, but you do not change its organization, the asset's lifecycle state is not changed. However, if you transfer an asset to another organization, the asset's state can change depending on the lifecycle model (LCM) that is in effect for the asset's type in the receiving organization.

The following table describes how the asset's lifecycle state is affected during a transfer to another organization:

If the originating organization uses...	And the receiving organization uses...	Then...
No LCM for the type	No LCM for the type	The asset's lifecycle state does not change (that is, it remains unset).
No LCM for the type	An organization-specific LCM for the type	The asset's lifecycle state switches to the initial state of the receiving organization's LCM.

If the originating organization uses...	And the receiving organization uses...	Then...
The system-wide LCM for the type	The system-wide LCM for the type	The asset's lifecycle state does not change.
An organization-specific LCM for the type	No LCM for the type	The lifecycle state is removed from the asset.
An organization-specific LCM for the type	An organization-specific LCM for the type	The asset's lifecycle state switches to the initial state of the receiving organization's LCM.

Download Documents

CentraSite Control offers two methods of retrieving the source files of CentraSite assets, namely *exporting* and *downloading*. The source file is the file that was imported into CentraSite in order to create the registry entry for the asset. For example, the source file for a web service asset is the service's WSDL file. The source file for an XML schema asset is its schema file. The difference between exporting and downloading is as follows:

- The *export* feature creates a zip file containing one or more assets from the repository, as well as all associated registry objects.
- The *download* feature creates a zip file containing just the source file of a single asset from the repository, without any of the associated registry objects. If the source file refers to other source files in the repository (for example, a WSDL file can reference XML schema files), the referenced files will also be included in the zip file. If the asset refers to files in the Supporting Document Library, these can optionally be included in the zip file.

If an asset was not created by an importer, but was instead created from scratch without using a source file, the download feature can still be activated. In this case, however, the downloaded zip file does not contain an asset source file but instead only contains files from the Supporting Document Library that are attached to the asset.

Structure of the Zip File

The zip file is organized as a directory that holds all the downloaded files. Subfolders are only created if any of the names of the downloaded files are not unique; in this case, the files are stored in consecutively numbered subfolders (for example: 1/SchemaA.xsd, 2/SchemaA.xsd, and so on.).

If a downloaded file refers to one or more other downloaded files, for example if a WSDL file refers to a schema, the reference within the file is adjusted so that it points relatively to the file in the zip file. This is also true if the referenced file is located within a numbered subfolder.

Example: The WSDL file `Service.wsdl` refers to `SchemaA.xsd`, to another `SchemaA.xsd` with a different namespace and to `SchemaB.xsd`. The resulting zip file has the following structure:

```
Service.wsdl
1/SchemaA.xsd
```

2/SchemaA.xsd

SchemaB.xsd

Impact Analysis

CentraSite offers the possibility of viewing associations between the registry objects and hence identifying the impact when updating or deleting an asset in the catalog. This is called *Impact Analysis*.

The impact analysis feature enables you to easily navigate and visualize the associations between the catalog assets and registry objects. This feature helps you to:

- Understand asset-to-objects associations by displaying the associations that exist between the catalog assets and other registry objects.
- Check that existing associations between the assets and objects are not violated when you make changes in the registry. Also, check the external links from registry objects to supporting documents.
- Determine the impact that updating or deleting an asset would have on its related objects.

You can visualize the currently defined associations for an asset with other registry objects, either in a graphical or a tabular form. The graphical representation is enabled through a Flash-based web browser.

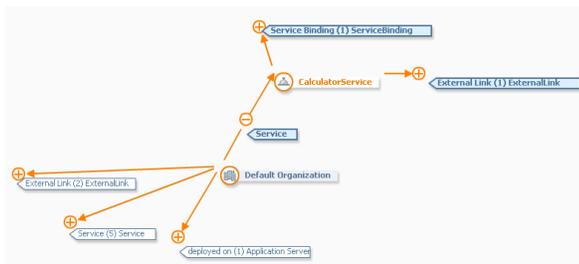
Note that apart from the assets, you can also view the impact analysis for other CentraSite objects like organizations, users, groups, and roles through their shortcut menu.

Graphical Visualization

By default, the impact analysis is shown in a graphical form. You can switch to the tabular view using the link **Switch to Tabular View**. When you are using the tabular view, you can switch to the graphical view using the link **Switch to Graphical View**.

The graphical view shows a visual representation of the selected asset, the objects referred to by the asset, the objects that refer to the asset, and the asset's associations.

Here is an example of the impact analysis diagram for the asset `CalculatorService`, which is an asset of type `Service`:



The asset for which the impact analysis is being displayed is shown in a box with orange colored text (in the example, **CalculatorService**). The objects that are associated with the central asset are

displayed initially in boxes with dark blue text on a lighter background (for example, the node **Default Organization**).

Associations between assets are represented by orange-colored arrows. Each association has a name and a direction (indicated by the arrowhead). For example, the diagram shows the association with the name **Service** that connects the **Default Organization** node to the **CalculatorService** node. This indicates that an association of type Service connects the two nodes. The arrowhead points to the **CalculatorService** node, indicating that **Default Organization** contains a service **CalculatorService**.

The display of the associations can be expanded or collapsed as required. If an association is shown with an orange plus sign, you can click on the plus sign to expand the association; this reveals the node or nodes at the other end of the association, and the plus sign changes to a minus sign. To collapse the association, that is, to hide the other end of the association, click on the minus sign, and the display reverts to its original state.

When you expand an association, the association's background color changes to blue. If you collapse a previously expanded association, its color remains blue; this way you can identify the associations that you have already visited. Associations that have not yet been expanded are displayed with a neutral background (that is, the same background color as the drawing canvas).

The text in the box for any collapsed association shows the following items of information:

- The type of the association.
- The number of currently invisible target nodes that are attached to the visible source node.
- The object type of the invisible target node(s).

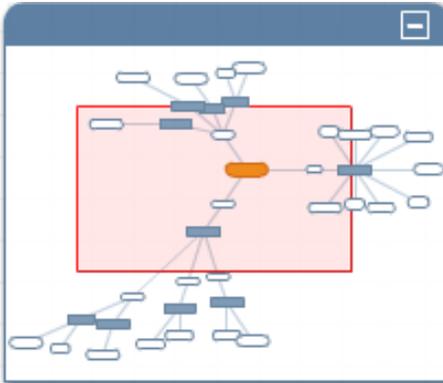
So, for example, the association labeled **deployed on (1) Application Server** in the diagram indicates an association of type **deployed on** between the visible source node **Default Organization** and a currently invisible node of type **Application Server**.

If you click on an object (as opposed to an association), a window appears with a short summary of the object's definition.

You can move the whole diagram within the web browser display by moving the cursor to an empty part of the diagram and dragging the diagram in the required direction.

You can rearrange the position of any node in the diagram by clicking on the node and dragging it to a new location on the canvas.

The display also contains a bird's eye view of the impact analysis diagram, for example:



The shaded central part is the part that is shown in detail in the full display. You can drag the shaded central part to any location in the bird's eye view, and the focus of the full display moves accordingly. You can minimize the bird's eye view by choosing the - icon. The minimized view shows just a menu bar with a + icon. To restore the view, click the + icon.

Predefined Configurations for Impact Analysis

You can use a filter to restrict the type of objects and associations displayed, and to specify the maximum depth of nesting for the displayed associations.

CentraSite provides the following built-in filter configurations that you can use to visualize the impact analysis for the various objects:

- **Asset Dependencies:** This shows associations that are of particular relevance for assets.
- **Schema Usage:** This shows associations that are of particular relevance for schema objects.
- **Organization Details:** This shows associations that are of particular relevance for organization objects.
- **Service Details:** This shows associations that are of particular relevance for service assets.
- **webMethods Assets:** This shows associations between services and webMethods Suite types like CAF and web applications.

By default, CentraSite displays the Asset Dependencies filter configuration. You can select a configuration that you require to visualize the asset's impact from the **Configuration** tab.

If required, you can choose the filter configuration that you want to customize, and change the filter settings accordingly. If you change any of the filter configurations and click the **Refresh Canvas** icon in the appropriate configuration menu, the display is updated as per your new settings. Any such changes you make apply also in subsequent login sessions. If you want to return to the original settings, delete the filter configuration; this deletes the current settings and restores the configuration to its original state.

Asset Versions

You can use the versioning feature in CentraSite to add an updated version of an asset to the catalog. For example, if you make significant changes to a Service asset (such as adding operations

to the service or modifying the data types that it uses), you can use the versioning feature to add the new version of the service to the catalog.

When you version an asset, you become the owner of the new version of the asset. Ownership is not carried forward from the previous version. The new version of the asset belongs to the same organization as its previous version.

Versioning can be active or inactive for any given asset type. If you are using CentraSite in conjunction with other components of the webMethods Suite, the versioning capability for the asset types defined by these components is by default not activated. Unless the documentation for the webMethods Product Suite components states otherwise, do not activate the versioning for these asset types.

When you create a new version of an asset, CentraSite internally treats the new asset version as a new registry object and assigns a new internal object ID to it. The new version is related to the old version by a *Supersedes* association from the new version to the old version. In cases where the detail page of an asset has a **Summary** profile, the association is displayed under the **Summary** profile.

The metrics and event information that was collected for the old version of the asset remains unchanged in the CentraSite Registry Repository. The old version's metrics and event information will not be copied to the new version. CentraSite begins collecting metrics and event information for the new version of the asset.

CentraSite maintains two sets of version numbers for an asset. One set is maintained for CentraSite's own internal use. CentraSite automatically assigns this version number when you create a new version of an asset. You cannot modify it. The version numbers assigned by CentraSite have the format *<MajorVersion>.<Revision>* and are always sequentially numbered starting from 1.0 (for example, 1.0, 2.0, 3.0). If the revision feature is enabled, the revision number is incremented automatically each time you modify the current version of the asset.

Each version of an asset also has a separate user-defined version identifier. This is the public version number that CentraSite Control shows to users when it displays the catalog. The user-defined version identifier does not need to be numeric. For example, you might use a value such as V2.a (beta) to identify a version.

Note:

Depending on the type of asset you version, some of the attributes are cloned from the original asset and others are not. For example, when you version a Service asset, the settings on the **Classifications** profile are cloned, however, the attribute settings on many of the other profiles, including the **Permissions** profile, are not. After you version an asset, you should always examine the attribute settings for the new version and set them appropriately.

Asset Revisions

Within each version of an asset you can have several revisions. When revision processing is enabled, CentraSite stores a new revision of the current asset version each time you update the asset. In CentraSite Control you can view the stored asset properties for each stored revision.

For example, if you have an asset whose current version is 2.1, you may want to modify the contents of the **Description** attribute of the asset in the asset's detail page, but without creating a new

version. In this case, when you save the new description of the asset, the version number is updated automatically by CentraSite to 2.2.

When revision processing is enabled, the revision number of an asset is initially 1 and is automatically incremented each time you save changes to the asset. When revision processing is disabled, all revisions of an asset except the most recent are discarded and the revision number is automatically reset to 0.

When you create a new revision of an asset, CentraSite internally treats the new asset revision as the same registry object and does not assign a new internal object ID to it.

Currently, when you switch the revision feature on or off, you can only do this for all assets in all organizations; there is no possibility of limiting the effects of revision processing to a subset of the assets or organizations.

By default, that is, immediately after the installation of CentraSite, revision processing is switched off.

Deleting an object also deletes all of its revisions. The constraints for deleting objects however apply only to the current revision. This means that all incoming associations that exist on the current state of the object have to be released before deletion.

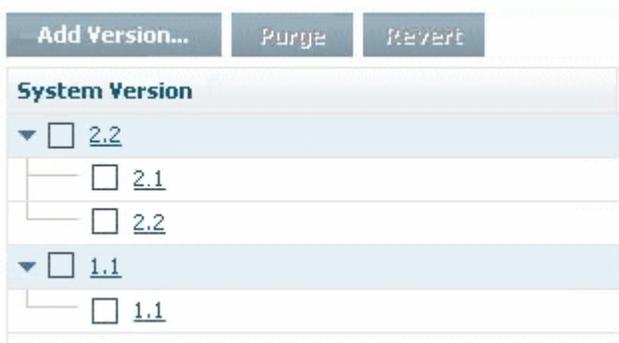
If an object with existing revisions is exported, then only the currently selected revision is exported and the revision history is not exported.

Searching (including Advanced Search) always defaults to the current revision of an object. It is possible to express a revision label in an advanced search. If an advanced search expects revisions to be found, it is not possible to define additional search criteria.

Visualization of Revisions

When revision processing is switched on, you can view the revisions of any given asset by choosing the **Versions** tab in the asset's detail view. If an asset has several revisions, these will be shown. If an asset has not been modified since it was created, the asset's version will be shown with a single revision with the number .1

The following example shows an asset with two versions. Version 2 of the asset has two revisions, namely 2.1 and 2.2. Version 1 is unchanged since it was created and has therefore a revision 1.1:



If you want to view the asset properties stored for any particular revision, choose the link for the required revision. Note that you cannot change the properties of a revision of an asset version if there is a newer revision of the same asset version.

Instance and Profile Permissions

Permissions determine the operations users can perform and the set of objects users can access. You define the user-specific or group-specific permissions of an asset through the **Permissions** profile.

Instance-Level Permissions

Instance-level permissions grant users or groups access to a specific asset in the catalog.

CentraSite allows you to set permissions on an entire asset. This feature enables you to specify the actions they enable a user or group to perform asset in CentraSite Control. For any given asset, you can define View, Modify, Full permissions for different users and groups.

The instance-level permissions that can be set on a given asset for any user or group are:

Permission	Description
View	Enables the specified user or group to the profile when they view the asset.
Modify	Enables the specified user or group to modify the attribute settings in the profile when they view the asset.
Full	Enables the specified user or group to delete the asset.

Profile Permissions

CentraSite allows you to set permissions on individual profiles within an asset. This feature enables you to specify which of the available profiles can be viewed or edited by users when they display an asset in CentraSite Control. For any given asset, you can define different profile permissions for different users. For example, if an asset includes a profile called Source Control that displays links to your source control systems, you may want to restrict the visibility of that profile to authorized developers.

The profile permissions that can be set on a given asset for any user or group are:

Permission	Description
View	Enables the specified user or group to the profile when they view the asset.
Modify	Enables the specified user or group to modify the attribute settings in the profile when they view the asset.

The individual profiles do not include the Full permission because users cannot delete a profile from an individual asset.

Important:

The profile permissions can be used to prevent users from viewing and accessing a particular set of attributes through CentraSite's graphical user interfaces. However, they do not restrict access to the attributes themselves at the asset level.

Propagation of Permissions

An asset can have one or more dependent objects. For example, a Service asset can refer to a WSDL which in turn can refer to one or more XML Schema assets. You can optionally choose whether the permissions assigned to an asset instance should be automatically propagated to the asset instance's dependent objects.

In the context of CentraSite Control, propagation of permissions means that the new permissions completely replace the old permissions; the new permissions are not merged with the old permissions. As an alternative, you can use a change-time policy containing the action Set Instance and Profile Permissions. With this action, you can choose whether the new permissions will be merged with the old permissions or will replace the old permissions.

Propagation of Instance Level Permissions

By default, the access level permissions that are assigned on an asset are implicitly propagated to these dependent objects. This behavior is activated when you select the **Propagate permissions to dependent objects** check box in the asset's **Permissions** tab. For example, assigning Modify permission on a Service asset propagates the Modify permission to the asset's WSDL, schemas, and so on.

If you do not have permission to assign instance-level permissions to a dependent object, the dependent object will not be modified and a warning message is issued.

You can propagate permissions only for the following asset types:

- (Web) Service
- REST Service
- OData Service
- XML Schema
- BPEL

Propagation of Profile Permissions

In addition to propagating permissions that control the access to an asset instance, it is also possible to propagate permissions that control the access to the asset instance's profiles. This means that the profile permissions that you define for an asset instance can be propagated to the asset's dependent objects. However, this is only possible if the dependent object is of the same asset type as the first object; this restriction arises because different asset types can have different sets of profiles.

This behavior is activated when you select the **Propagate profile permissions** check box in the asset's **Permissions** tab. This check box is only available for the following asset types:

- (Web) Service
- REST Service
- OData Service
- XML Schema

Managing Assets through CentraSite Business UI

This section describes operations you can perform to manage assets, such as, web services, REST services, OData services, and Applications through CentraSite Business UI.

Searching and Browsing the Asset Catalog

The set of assets available to you when you search or browse the asset catalog are the assets on which you have the View permission. You can obtain View permission on an asset in the following ways:

- By belonging to a role that includes any of the following permissions.

This permission...	Allows you to...
View Assets	View all assets within a specified organization.
Modify Assets	View and edit all assets within a specified organization.
Manage Assets	View, edit, and delete all assets within a specified organization, and set instance-level permissions on those assets. This permission also allows you to create assets.
Create Assets	Add new assets to a specified organization. You automatically receive Full permission (which implies Modify and View permission) on all assets that you create.

- By having View, Modify, or Full instance-level permissions on an asset.

By default, all CentraSite users belong to the Asset Consumer role. This role includes the View Assets permission for the organization to which a user belongs.

Having the Asset Consumer role gives you implicit view permission on all the assets in your organization. You can view assets from other organizations only if you are given permission to do so through the assignment of additional role-based or instance-level permissions.

Note:

In rare instances, an administrator might not grant view permissions to all of the users in an organization. If the administrator of your organization has done this, you will need instance-level permissions on an asset in order to view it.

Using Search Metacharacters in the Keyword Search

Certain characters have a special function when used in the keyword search:

- Wildcard characters allow you to search for keywords that match a string pattern.
- The quote character (") is used to group keywords into phrases.

- To force the keyword search to treat these metacharacters as normal characters, precede the character with a backslash (\). If you want to include the backslash character itself in the search, type two backslashes.

Using Keywords

You can define the input for the keyword search in the following ways:

- A keyword search consists of 1-n search keywords. Multiple keywords are space separated. If multiple keywords are given, a logical disjunction (OR) is implied.
- A keyword is treated as partial text which can occur at the beginning of the searched strings. The starts with semantics are implied.

Example: If the keyword is `customer`, then the following matches are returned: `A sample svc for customers` as well as `customerservice`.

- As multiple keywords are OR combined, the keywords can match a single phrase (for example, in the description) or individual keywords can occur in different attributes.

Example: If a search is conducted for `customer service`, then `customer` could be matched in the description and `service` in an object specific attribute.

- If quotes (" ") exist around a phrase, then a search is performed on the exact phrase within the quotes. A space within a quoted phrase is considered as a space character and not as a logical operation.
- You can mix and match any number of words and quoted phrases within the keyword field.
- The search is neither case nor accent sensitive, even within a quoted phrase.

Example: A search for `abc` will return the same results as a search for `ABC` or `Abc`.

- If you type a string that contains an odd number of double-quote characters, then the last double-quote character is ignored when the search is performed.
- The simple search can include wildcard characters.

In addition to the general guidelines listed above, CentraSite Business UI has its own search functionality that differs from CentraSite Control in the following ways:

- If the keyword search input field is empty, then the search execution will not happen.
- If an asset name exceeds `n` characters, the name is automatically truncated. You can select the maximum number of characters to display in the asset name using the property `maxCharactersToShow` in the configuration file. By default, the maximum character limit is set to "60".

Using Wildcards

The available wildcard characters are as follows:

Character	Usage
* or %	If you use the percent symbol (%) or the asterisk (*), CentraSite replaces the wildcard symbol with as many characters as necessary to find a match. For example, an entry of A%n returns both Amazon and American. If you type *a1, then CalcService, Calendar and AustralianPostCode all fit the search criteria.
? or _	If you use the question mark (?) or the underscore (_), CentraSite replaces the wildcard symbol with a single character in order to find a match. Example: CustomerSVC?Request matches any character for ?.

You can use a wildcard character at any point in the keyword text and multiple times throughout the keyword text. If you type a wildcard character in the middle of a string, for example `cat*dog`, then at least one of the searched attributes must contain the string in order for the asset or supporting document to be included in the result set.

If a wildcard character between two words is surrounded by spaces, such as `word1 * word2`, the wildcard will match one word.

Note:

Here are some general guidelines:

- Certain non-alphanumeric characters that can appear in the name of an asset are currently ignored by CentraSite's wildcard mechanism when you include them in a keyword search. In particular, the hyphen (-) is ignored. Thus, if you have created the assets `asset-1` and `asset_1`, the wildcard search for `asset?1` will find `asset_1` but not `asset-1`.
- The percent (%) character acts as a word delimiter when it appears in the text to be searched. Thus, for example, if the description field of an asset contains the text `abc%def` (the characters a, b, c, %, d, e, f), this is treated by the search mechanism as two adjacent words `abc` and `def`. A wildcard search such as `abc*def` looks for a single word beginning with `abc` and ending with `def`, so the search will not find this asset.

Searchable Attributes

You can specify generic attributes (that is, attributes common to all asset types) and type-specific attributes as search criteria.

Generic Attributes

The generic attributes, also called, Common Attributes, that can be used as search criteria in the CentraSite Business UI are described in the following table:

Search Attribute	Usage
Name	Use this attribute to search for assets whose name matches a specified text string. You can specify a substring or expression that can be combined with a <code>contains word</code> (default option), <code>starts with</code> , <code>equals</code> , or <code>not equals</code> expression. The search is neither case nor accent sensitive. If <code>starts with</code> is used, no wildcard is necessary as a postfix. If <code>contains word</code> is used, the

Search Attribute	Usage
	<p>word given is treated as a partial string with implicit wildcards. If <code>equals</code> or <code>not equals</code> is used, no wildcards are supported.</p> <p>If multiple substrings have been given the parameters are implicitly quoted. Explicit quotations and wildcards can be used, and behave in the same way as for keyword searches.</p>
Description	<p>Use this attribute to search for assets whose description matches a specified text string.</p> <p>Usage is the same as for the Name attribute.</p>
Created Date	<p>Use this attribute to search for assets with a specified creation date.</p> <p>You can select a date and apply a <i>before</i>, <i>after</i>, <i>on</i>, or <i>between</i> criterion. If <i>between</i> is used, a second input field allows you to specify the end date.</p> <p>The date input parameters allow year, month and day input as well as hour and minute. Hour and minute default to 0. The data format is used as specified in the account preferences of a user (defaults to <code>yyyy-mm-dd</code>). No wildcards are supported.</p>
Last Updated	<p>Use this attribute to search for assets with a specified modification date.</p> <p>Usage is the same as for the Creation Date attribute.</p>
Key	<p>Use this attribute to search for an asset that exactly matches the given UDDI V3 key.</p> <p>If no prefix <code>uddi:</code> is given, this is implied automatically. No wildcards are supported.</p>
Owner	<p>Use this attribute to search for assets belonging to a specified user.</p> <p>Select the user through a Browse list.</p>
Organization	<p>Use this attribute to search for assets provided by a specified organization.</p> <p>Select the organization through a Browse selection list.</p>
Version	<p>Use this attribute to search for a specific version of asset.</p>
Subscribers	<p>Use this attribute to search for users receiving notifications when changes are made to assets.</p>
Consumers	<p>Use this attribute to search for users or applications consuming the asset.</p>

Type-Specific Attributes

In addition to the generic attributes listed in the table above, each asset type can have its own type-specific search criteria, based on the type-specific attributes of the asset type.

The type-specific attributes can be selected in two ways in CentraSite Control:

- Select the entry **Type-specific property** in the drop-down list of generic search criteria, as described in the table above.
- Select one of the additional entries in the **Criteria** list as follows:

The type-specific search criteria are shown in the **Criteria** list in the form `<AttributeName> (<DataType>)`, where `<AttributeName>` is the name of the type-specific attribute, and `<DataType>` is the data type of the attribute.

For example, if you select the asset type **Service** in the **Types** field, the **Criteria** drop-down list contains search criteria like `SOAP-Version (String)`, which refers to the service's type-specific attribute `SOAP-Version` which has the data type `String`.

Depending on the data type of the type-specific attribute you select, the **Criteria** section of the dialog changes to reflect the search possibilities for that data type.

Attribute Data Types and Supported Search Operators

Not all attribute data types support the full set of search operators. Some data types execute only with certain operators.

The following tables lists the supported search operators you can use when searching for attributes depending on their data types.

Data Type	Search Operators and Description
String	Equals
	NotEquals
	StartsWith
	Contains
International String	Equals
	NotEquals
	StartsWith
	Contains
Multiline String	Equals
	NotEquals
	StartsWith
	Contains
Email	Equals
	StartsWith
URL/URI	Equals

Data Type	Search Operators and Description
Number	StartsWith
	Equals
	NotEquals
	Greater
	Smaller
	GreaterorEquals
	SmallerorEquals
Boolean	N/A
Date/Time	Before
	After
	Between
	On
Duration	N/A
IP Address	Equals
	Between
File	Equals
	StartsWith
Classification	N/A. Assumed to Equals.
Relationship	N/A. Assumed to Equals.

Searching the Asset Catalog

CentraSite Business UI supports the following types of search filters:

- Scope
- Keywords (*Type-ahead Search*)
- Modifiers (*Advanced Search*)
- Logical operators

The following sections describe how to locate assets in the catalog using these search filters.

Type-ahead Search

The type-ahead search is an easy to use keyword-based search facility.

The type-ahead search is a simple and easy to use search facility in which you can specify arbitrary search patterns. It helps you choose relevant and popular terms related to your selected key pattern.

You can search for all assets that contain one or more specified keywords (that is, text strings) in the asset's string attributes (name, description, and so on). You may use the **Scope** list to restrict the asset types on which you want to execute the search.

The number of search results is displayed directly below the title line of the Search Results area, for example *About 43 results*. If no results are found, this is displayed as *Your search did not match any data*.

➤ To search the catalog by type-ahead search

1. In CentraSite Business UI, locate the type-ahead search text box in the upper-left corner of the menu bar.
2. In the **Scope** list next to the search text box, choose your search scope.

A list of scopes that are available to you is displayed. By default, CentraSite contains the following predefined scopes:

Scope	Description
Everything	Displays a list of the currently available assets, organizations, and users in CentraSite registry.
Assets	Displays a list of the currently available assets in CentraSite registry. This is the default scope.

3. Type the characters in the type-ahead search text box. You can use a wildcard character at any point in the search text, and multiple times throughout the search text.

As you start to type your search, CentraSite predicts what you are looking for and displays the results for your search. If you do not see the results you want, keep typing and the results will update.

CentraSite executes the search within an asset for the **Name** attribute matching a given character string.

If you type...	CentraSite displays...
b	Names that contain "b"
bar	Names that contain "bar"
%	All names

To view the complete list of assets, leave the Search text box blank, and then click the **Search** icon.

The type-ahead search invokes a panel of recommended results.

The Search Result set is ordered by relevance. Relevance is decided as follows:

- If the search text contains more than one keyword, the assets that match the most keywords are ranked higher.
 - Assets whose basic attributes, for example, Name and Description, match the search text are rated higher than those where other attributes match.
4. Select a result to directly open its details page, or choose the **See more results** option to display the Search Results page.

Use the Up Arrow key and Down Arrow key to scroll one row up or down the list of recommended results.

Advanced Search

CentraSite's advanced search capabilities allow you to build sophisticated search clauses to search for assets on the basis of asset types, taxonomies, and attribute values. The search criteria can be combined by a conjunction (ALL) or disjunction (ANY) operation.

Searching the Catalog by Keyword

> To search the catalog by keyword

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Type in a keyword in the **Keyword** text box.

To view the complete the list of assets, type a wildcard character. Note that if you simply leave this text box blank, CentraSite executes the search using the default scope **Assets**.

You may use the chevron to expand or collapse the **Keyword** panel.

3. To add the specified keyword to the Search Recipe, click the plus symbol next to the text box, or press Enter.

To add additional keywords to the Search Recipe, repeat the above steps.

To delete the keywords that were previously added to the Search Recipe, click the **Remove** button next to the keyword.

CentraSite displays the list of assets that match the specified keyword(s) in the Search Results page.

Searching the Catalog by Taxonomy

The taxonomy search capability allows you to search assets that have been classified according to a specific taxonomy or a taxonomy category.

> To search the catalog by taxonomy

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Taxonomies**. Do one of the following:

- Type in a keyword in the **Taxonomy** text box.

To add the selected keyword to the Search Recipe, click the plus symbol next to the text box, or press Enter. CentraSite displays the list of assets that match the specified keyword.

To add additional keywords to the Search Recipe, repeat the above steps.

To delete the keywords that were previously added to the Search Recipe, click the **Remove** button next to the keyword.

Note that if you simply leave this text box blank, CentraSite executes the search using the default scope **Assets**.

- To search for assets classified by a specific taxonomy or taxonomy category, click **Choose**. This opens the **Choose Taxonomies** dialog box.

This dialog box displays a list of the predefined and user-defined taxonomies and taxonomy categories that are currently browsable in CentraSite.

1. In the **Choose Taxonomies** dialog box, follow these steps:

- a. Click the expand chevron of a taxonomy to display all of the categories of that particular taxonomy.
- b. Select one or more taxonomies and taxonomy categories, and then click **OK**.

CentraSite displays the list of assets that match the specified taxonomies in the Search Results page.

Searching the Catalog by Lifecycle Model

You can search for assets on the basis of lifecycle models and states.

Note:

This functionality is available when the lifecycle model itself is in the Productive or Retired state.

> To search for an asset by lifecycle model

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Lifecycle Models**. Do one of the following:

- Type in a keyword in the **Lifecycle Model** text box.

To add the specified keyword to the Search Recipe, click the plus symbol next to the text box, or press Enter. CentraSite displays the list of assets that match the specified keyword.

To add additional keywords to the Search Recipe, repeat the above steps.

To delete the keywords that were previously added to the Search Recipe, click the **Remove** button next to the keyword.

Note that if you simply leave this text box blank, CentraSite executes the search using the default scope **Assets**.

- To search for assets applied to a specific lifecycle model, click **Choose**. This opens the **Choose Lifecycle Models** dialog box.

This dialog box displays a list of the predefined and user-defined lifecycle models and states that are currently defined in CentraSite.

1. In the **Choose Lifecycle Models** dialog box, follow these steps:

- a. Click the expand chevron of a lifecycle model to display all of the defined states of that particular taxonomy.
- b. Select one or more lifecycle models and lifecycle states, and then click **OK**.

CentraSite displays the list of assets that match the specified lifecycle model(s) in the Search Results page.

Searching the Catalog by Asset Type

The type search capability allows you to search for assets on the basis of asset types.

> To search the catalog by asset type

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**. Do one of the following:

- Type in a keyword in the **Asset Type** text box.

To add the specified keyword to the Search Recipe, click the plus symbol next to the text box, or press Enter. CentraSite displays the list of assets that match the specified keyword.

To add additional keywords to the Search Recipe, repeat the above steps.

To delete the keywords that were previously added to the Search Recipe, click the **Remove** button next to the keyword.

Note that if you simply leave this text box blank, CentraSite executes the search using the default scope **Assets**.

- To search for assets belonging to a specific type, click **Choose**. This opens the **Choose Asset Types** dialog box.

This dialog box displays a list of the predefined and user-defined assets that are currently defined in CentraSite.

1. In the **Choose Asset Types** dialog box, follow these steps:

- a. Click the expand chevron of a scope to display all of the asset types of that particular scope.
- b. Select one or more asset types, and then click **OK**.

CentraSite displays the list of assets that match the specified type(s) in the Search Results page.

Searching the Catalog by Attribute Value

You can further refine your search by specifying asset attributes to search for.

If you specify a single asset type for the search, the set of attributes that you can use for the search is the set of attributes for that asset type. If you have specified several asset types for the search, the set of attributes available for the search is a combination of the respective type-specific attributes.

> To search for an asset by attribute value

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Attributes**.
3. To search for assets containing a specific attribute, click **Choose**.

This opens the **Choose Attributes** dialog box.

4. In the **Choose Attributes** dialog box, follow these steps:
 - a. Click the expand chevron of a category to display all of the attributes of that particular category.
 - b. Select one or more attributes, and then click **OK**.

CentraSite displays the list of assets that match the specified attribute(s) in the Search Results page.

Combining Search Filters

You can specify in which way the search filters should be combined:

➤ To specify how the search filters should be combined

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. After specifying your required search criteria, in the area labeled **Current Search Criteria**, choose one of the following:
 - To specify that an asset must meet all criteria to be considered a match, select **All**.
 - To specify that an asset must meet at least one of the criteria to be considered a match, select **Any**.

Searching the Asset Catalog

CentraSite Business UI supports the following types of search filters:

- Scope
- Keywords (*Type-ahead Search*)
- Modifiers (*Advanced Search*)

- Logical operators

The following sections describe how to locate assets in the catalog using these search filters.

Managing the Search Recipe

The Search Recipe contain the combination of the search filters.

Adding Search Filter to Search Recipe

Consider the following guidelines, when adding a search filter to your Search Recipe:

- Executing a search with empty Search recipe results in a blank Search Results page.
 - If there is no single search filter in the Search Recipe, validation fails and CentraSite Business UI displays an error message.
 - When you add a search filter to the Search Recipe, CentraSite triggers validation for all of the specified search values. If any part of the validation fails, it prompts you with a warning message, and sets the focus to the search filter which holds an incorrect value or duplicate values.
 - When trying to execute a saved search using the deep linking functionality, if the Search Recipe includes two or more search filters (for example, an asset type **Application** and/or an attribute **SOAP Version**) that currently do not exist in the registry, CentraSite Business UI displays an error symbol next to the non-existing search filter with a message *Invalid Search Condition*.
 - If there is a saved search with two or more search scopes, when you remove one of the scopes from the Search Recipe, CentraSite internally revokes the attributes that were specific to the scope that you just removed from the Search Recipe, and displayed in the **Sort by** and **View** list.
 - When trying to add a search filter that already exists in the Search Recipe, CentraSite displays an error symbol next to the duplicate entry (latest) in the Search Recipe. After you remove the existing search filter from the Search Recipe, CentraSite automatically revokes the error symbol from the duplicate entry.
 - When trying to add an invalid search scope using the **Keyword** text box, the invalid scope will not be added to the Search Recipe.
1. In CentraSite Business UI, display the Advanced Search panel in one of the following ways:
 - Click the **Browse** link that is located in the upper-left corner of the menu bar.
 - Click the **Search** icon that is located next to the **Scope** box.

The Search Recipe list is displayed in the area labeled **Current Search Criteria**.
 2. To add a search filter of one of the following types to the Search Recipe box, see the procedures described in the corresponding sections:

Filter Type	Follow the steps in...
Scope	Searching the Catalog by Asset Type
Keywords	Searching the Catalog by Keyword
Taxonomies	Searching the Catalog by Taxonomy
Lifecycle Models	Searching the Catalog by Lifecycle Model
Attributes	Searching the Catalog by Attribute Value

Rendering of Search Scopes in Search Recipe

When rendering the results in the Search Results page, CentraSite considers the search scopes that are currently defined in the Search Recipe in a hierarchical order. This rendering behavior can be best understood with the following scenarios:

- The predefined scope **Everything** overrides all other predefined and custom scopes in the Search Recipe. If you have, for example, the default scope **Assets** already defined in the Recipe, and you add the predefined scope **Everything** to the Recipe, CentraSite automatically removes the default scope **Assets** from the Recipe and returns all the registry objects, for example, organizations, users, and assets, that match with the scope **Everything**. On the other hand, if you have, for example, the predefined scope **Everything** already defined in the Recipe, CentraSite will not allow you to add any other predefined scope and custom scope to the Recipe.
- In the same way, if you have, for example, the predefined scope **Service** already defined in the Recipe, and you add the predefined scope **Assets** to the Recipe, CentraSite automatically removes the existing scope **Service** from the Recipe and returns all the assets that match with the scope **Assets**. On the other hand, if you have the predefined scope **Assets** already defined in the Recipe, CentraSite will not allow you to add any other predefined scope, except for **Everything**, which takes the priority, and custom scope to the recipe.

If at any time you revert back to the Search Results page, the results get rendered based on the search scopes that are available in the Search Recipe.

Removing Search Filter from Search Recipe

1. In CentraSite Business UI, display the Advanced Search panel in one of the following ways:

- Click the **Browse** link that is located in the upper-left corner of the menu bar.
- Click the **Search** icon that is located next to the **Scope** box.

The Search Recipe list is displayed in the area labeled **Current Search Criteria**.

2. In the Search Recipe list, click the **Remove** symbol next to the filter name you want to remove.

This temporarily removes the search filter from the Search Recipe list.

To delete a search filter permanently, remove the required property statement from the CentraSite's customisation file, **centrasite.xml**, and restart Software AG Runtime.

```
<SearchFilterCategories>
  <SearchFilterCategory
    id="Types"
    displayName="CS_MSG_INMBU_STR_ASSET_TYPES" />
  <SearchFilterCategory
    id="Taxonomy"
    displayName="CS_MSG_INMBU_STR_TAXONOMIES" />
  <SearchFilterCategory
    id="FurtherAttribute"
    displayName="CS_MSG_INMBU_STR_FURTHER_ATTRIBUTES" />
  <SearchFilterCategory
    id="LCM"
    displayName="CS_MSG_INMBU_STR_LCM" />
</SearchFilterCategories>
<SearchScopes>
  <SearchScope
    id="Assets" isExpandable="false"
    class="com.softwareag.centrasite.api.csom.search.impl.AssetScope"
    exclude="uddi:7613515f-77eb-11dd-bc9f-f62b6cf80b00">
    INMCL_STR_Assets</SearchScope>
  <SearchScope id="Everything" isExpandable="true"
    class="com.softwareag.centrasite.api.csom.search.impl.EverythingScope"
    exclude="uddi:7613515f-77eb-11dd-bc9f-f62b6cf80b00">
    INMCL_STR_Everything</SearchScope>
  <SearchScope id="Users" isExpandable="false"
    class="com.softwareag.centrasite.api.csom.search.impl.CS0TypeScope"
    types="User" include="uddi:2ebc76b4-128b-11dd-8c31-ae80cb45c029"
    exclude="uddi:7613515f-77eb-11dd-bc9f-f62b6cf80b00">
    INMCL_STR_Users</SearchScope>
  <SearchScope id="Organizations" isExpandable="false"
    class="com.softwareag.centrasite.api.csom.search.impl.CS0TypeScope"
    types="Organization"
    exclude="uddi:7613515f-77eb-11dd-bc9f-f62b6cf80b00">
    INMCL_STR_Organizations</SearchScope>
</SearchScopes>
```

Saving and Re-Executing Saved Searches

After you define a keyword search or an advanced search, you might want to save the search definition, so that you can execute the same search again at a later stage.

This section describes how to save a search and re-execute the saved search.

Note:

A saved search can only be executed by the user who created the saved search.

Saving a Search Definition

➤ To save a search definition

1. In CentraSite Business UI, display the Advanced Search panel in one of the following ways:

- Click the **Browse** link that is located in the upper-left corner of the menu bar.
- Click the **Search** icon that is located next to the **Scope** box.

The Search Recipe list is displayed in the area labeled **Current Search Criteria**.

2. To add a search filter of one of the following types to the Search Recipe box, see the procedures described in the corresponding sections:

Filter Type	Follow the steps in...
Scope	Searching the Catalog by Asset Type
Keywords	Searching the Catalog by Keyword
Taxonomies	Searching the Catalog by Taxonomy
Lifecycle Models	Searching the Catalog by Lifecycle Model
Attributes	Searching the Catalog by Attribute Value

3. In the **Save Your Search** text box, type a name for the new saved search.
4. Click **Save**.

The saved search is displayed using this name in the User Preferences page.

You can manage your list of saved searches using the User Preferences page.

Re-executing a Saved Search

To execute an existing saved search, proceed as follows:

> To re-execute a saved search

1. In CentraSite Business UI, click your user name that is located in the upper-right corner of the header area.

This opens the User Preferences page.

2. In the **Saved Searches** panel, click the link of a saved search you want to execute. This starts the search directly.

A list of results for the saved search is displayed in the Search Results page.

In addition, CentraSite Business UI supports the deep linking functionality for a saved search. You can simply copy the URL of the saved search from your browser and use it as a deep link to directly navigate to the Search Results page.

Creating a New Search from a Saved Search

To create a new search based on an existing saved search, proceed as follows:

➤ To create a new search using an existing saved search

1. In CentraSite Business UI, click your user name that is located in the upper-right corner of the header area.

This opens the User Preferences page.

2. In the **Saved Searches** panel, click the link of a saved search you want to reuse.

You can also use the deep linking functionality of a saved search to display the Search Results page.

This opens the Advanced Search panel containing the filters of the particular saved search.

3. Modify the search filters as required.
4. To save the updated search as a saved search, in the **Save Your Search** text box, type a name for the new saved search.
5. Click **Save**.

The updated search is displayed using this name in the User Preferences page.

Note:

Remember that you cannot overwrite an existing saved search with an updated saved search.

Managing the Search Results page

In CentraSite Business UI the result view is rendered in the Search Results page. The Search Results page includes several components that you can configure to improve the end-user experience.

The relevant results of a query are displayed in the Search Results. By configuring the properties in this Search Results page, you can control how search results are displayed. You can for example:

- Change the number of results that appear on a page
- Change the number of characters in the asset name
- Change the number of characters in the asset description
- Change the number of character in the attribute name
- Use refiners to drill down in search results

Refiners enable you to drill down into the search results based on the searchable attributes, for example creation date, owner, and organization name. The refiners include a collection of

common attributes and the type-specific attributes specific to the asset type that you choose in the **Additional Search Criteria**.

By using refiners, you can narrow the search results to only show search items like for example certain assets, created in a certain time period, created by a given person. Refiners are displayed in the **View** menu that is located just above the result view area.

- Add sort options to search results

You can sort the search results by choosing one of the options from the drop-down list labeled **Sort by** on the Search Results page. The sort options are rendered based on respect to the attributes that were earlier selected in the View menu. Default sorting is by the **Name** field.

You can choose to reorder the list of assets by toggling the ascending or descending arrow located next to the option.

Viewing the Search Results Page

The Browse, Keyword, and Advanced searches return a list of assets that match the search criteria in the Search Results page.

The number of search results is displayed below the title line of the Search Results page, for example About 114 results. If no results are found, then the search result is displayed as "query does not match any existing data" in the CentraSite registry.

The Search Results's **View** box shows various attributes of the assets, such as the name of the asset, the asset type and, the description.

You can perform various actions on the displayed list of assets. If you want to perform an action on just one or several of the displayed assets, you can mark the check boxes of the required assets, and then select an action from the actions menu.

Here is a sample of the available actions:

- Export one or more of the displayed assets.
- Delete one or more of the displayed assets.
- Change the lifecycle state of one or more of the displayed assets.
- Add one or more of the displayed assets to the **My Favorites** list.
- In CentraSite Business UI, display the Search Results page in one of the following ways:
 - Click the **Browse** link that is located in the upper-left corner of the menu bar.
 - Click the **Search** icon that is located next to the **Scope** box.

This displays a list of assets for which you have View permission in CentraSite.

Configuring the Search Results Page

The results appear by default on the Search Results page. When you query the CentraSite registry with a simple search or an advanced search, if more than one page of results is returned, the Search

Results displays page numbers and forward and back arrows. You can edit the properties of the Search Results to change how it looks. This includes how many page links appear and how they are displayed.

Change the Number of Results that Appear on Search Results page

1. Open the customization file, `centrasite.xml`, in a rich text editor.

You can find the **centrasite.xml** file on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\custom\conf`.

2. Locate the property statement: `<SearchResult noOfRows="20" />`.
3. Type the maximum number of results you want to appear in the Search Results page before pagination is required.

The maximum is 50 results. The default is 20 results. Increasing this number affects how quickly the user interface renders.

Change the Number of Characters in Attribute Display Name

1. Open the customization file, `centrasite.xml`, in a rich text editor.

You can find the **centrasite.xml** file on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\custom\conf`.

2. Locate the property statement `<AttributeDisplaySize>10</AttributeDisplaySize>`.
3. Type the maximum number of characters to truncate an attribute's name.

The default is 32 characters.

Browsing the Asset Catalog

With the Browse feature in CentraSite Business UI, you have the following options:

- View the complete list of assets.
- View a list of assets whose name attribute contains a certain keyword (character string).
- View a list of assets that belong to certain asset types.
- View a list of assets that have been classified according to a specific taxonomy or lifecycle model.

Important:

In CentraSite Business UI, when the **Exclude sub types from Business UI search** check box is selected in a base asset type, for example, in a Service type definition, you cannot browse for asset instances of its sub type, Virtual Service, in the Search Results page. You can only browse

for asset instances of Service type. In this scenario, to browse for asset instances of the Service type, you must manually add the Service type to Search Recipe.

You can select or clear the **Exclude sub types from Business UI search** check box in the **Edit Asset Type - Advanced Settings** dialog of the base type definition. However, to select or clear the selection, you must have the Manage Asset Types system-level permission.

> To browse the catalog

- In CentraSite Business UI, click the **Browse** link in the upper-left corner of the menu bar.
CentraSite displays a list of assets for which you have View permission in the Search Results page.

Web Service Management

This section describes operations you can perform to manage web services through CentraSite Business UI.

Adding Web Service from Scratch

Pre-requisites:

To add a Web Service asset to an organization's asset catalog, you must belong to a role that has the Create Assets or Manage Assets permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

Before you add a Web Service asset to the catalog, you must have the WSDL specification file that you want to import. This file can reside on the file system of the computer where your browser is running or it can reside anywhere on the network, as long as its location is addressable through a URL.

> To add a Web Service asset from scratch

1. In the CentraSite Business UI activity bar, click **Create Asset**.

This opens the **Create New Asset** wizard.

2. In the **Basic Information** profile, provide the required information for each of the displayed fields:

Field	Description
Name	(Optional). Name of the Web Service asset.

Field	Description						
	<p>Note: This is the name that users will see when they view this Web Service in the CentraSite User Interfaces. Therefore, we recommend that you specify a meaningful name for each Web Service.</p> <p>A Web Service name can contain any characters (including spaces), and must be unique within an organization.</p> <p>If you have not specified a name for the Web Service, CentraSite automatically populates the Name field with the data extracted from the WSDL URL or the WSDL file name.</p>						
Type	The asset type, Service .						
Organization	The organization to which you want to add the Web Service. (The Organization list only displays organizations for which you have the Manage Assets permission or at least the Create Assets permission.)						
Version	<p>(Optional). The version identifier for the Web Service. You can enter any string in this field, that is, the version identifier does not need to be numeric. You can also leave the field blank. You can later create new versions of the Web Service. The default is 1.0.</p> <p>Examples:</p> <pre>0.0a 1.0.0 (beta) Pre-release 001 V1-2007.04.30</pre>						
Description	<p>(Optional). The description for the Web Service.</p> <p>Note: This is the description information that users will see when they view this Web Service in the CentraSite User Interfaces. Therefore, we recommend that you specify a meaningful description for each Web Service.</p>						
Import a File	<p>The input WSDL file for the Web Service. You may want to read the input WSDL file from a URL-addressable location on the network (the URL option) or from your local file system (the File option).</p> <table border="1"> <thead> <tr> <th>Option</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>URL</td> <td>If the WSDL file you are importing resides on the network, you can specify its URL.</td> </tr> <tr> <td>File</td> <td>If the specification resides in your local file system, specify the file name. You can use the Browse button to navigate to the required folder.</td> </tr> </tbody> </table>	Option	Description	URL	If the WSDL file you are importing resides on the network, you can specify its URL.	File	If the specification resides in your local file system, specify the file name. You can use the Browse button to navigate to the required folder.
Option	Description						
URL	If the WSDL file you are importing resides on the network, you can specify its URL.						
File	If the specification resides in your local file system, specify the file name. You can use the Browse button to navigate to the required folder.						

3. Click the **Advanced Settings** chevron to expand the additional options that are available for the Web Service asset. Provide the required information for each of the displayed fields:

Field	Description
Credentials	If you have specified a URL and the site you want to access through the URL requires user authentication, type a username and password for authentication at the URL site.
Resolution	Select a resolution strategy, which will allow you to specify how an already existing imported and included file is handled. For each of the imported and included files you have one of these options: <ul style="list-style-type: none">■ Create new version: Creates a new version of the file with the new content (if, for example, you want to modify a WSDL file but want to retain its previous version).■ Always overwrite: Overwrites the importing file with new content.

4. Click **Next**.

You cannot navigate to the next screen unless all the required attributes have been set.

5. In the **Preview** panel, review the basic information for the Web Service before you actually add to the CentraSite registry.
6. Click **Save**.

A Web Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the Web Service asset that you just created is displayed.

7. Configure the extended attributes of the Web Service asset as described later in this topic.

Tip:

If you had previously imported a WSDL file that has an associated schema file and you now re-import just the schema file with modifications, your browser may not display the updated contents of the schema file. This can happen if the browser cache is not being updated automatically. To rectify the problem, you can change your browser settings so that pages are always updated on every visit.

Adding Web Service using an Archive

Pre-requisites:

To add a Web Service asset to an organization's asset catalog, you must belong to a role that has the Create Assets or Manage Assets permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

Before you import a Web Service asset to the catalog, you must have the archive file (.zip file). This file must reside on the file system of the computer where your browser is running.

You can import a Web Service asset using the archive file (.zip file) to which the Web Service was previously exported. You can import Web Services into the same CentraSite registry from which they were originally exported or to a different CentraSite registry.

➤ To add a Web Service asset using importer

1. In the CentraSite Business UI activity bar, click **Create Asset**.

This opens the **Create New Asset** wizard. The bottom panel of the **Create New Asset** wizard displays the option to import an archive file.

2. To import an archive file, click **Choose** to navigate to the folder where the exported archive file of a Web Service asset resides, and select the file.

When you select a file to import, the fields in the area labeled **Basic Information** cannot be modified.

3. Click **Next**.

The **Create New Asset** wizard displays a list of the referenced objects for the Web Service to import.

4. The check box next to each referenced object indicates whether the object should be imported. By default, all objects displayed are included in the import set. To exclude any referenced object from the import set, clear the corresponding check box.
5. Click the **Advanced Settings** chevron to expand the additional options that are available for the Web Service asset. Provide values for each of the displayed options:

Option	Description
Change Owner	The imported Web Service can be assigned to the same owner as in the source CentraSite registry, or you can assign a new owner. The Change Owner field is type-ahead field. As you type characters in this field, the dialog box lists the user names that match the characters you specify.
Change Organization	When you import a Web Service, you can import it into the same organization in the target CentraSite registry as in the source CentraSite registry from which they were exported, or you can assign a new organization.

Option	Description
	The Change Organization field is type-ahead field. As you type characters in this field, the dialog box lists the organization names that match the characters you specify.
Retain lifecycle state	This option determines whether the lifecycle state of the imported Web Service is preserved. Enable the option to retain the lifecycle state of the Web Service which is imported.
Overwrite existing entities	This option specifies that an existing Web Service with the same uuid in the target CentraSite registry will be overwritten, even if the Web Service in the archive is older than the one in the target CentraSite registry. Enable the option to overwrite the existing Web Service.
Import groups that the user belongs to	This option determines that when you import a user, whether you want to import the groups that the user belongs to. This applies only to user-defined groups. System-defined groups are not imported. Enable the option to import the groups.
Ignore API keys and OAuth2 tokens	This option determines whether the API keys and OAuth2 tokens of the imported assets are to be imported into the target CentraSite registry. Enable the option to ignore the API keys and OAuth2 tokens during the import process.

6. Click **Import** to import the Web Service.

After the import operation completes, the import wizard informs you if the import was successful or if there were any errors and warnings.

7. Click **Download Import Log** to view the import logs.

The import log lists the status of all the objects stating whether they were successfully imported or if there were errors and warnings.

8. Click **OK** to terminate the import wizard.

A Web Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the Web Service asset that you just created is displayed.

Viewing Web Service List

You use the Search Results page to display the list of Web Service assets.

> To view the list of Web Service assets

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Web Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

A list of scopes that are available to you is displayed. By default, CentraSite contains the following predefined scopes:

Scope	Description
Everything	Displays a list of the currently available assets, organizations, and users in CentraSite registry.
Assets	Displays a list of the currently available assets in CentraSite registry. This is the default scope.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and perform the following:
 - a. Click the chevron next to **Assets** option button.
A list of defined asset types in CentraSite is displayed.
 - b. In the displayed list of asset types, select **Service**.
 - c. Click **OK**.

A list of defined Web Service assets is displayed in the Search Results page.

5. To filter the list to see just a subset of the available Web Service assets, type a partial string in the **Keyword** text box. Add the specified keyword to the Search Recipe, by clicking the plus symbol (+) next to the text box, or press **Enter**.

The Search Results page provides the following information about each Web Service asset:

Column	Description
Name	Name of the Web Service asset.
Description	The description for the Web Service.
Asset Type	The asset type, Service .

Column	Description
Last Updated Date	The date on which the Web Service was last modified.
Owner	The user who owns the Web Service.
Organization	The organization which owns the Web Service.
Version	The user-assigned version identifier for the Web Service.

To specify the sorting preference, select an attribute from the **Sort by** list.

Note:

Use the **View** menu to display the additional attributes.

Viewing Web Service Details

You use the Web Service details page to examine the WSDL documentation.

The following general guidelines apply when examining the details of a Web Service asset in CentraSite Business UI:

- If you are not the owner of a Web Service, you cannot view the Web Service details page unless you have a View permission on the Web Service (granted though either a role-based permission or instance-level permission).
- You will only see profiles of the Web Service for which you have an instance-level View permission.

In this task you examine the basic and type-specific attributes that are associated with a Web Service asset. You can view the bindings, operations, WSDL file, associated schema files, and the external links to the WSDL and schema files.

➤ To view the details of a Web Service asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Web Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and perform the following:

- a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.

- b. In the list of asset types, select **Service**.

- c. Click **OK**.

A list of defined Web Service assets is displayed in the Search Results page.

5. Click the Web Service asset you want to examine the attributes.

This opens the Web Service details page. Also, the actions bar displays a set of actions that are available for working with the Web Service.

You can hover over the **info** symbol next to an attribute to display the tooltip text, which describes the purpose of the attribute. The tooltip text displays the values of the attribute's Name and Description fields that are contained in the Service type definition.

6. Examine the generic attributes that are displayed in the **Basic Information** profile.
7. Examine the extended attributes displayed in the individual profiles. Follow these steps:
 - a. Select the profile that contains the attribute(s) you want to display.
 - b. Examine the attributes on the profile as required.
 - c. Repeat steps 7.a and 7.b for each profile for which you want to display the details.

Modifying Web Service Details

You use the details page of a Web Service asset to examine and modify the WSDL specification.

The asset type **Service** has a unique set of profiles. However, an administrator can configure this asset type to display a customized set of profiles and attributes.

The following general guidelines apply when modifying the details of a Web Service asset in CentraSite Business UI:

- If you are not the owner of the Web Service, you cannot examine or modify the details of the Web Service, unless you have the View or Modify permission on the Web Service (granted though either a role-based permission or an instance-level permission).
- When you view the details page of a Web Service, you will only see profiles for which you have the profile-level View permission. You will only be able to modify the attributes of the profiles on which you have the Modify permission.

- When you hover over an attribute, CentraSite displays the tooltip text that provides a short description of the attribute's purpose.
- Some attributes accept only specific types of information. For example, if the Service asset type includes a URL type attribute, you must supply a URL when you modify that attribute. Other attribute types that require a specific type of value include the Date and Email attributes.
- Some attributes are designed to be read-only. You cannot modify the read-only attributes even if you have the CentraSite Administrator role.

In this task you modify the basic and type-specific attributes that are associated with a Web Service. You can view the bindings, operations, WSDL file, associated schema files, and the external links to the WSDL and schema files.

➤ To modify the details of a Web Service asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Web Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and perform the following:
 - a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.
 - b. In the list of asset types, select **Service**.
 - c. Click **OK**.

A list of defined Web Service assets is displayed in the Search Results page.

5. Click the Web Service you want to examine and modify the attributes.

This opens the Web Service details page. Also, the actions bar displays a set of actions that are available for working with the Web Service.

You can hover over the **info** symbol next to an attribute to display the tooltip text, which describes the purpose of the attribute. The tooltip text displays the values of the attribute's Name, and Description fields that are contained in the Service type definition.

6. To modify the generic attributes that are displayed in the **Basic Information** profile, on the actions bar, click **Edit**. Change values of the attributes in the respective data fields as required.
7. To modify the extended attributes that are displayed in the individual profiles, follow these steps:
 - a. Select the profile that contains the attribute(s) you want to modify.
 - b. Change values of the attributes in the respective data fields as necessary.
 - c. Repeat steps 7.a and 7.b for each profile for which you want to modify the attributes.
8. Click **Save**.

Deleting Web Services

If you are not the owner of a Web Service asset, you cannot delete the Web Service unless you have Full permission on the Web Service (granted through either a role-based permission or instance-level permission).

The following general guidelines apply when deleting a Web Service asset in CentraSite Control:

- Deleting a Web Service permanently removes it from the catalog.
 - A Web Service can only be deleted, if it is not the target of an association from another registry object.
 - When you delete a Web Service, CentraSite removes the catalog entry for the Web Service (that is, it removes the instance of the Web Service from CentraSite's object database). Also note that:
 - The performance metrics and event information of the Web Service are also deleted.
- Note:**
When you delete the Web Service, this information is deleted by the built-in action **Delete RuntimeEvents and RuntimeMetrics of Service**, which is installed with CentraSite.
- When you delete a composite Web Service, all of its nonshared components are also deleted.
 - Deleting a Web Service will *not* remove:
 - Other assets to which the Web Service refers (unless the reference is to an asset that is a nonshared component of the Web Service you are deleting). For example, if you are deleting a Web Service which has a Consumes or Consumed By relationship with other services in the registry, the related services will not be deleted.
 - Supporting documents that are attached to the Web Service.
 - Earlier versions of the Web Service. Only the latest version of the Web Service can be deleted; to remove earlier versions, they must be purged.

- You cannot delete a Web Service if:
 - The Web Service is in a pending state (for example, awaiting approval).
 - Any user in your CentraSite registry is currently modifying the Web Service.
 - The Web Service is associated with access tokens such as API key or OAuth token.

Note:

In such cases, you can delete the web service, only when you revoke and then delete the access tokens associated with it. For more information on revoking the access tokens, refer to ["Revoking Access Tokens as API Consumer" on page 1449](#) or ["Revoking Access Tokens as API Provider" on page 1448](#). Similarly, for deleting the access tokens, refer to ["Deleting Access Tokens" on page 1450](#).

> To delete Web Service assets

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Web Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and perform the following:
 - a. Click the chevron next to **Assets** option button.
A list of defined asset types in CentraSite is displayed.
 - b. In the displayed list of asset types, select **Service**.
 - c. Click **OK**.

A list of defined Web Service is displayed in the Search Results page.

5. In the displayed list of Web Service assets, select one or multiple assets you want to delete.
6. On the actions bar of the Search Results page, click **Delete**.

You can also delete a single Web Service asset from the Actions bar of that particular Web Service Details page.

7. When you are prompted to confirm the delete operation, click **Yes**.

Note:

If you have selected a set of Web Services, where one or multiple Web Services are in a pending state (for example, awaiting approval), CentraSite ignores the pending list of Web Services, and deletes any remaining Web Services for which you have the required permission.

Revoking Access Tokens as API Consumer

An API Consumer will use the User Preferences page to revoke an access token.

After issuing an access token, you might want to revoke the token if you find a serious error in the virtual instance of an asset.

When you revoke an access token, access to the associated virtual asset, and its resources is blocked when you try to access them using that particular access token.

- You have configured the API key authentication or OAuth 2.0 token authentication using the **API Consumption Settings** action in the details page of the asset.
- A gateway instance (for example, Mediator) is up and running.

> To revoke access token as an API Consumer

1. Open a web browser and navigate to the CentraSite Business UI.
2. In CentraSite Business UI, click your user name that is located in the upper-right corner of header area.

This opens the User Preferences page.

3. Locate the section **My Access Keys**.
4. In the displayed list of access tokens, hover over the access token you want to revoke.

CentraSite displays one or more actions you can perform on the access token.

5. Click the **Delete** icon.

A confirmation message appears that the access token is revoked from the CentraSite Registry Repository.

6. Click **Yes** in the confirmation dialog box.

Once the access token revocation is processed, CentraSite sends an email message to the API Consumer informing that the request has been processed successfully.

CentraSite provides predefined email template for the access token revocation. By default, this template is configured in the **centrasite.xml** file. But, if you do not want to use the predefined email template, you can create and add your own email template to CentraSite, and configure the **centrasite.xml** file, as required.

Revoking Access Tokens as API Consumer

An API Consumer will use the User Preferences page to revoke an access token.

After issuing an access token, you might want to revoke the token if you find a serious error in the virtual instance of an asset.

When you revoke an access token, access to the associated virtual asset, and its resources is blocked when you try to access them using that particular access token.

- You have configured the API key authentication or OAuth 2.0 token authentication using the **API Consumption Settings** action in the details page of the asset.
- A gateway instance (for example, Mediator) is up and running.

➤ To revoke access token as an API Consumer

1. Open a web browser and navigate to the CentraSite Business UI.
2. In CentraSite Business UI, click your user name that is located in the upper-right corner of header area.

This opens the User Preferences page.

3. Locate the section **My Access Keys**.
4. In the displayed list of access tokens, hover over the access token you want to revoke.

CentraSite displays one or more actions you can perform on the access token.

5. Click the **Delete** icon.

A confirmation message appears that the access token is revoked from the CentraSite Registry Repository.

6. Click **Yes** in the confirmation dialog box.

Once the access token revocation is processed, CentraSite sends an email message to the API Consumer informing that the request has been processed successfully.

CentraSite provides predefined email template for the access token revocation. By default, this template is configured in the **centrasite.xml** file. But, if you do not want to use the predefined email template, you can create and add your own email template to CentraSite, and configure the **centrasite.xml** file, as required.

REST Service Management

This section describes operations you can perform to manage REST services through CentraSite Business UI.

About REST Service Assets

CentraSite's REST framework enables you to model APIs conforming to the (Resource Oriented Architecture) ROA design. For example, you might model an API that serves to expose the web service data and functionality as a collection of resources. Each resource will be accessible with unique Uniform Resource Identifiers (URLs). In your API, you expose a set of HTTP operations (methods) to perform on a specific resource and capture the request and response messages and status codes that will be unique to the HTTP method and linked within the specific resource of the API.

The basic elements of a REST API in the CentraSite registry are as follows:

- The API itself (For example, *phonestore*)
- Its resource (*phones*), available on the unique base URL (*/phones*)
- The defined HTTP method (*GET*) for accessing the resource (*phones*)
- Parameters for request representations (*412456*)
- A request generated for this method (*Request 123*)
- A response with the status code received for this request (*Response ABCD*)

Instructions throughout this guide use the term *REST API* when referring to REST Service assets.

About REST Applications (APIs)

In CentraSite, you document a REST API as an asset instance of the type REST Service or Virtual REST Service.

These APIs are typically collections of resources.

For example, consider an API that is defined to support an online phone store application. Assume, this sample Phone Store API currently has a database that defines the various brands of phones, features in the individual phones, and the inventory of each phone.

The Phone Store API is used as a sample to illustrate how to model URL patterns for resources, resource methods, HTTP headers and response codes, content types, and parameters for request representations to resources.

Base URL

The base URL of an API is constructed by the domain, port, and context mappings of the API. For example, if the server name is `www.phonestore.com`, port is `8080`, and the API context is `api`. The full Base URL is:

```
http://www.phonestore.com:8080/api
```

REST (API) Parameters

Parameters defined at the higher API level are inherited by all Resources and by all Methods included in the individual Resources.

CentraSite permits different types of parameters at the API level, as described later in this topic.

REST Resources

Resources are the basic components of an API. Examples of resources from an online Phone Store API include a phone, an order from a store, and a collection of customers.

After you identify a service to expose as an API, you define the resources for the API. CentraSite's flexible metadata store captures the relationships of APIs with resources and ensures that the APIs are available in the right way.

For example, consider the case of an online Phone Store API. In this example, there are a number of ways to represent the data in the phone store database as an API. The verbs in the HTTP request maps to the operations that the database supports, such as select, create, update, delete.

Each resource needs to be addressable by a unique URI. Along with the URI you're going to expose for each resource, you also need to decide what can be done to each resource. The HTTP methods passed as part of an HTTP request header direct the API what needs to be done with the addressed resource.

Resource URLs

An URL identifies the location of a specific resource.

For example, consider the case of our sample Phone Store API designed to support an online phone store application. The resources will have the following URLs:

URL	Description
http://www.phonestore.com/api/phones	Specifies the collection of phones contained in the online store.
http://www.phonestore.com/api/phones/412456	Accesses a phone referenced by the product code 412456.
http://www.phonestore.com/api/phones/412456/reviews	Specifies a set of reviews posted for a phone of code 412456.
http://www.phonestore.com/api/phones/412456/reviews/78	Accesses a specific review referenced by the unique ID 78 contained in the reviews of the phone of code 412456.

CentraSite supports the following patterns of resource URL: a collection of resources or a particular resource.

For example, consider the above example of an online Phone Store API.

Collection URL: <http://phonestore.com/api/phones>

Unique URL: <http://phonestore.com/api/phones/412456/features>

to retrieve a collection resource describing the key features of phone whose product code is 412456.

Resource Parameters

Parameters defined at the higher Resource level are inherited by all Methods in the particular resource; it does not affect the API.

CentraSite permits different types of parameters at the Resource level, as described later in this topic.

Resource Methods

Individual resources can define their capabilities using supported HTTP methods. To invoke an API, the client would call an HTTP operation on the URL associated with the API's resource. For example, to retrieve the key feature information for phone whose product code is 412456, the client would make a service call HTTP GET on the following URL:

```
http://www.phonestore.com/phones/412456/features
```

Supported HTTP Methods

CentraSite supports the standard HTTP methods for modeling APIs: GET, POST, PUT, DELETE, HEAD, OPTIONS, PATCH, TRACE, and CONNECT.

Important:

During virtualization of a REST Service, CentraSite does not support the following HTTP methods: HEAD, OPTIONS, TRACE, and CONNECT.

This is because, when a Virtual REST Service is published to Mediator gateway, CentraSite only supports the HTTP methods: GET, POST, PUT, PATCH, and DELETE, at run-time.

The following table describes the semantics of HTTP methods for our sample Phone Store API:

Resource URI	Supported HTTP Methods	Description
/phones	GET	List all phones.
/phones	POST	Creates a new phone with product code 412456.
/phones/412456	GET	Retrieves details of a phone whose product code is 412456.
/phones/412456	DELETE	Removes a phone whose product code is 412456.
/phones/412456?fields=(make, features, bodytype)	GET	Retrieves additional details (such as Brand, Features, Body Type) of a phone whose product code is 412456.
/phones/412456/make	GET	Identifies the brand of a phone whose product code is 412456.

Resource URI	Supported HTTP Methods	Description
/phones/search?q=(make,eq,apple)	GET	Retrieves a list of all phones whose brand is Apple.
/phones/412456?make=apple&features=3g	PUT	Updates a phone whose product code is 412456, brand is Apple, and also 3G compatible.
/phones/412456/	PATCH	Partial update for a specific phone.

For information on the HTTP methods that CentraSite ships, in CentraSite Control, go to **Administration > Taxonomies**. On the Taxonomies page, enable the **Show all Taxonomies** option. Navigate to **HTTP Methods** in the list of taxonomies.

Method Parameters

Parameters defined at the lower Method level apply only to that particular method; it does not affect either the API or the Resource.

CentraSite permits different types of parameters at the Method level, as described later in this topic.

REST Parameters

Parameters specify additional information to a request. You use parameters as part of the URL or in the headers or as components of a message body.

Parameter Levels

A parameter can be set at different levels of an API. When you document a REST API in CentraSite, you define parameters at the API level, Resource level, or Method level to address the following scenarios:

- If you have the parameter applicable to all resources in the API, then you define this parameter at the API level. This indirectly implies that the parameter is propagated to all resources and methods under the particular API.
- If you have the parameter applicable to all methods in the API, then you define this parameter at the Resource level. This indirectly implies that the parameter is propagated to all methods under the particular resource.
- If you have the parameter applicable only to a method in the API, then you define this parameter at the Method level.

API-Level Parameters

Setting parameters at the API level enables the automatic assignment of the parameters to all resources and methods included in the API. Any parameter value you specify at the higher API

level overrides the parameter value you set at the lower Resource level or the lower Method level if the parameter names are the same.

For example, say you have a header parameter called API Key that is used for consuming an API.

```
x-CentraSite-APIKey:a4b5d569-2450-11e3-b3fc-b5a70ab4288a
```

This parameter is specific to the entire API and to the individual components - resources and methods directly below the API. Such a parameter can be defined as a parameter at the API level.

At an API level, CentraSite allows you to define the following types of parameters:

- Query-String parameter
- Header parameter
- Form parameter

Resource-Level Parameters

Setting parameters at the Resource level enables the automatic assignment of the parameters to all methods within the resource. Any parameter value you specify at the higher Resource level overrides the parameter value you set at the lower Method level if the parameter names are the same. In contrast, the lower Resource level parameters will not affect the higher API level parameters.

Consider our sample Phone Store API maintains a database of reviews about different phones. Here is a request to display information about a particular user review, 78 of the phone whose product code is 412456.

```
GET /phones/412456/user_reviews/78
```

In the example, `/user_reviews/78` parameter narrows the focus of a GET request to review `/78` within a particular resource `/412456`.

This parameter is specific to the particular resource phone whose product code is 412456 and to any individual methods that are directly below the particular resource. Such a parameter can be defined as a parameter at the Resource level.

At a Resource level, CentraSite allows you to define the following types of parameters:

- Path parameter
- Query-String parameter
- Header parameter
- Form parameter

Method-Level Parameters

If you do not set parameters at the API level or Resource level, you can set them at a Method level. Parameters you set at the Method level are used for the HTTP method execution. They are useful to restrict the response data returned for a HTTP request. Any parameter value you specify at the lower Method level is overridden by the value set at higher API level parameter or the higher

Resource level parameter if the names are the same. In contrast, the lower Method level parameters will not affect the higher API level or Resource level parameters.

For example, the Phone Store API described might have a request to display information contributed by user Allen in 2013 about a phone whose product code is 412456.

```
GET /phones/412456/user_reviews/78?year=2013&name=Allen
```

In this example, `year=2013` and `name=Allen` narrow the focus of the GET request to entries that user Allen added to user review 78 in 2013.

At a Method level, CentraSite allows you to define the following types of parameters:

- Query-String parameter
- Header parameter
- Form parameter

Parameter Types

CentraSite supports four types of parameters in REST API: Query-String, Path, Header, and Form.

Let's have a look at different parameter types to see how they can be used for parameterizing the resources.

Query-String Parameters

Query-String parameters are appended to the URI after a `?` with name-value pairs. The name-value pairs sequence is separated by either a semicolon or an ampersand.

For instance, if the URL is `http://phonestore.com/api/phones?itemID=itemIDValue`, the query parameter name is `itemID` and value is the `itemIDValue`. Query parameters are often used when filtering or paging through HTTP GET requests.

Now, consider the online Phone Store API. A customer, when trying to fetch a collection of phones, may wish to add options, such as, `android v4.3 OS` and `8MP camera`. The URI for this resource could look like this:

```
/phones?features=androidosv4.3&cameraresolution=8MP
```

Path Parameters

Path parameters are defined as part of the resource URI. For example, the URI can include `phones/item`, where `/item` is a path parameter that identifies the item in the collection of resource `/phones`. Because path parameters are part of the URI, they are essential in identifying the request.

Now, consider the above online Phone Store API example. A customer may wish to fetch details about a phone `{phone-id}` whose product code is "412456". The URI for this resource could look like this:

```
/phones/412456
```

Important:

As a best practice, we recommend that you adopt the following conventions when specifying a path parameter in the resource URI:

- Append a path parameter variable within curly {} brackets.
- Specify a path parameter variable such that it exactly matches the path parameter defined at the Resource level.

Header Parameters

Header parameters are HTTP headers. Headers often contain metadata information for the client, or server.

```
x-CentraSite-APIKey:a4b5d569-2450-11e3-b3fc-b5a70ab4288a
```

You can create custom headers, as needed. As a best practice, we recommend that you prefix the header name with x-.

HTTP/1.1 defines the headers that can appear in a HTTP response in three sections of RFC 2616: 4.5, 6.2, and 7.1. Examine these codes to determine which are appropriate for the API.

Form Parameters

Form parameters and values are encoded in the request message body, in the format specified by the content type (application/x-www-form-urlencoded).

```
features=androidosv4.3&cameraresolution=8MP
```

Important: Although CentraSite allows you to define parameters of the type Form at the API level, these parameters are not supported at run-time.

Parameter Data Types

When you add a parameter to the API, you specify the parameter's data type. The data type determines what kind of information the parameter can hold.

CentraSite supports the following data types for parameters:

Data Type	Description
String	Specifies a string of text.
URL	<p>Holds a URL/URI. This type of parameter only accepts values in the form:</p> <pre>protocol://host:port/path</pre> <p>Where:</p> <ul style="list-style-type: none"> ■ <i>protocol</i> is any protocol that java.net.URL supports. ■ <i>host</i> is the name or IP address of a host machine. ■ <i>port</i> is the port on which the host machine is listening. ■ <i>path</i> (optional) is the path to the requested resource on the specified host.
Boolean	Specifies a true or false value.

Data Type	Description
Email	Specifies an email address. This data type only accepts values in the format: <code>anyString@anyString</code>
Number	Specifies a numeric value.
Duration	Specifies a value that represents a period of time as expressed in years, months, days, hours, minutes, and seconds. This duration is specified using an <code>xs:duration</code> format. It specifies a duration in terms of years (either 0 or 1), months, days, hours, minutes, and seconds.
Date/Time	Specifies a timestamp that represents a specific date and/or time. The date/time input parameters allow year, month, and day input as well as hour and minute. Hour and minute default to 0. This data type only accepts date values in the format <code>yyyy-mm-dd</code> ; and time values in the format <code>hh:mm:ss</code> .
IP Address	Specifies a numeric IP address in the v4 or v6 format.

Supported Content Types

Clients can optionally specify the content-type format they want the response to use.

CentraSite includes a set of predefined content types that are classified in the following taxonomy categories:

Predefined Taxonomy Category	Description
Applications	An application content-type value to transmit API data or binary data and among other uses to implement an electronic mail file transfer service. Example: - <code>application/xml</code> - <code>application/json</code>
Audio Files	An audio content-type value for transmitting audio or voice data. Example: - <code>audio/basic</code> - <code>audio/mp4</code>

Predefined Taxonomy Category	Description
Image Files	<p>An image content-type value, for transmitting still image (picture) data.</p> <p>Example:</p> <ul style="list-style-type: none"> - image/gif - image/png
Text Files	<p>A text content-type value to represent textual information in a number of character sets and formatted text description languages in a standardized manner.</p> <p>Example:</p> <ul style="list-style-type: none"> - text/html - text/plain
Video Files	<p>A video content-type value for transmitting video or moving image data, possibly with audio as part of the composite video data format.</p> <p>Example:</p> <ul style="list-style-type: none"> - video/mpeg

Supported HTTP Status Codes

An API response returns a HTTP status code that indicates success or failure of the requested operation.

CentraSite allows you specify HTTP codes for each method to help clients understand the response. While responses can contain an error code in XML or other format, clients can quickly and more easily understand an HTTP response status code. The HTTP specification defines several status codes that are typically understood by clients.

CentraSite includes a set of predefined response codes that are classified in the following categories:

Predefined Response Code Categories	Description
1xx	Informational.
2xx	Success.
3xx	Redirection. Need further action.
4xx	Client error. Correct the request data and retry.
5xx	Server error.

For information on the status codes that CentraSite supports out-of-the-box, in CentraSite Control, go to **Administration > Taxonomies**.

On the Taxonomies page, enable the **Show all Taxonomies** option. Navigate to **HTTP Status Codes** in the list of taxonomies.

HTTP/1.1 defines all the legal status codes. Examine these codes to determine which are appropriate for your API.

Consider the case of online Phone Store API. The following table describes the HTTP status codes that each of the URIs and HTTP methods combinations will respond:

Resource URI	Supported HTTP Methods	Supported HTTP Status Codes
/phones/orders	GET	200 (OK, Success)
/phones/orders	POST	201 (Created) if the Order resource is successfully created, in addition to a Location header that contains the link to the newly created Order resource; 406 (Not Acceptable) if the format of the incoming data for the new resource is not valid
/phones/orders/{order-id}	GET	200 (OK); 404 (Not Found) if Order Resource not found
/phones/orders/{order-id}	DELETE	204 (OK); 404 (Not Found) if Order Resource not found
/phones/orders/{order-id}/status	GET	200 (OK); 404 (Not Found) if Order Resource not found
/phones/orders/{order-id}/paymentdetails	GET	200 (OK); 404 (Not Found) if Order Resource not found
/phones/orders/{order-id}/paymentdetails	PUT	201 (Created); 406 (Not Acceptable) if there is a problem with the format of the incoming data on the new payment details; 404 (Not Found) if Order Resource not found
/phones/orders/{order-id}/paymentdetails	PATCH	200 (ok); 404 (Not Found) if Order Resource not found

Sample Requests and Responses

To illustrate the usage of an API, you provide sample request and response messages. Consider the sample Phone Store API that maintains a database of phones in different brands. The Phone Store API might provide the following examples to illustrate its usage:

Sample 1 - Retrieve a list of phones

Client Request

```
GET /phones HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.api.phonestore.com
Accept-Language: en-us
Accept-Encoding: text/xml
Connection: Keep-Alive
```

Server Response

```
HTTP/1.1 200 OK
Date: Mon, 14 July 11:53:27 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 18 June 2014 09:18:16 GMT
Content-Length: 356
Content-Type: text/xml
<phones>
  <phone>
    <phone-id>412456</phone-id>
    <name>Asha</name>
    <brand>Nokia</brand>
    <price currency="irs">11499</price>
    <features>
      <camera>
        <back>3</back>
      </camera>
      <memory>
        <storage scale="gb">8</storage>
        <ram scale="gb">1</ram>
      </memory>
      <network>
        <gsm>850/900/1800/1900 MHz</gsm>
      </network>
    </features>
  </phone>
  <phone>
    <phone-id>412457</phone-id>
    <name>Nexus7</name>
    <brand>Google</brand>
    <price currency="irs">16499</price>
    <features>
      <camera>
        <front>1.3</front>
        <back>5</back>
      </camera>
      <memory>
        <storage scale="gb">16</storage>
        <ram scale="gb">2</ram>
      </memory>
      <network>
        <gsm>850/900/1800/1900 MHz</gsm>
        <HSPA>850/900/1900 MHz</HSPA>
      </network>
    </features>
  </phone>
</phones>
```

Sample 2 - Find a phone that does not exist

Client Request

```
GET /phone/4156 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.api.phonestore.com
Accept-Language: en-us
Accept-Encoding: text/xml
Connection: Keep-Alive
```

Server Response

```
HTTP/1.1 404 Not Found
Accept: application/xml
Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Server: Apache/2.2.14 (Win32)
```

Sample 3 - Create a phone

Client Request

```
POST /phones
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.api.phonestore.com
Accept-Language: en-us
Accept-Encoding: text/xml
Content-Length: 156
Connection: Keep-Alive
<phone>
  <name>iPhone5</name>
  <brand>Apple</brand>
  <price currency="irs">24500</price>
  <features>
    <camera>
      <front>1.2</front>
      <back>8</back>
    </camera>
    <memory>
      <storage scale="gb">32</storage>
      <ram scale="gb">2</ram>
    </memory>
    <network>
      <gsm>850/900/1800/1900 MHz</gsm>
      <HSPA>850/900/1900 MHz</HSPA>
    </network>
  </features>
</phone>
```

Server Response

```
HTTP/1.1 201 OK
Date: Mon, 14 July 11:53:27 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 18 June 2014 09:18:16 GMT
Content-Type: text/xml
Content-Length: 15
<phone>
  <phone-id>2122</phone-id>
  <name>iPhone5</name>
  <brand>Apple</brand>
```

```

<price currency="irs">24500</price>
<features>
  <camera>
    <front>1.2</front>
    <back>8</back>
  </camera>
  <memory>
    <storage scale="gb">32</storage>
    <ram scale="gb">2</ram>
  </memory>
  <network>
    <gsm>850/900/1800/1900 MHz</gsm>
    <HSPA>850/900/1900 MHz</HSPA>
  </network>
</features>
</phone>

```

Sample 4 - Adjusting the phone name

Client Request

```

PATCH /phones/2122
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.api.phonestore.com
Accept-Language: en-us
Accept-Encoding: text/xml
Content-Length: 156
Connection: Keep-Alive
<phone>
  <name>iPhone6</name>
  <brand>Apple</brand>
  <price currency="irs">24500</price>
  <features>
    <camera>
      <front>1.2</front>
      <back>8</back>
    </camera>
    <memory>
      <storage scale="gb">32</storage>
      <ram scale="gb">2</ram>
    </memory>
    <network>
      <gsm>850/900/1800/1900 MHz</gsm>
      <HSPA>850/900/1900 MHz</HSPA>
    </network>
  </features>
</phone>

```

Server Response

```

HTTP/1.1 200 OK
Date: Mon, 14 July 11:53:27 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 18 June 2014 09:18:16 GMT
Content-Type: text/xml
Content-Length: 374
<id>2122</id>
<phone>
  <phone-id>2111</phone-id>
  <name>iPhone6</name>

```

```
<brand>Apple</brand>
<price currency="irs">24500</price>
<features>
  <camera>
    <front>1.2</front>
    <back>8</back>
  </camera>
  <memory>
    <storage scale="gb">32</storage>
    <ram scale="gb">2</ram>
  </memory>
  <network>
    <gsm>850/900/1800/1900 MHz</gsm>
    <HSPA>850/900/1900 MHz</HSPA>
  </network>
</features>
<phone>
```

REST Service Compatibility

Beginning with version 9.7, CentraSite supports the enhanced interface for REST Services (in contrast, earlier versions of CentraSite supported a standardized interface for REST Service). Documentation of the prior REST service interface is available to CentraSite customers who have a current maintenance contract in Empower Product Support website.

- **If you Migrate REST Services from a Pre-9.7 Release:** If you have REST Services that were created prior to CentraSite version 9.7, these REST Services will continue to hold the version's metadata in the enhanced REST Service interface implemented by current version of CentraSite.
- **If you Migrate REST Services from a 9.7 Release:** If you have REST Services that were created in the CentraSite version 9.7 using the CentraSite Business UI, these REST Services will again continue to hold the version's metadata in the enhanced REST Service interface implemented by current version of CentraSite. However, you will find the following information in the migrated REST Service:
 - The sample request and response messages that existed under the REST Method display without changes.
 - The status codes that existed under the REST Method will now display under the REST Response.

Note:

Beginning with version 9.8, although CentraSite supports the existing REST Sample Requests and Responses, we enforce you use the REST Requests and REST Responses to specify additional details about the REST payload. You may use the Sample Requests and Responses if required.

Adding REST Service using Importer

Pre-requisites:

To add a REST Service asset to an organization's asset catalog, you must belong to a role that has the Create Assets or Manage Assets permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

An importer in CentraSite is a utility that takes in the specification file of a particular format as the input and then creates an asset of the particular metadata format in the registry. For example, the CentraSite importer for REST Service reads a RAML specification and from it, creates a REST Service asset that best describes the RESTful Service with RAML specification. The importer also uploads the input file to the CentraSite repository and links the file to the REST Service. When you import a REST Service using a RAML file, for example, the importer copies the RAML file into the repository and then links the file to the REST Service.

CentraSite supports the RAML and Swagger REST metadata formats. The following table also identifies the types of files they require as input:

To import this format of REST Service...	You must supply this type of file...
RAML	RESTful Service Modeling Language (RAML) file.
Swagger	JavaScript Object Notation (JSON) file.
	- OR -
	Yet Another Markup Language (YAML) file.

Importing REST Service using a RAML File

Pre-requisites:

To add a REST Service asset to an organization's asset catalog, you must belong to a role that has the Create Assets or Manage Assets permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

Before you import a REST Service asset to the catalog, you must have the RAML file that you want to import. This file can reside on the file system of the computer where your browser is running or it can reside anywhere on the network, as long as its location is addressable through a URL.

The CentraSite importer for RAML copies a RAML file to the repository, and creates a REST Service asset that best describes the RESTful Service representing the RAML specification.

The registry objects are as follows:

- A REST Service asset (REST Service) with RAML specification. This RAML specification is stored as a file attribute in the **Documents** field of the **Specification** profile.
- A REST endpoint URI object that refers to the REST Service.
- A REST Resource object that refers to the REST Service.

- A REST Method object that refers to the REST Service.
- A REST Parameter object that is referred to by the one of the above objects.
- A REST Request Content-Type (REST Payload) object that is referred to by the REST Service or REST Method object.
- A REST Response Content-Type (REST Payload) object that is referred to by the REST Service or REST Method object.
- A REST Status Code object that is referred to by the Response Content-Type object.

➤ **To import a REST Service asset using RAML specification**

1. In the CentraSite Business UI activity bar, click **Create Asset**.

This opens the **Create New Asset** wizard.

2. In the **Basic Information** profile, provide the required information for each of the displayed data fields.

Field	Description
-------	-------------

Name	(Optional). Name of the REST Service.
-------------	---------------------------------------

Note:

This is the name that users will see when they view this REST Service asset in the CentraSite User Interfaces. Therefore, specify a meaningful name for each REST Service.

The name of a REST Service asset must be NCName-conformant, meaning that:

- The name must begin with a letter or the underscore character (_).
- The remainder of the name can contain any combination of letters, digits, or the following characters: . - _ (that is, period, dash, or underscore). It can also contain combining characters and extender characters (for example, diacriticals).
- The name cannot contain any spaces.
- Furthermore, if the REST Service name contains any non-conformant character, upon publishing the REST Service to any gateway, the non-conformant character is simply replaced with the underscore character (_) in Mediator. However, in CentraSite the REST Service name defined by you is displayed.

For more information about the NCName type, see <http://www.w3.org/TR/xmlschema-2/#NCName>

Field	Description						
	If you have not specified a Name for the REST Service, CentraSite automatically maps this field with the <code>Service Title</code> property, that is defined in the RAML specification.						
Type	The asset type, REST Service .						
Organization	The organization to which you want to add the REST Service. (The Organization list only displays organizations for which you have the Manage Assets permission or at least the Create Assets permission.)						
Version	(Optional). The version identifier for the REST Service. You can enter any string in this field, that is, the version identifier does not need to be numeric. You can also leave the field blank. You can later create new versions of the REST Service. The default is 1.0. Examples: <pre>0.0a 1.0.0 (beta) Pre-release 001 V1-2007.04.30</pre>						
	If you have not specified a Version for the REST Service, CentraSite automatically maps this field with the <code>Service Version</code> property, which is defined in the RAML specification.						
Description	(Optional). The description for the REST Service. Note: This is the description information that users will see when they view this REST Service asset in the CentraSite User Interfaces. Therefore, specify a meaningful description for each REST Service. If you have not specified a Description for the REST Service, CentraSite automatically maps this field with the <code>Description</code> property, that is defined in the RAML specification.						
Import From Specification File	Select the specification file type, RAML-0.8 .						
	<table border="1"> <thead> <tr> <th>Option</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>URL</td> <td>If the RAML file you are importing resides on the network, you can specify its URL.</td> </tr> <tr> <td>File</td> <td>If the specification resides in your local file system, specify the file name. You can use the Browse button to navigate to the required folder.</td> </tr> </tbody> </table>	Option	Description	URL	If the RAML file you are importing resides on the network, you can specify its URL.	File	If the specification resides in your local file system, specify the file name. You can use the Browse button to navigate to the required folder.
Option	Description						
URL	If the RAML file you are importing resides on the network, you can specify its URL.						
File	If the specification resides in your local file system, specify the file name. You can use the Browse button to navigate to the required folder.						

- Click the **Advanced Settings** chevron to expand the additional options that are available for the Service asset. Provide the required information for each of the displayed fields:

Field	Description
Credentials	If you have specified a URL and the site you want to access through the URL requires user authentication, type a username and password for authentication at the URL site.
Resolution	Select a resolution strategy, which will allow you to specify how an already existing imported and included file is handled. For each of the imported and included files you have one of these options: <ul style="list-style-type: none">■ Overwrite the importing file with new content.■ Create a new version of the file with the new content (if, for example, you want to modify a RAML file but want to retain its previous version).

Note:

Currently, CentraSite does not support this option for a REST Service with RAML specification.

4. Click **Next**.

You cannot navigate to the next screen unless all of its required attributes have been set.

5. In the **Preview** panel, review the basic information for the REST Service before you actually add to the CentraSite registry.
6. Click **Save**.

A REST Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the REST Service asset that you just created is displayed.

7. Configure the extended attributes of the REST Service as described later in this topic.

Tip:

If you had previously imported a RAML file that has an associated schema file and you now re-import just the schema file with modifications, your browser may not display the updated contents of the schema file. This can happen if the browser cache is not being updated automatically. To rectify the problem, you can change your browser settings so that pages are always updated on every visit.

Importing REST Service using a Swagger File

Pre-requisites:

To add a REST Service asset to an organization's asset catalog, you must belong to a role that has the Create Assets or Manage Assets permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

Before you import a REST Service asset to the catalog, you must have the Swagger file that you want to import. This file can reside on the file system of the computer where your browser is running or it can reside anywhere on the network, as long as its location is addressable through a URL.

The CentraSite importer for Swagger copies a Swagger file to the repository and creates a REST Service asset that best describes the RESTful Service representing the Swagger specification.

The registry objects are as follows:

- A REST Service asset (REST Service) with Swagger specification. This Swagger specification is stored as a file attribute in the **Documents** field of the **Specification** profile.
- A REST endpoint URI object that refers to the REST Service.
- A REST Resource object that refers to the REST Service.
- A REST Method object that refers to the REST Service.
- A REST Parameter object that is referred to by the one of the above objects.
- A REST Request Content-Type object that is referred to by the REST Service or REST Method object.
- A REST Response Content-Type object that is referred to by the REST Service or REST Method object.
- A REST Status Code object that is referred to by the Response Content-Type object.

➤ **To import a REST Service asset using Swagger specification**

1. In the CentraSite Business UI activity bar, click **Create Asset**.

This opens the **Create New Asset** wizard.

2. In the **Basic Information** profile, provide the required information for each of the displayed data fields.

Field	Description
Name	(Optional). Name of the REST Service.

Note:

This is the name that users will see when they view this REST Service asset in the CentraSite User Interfaces. Therefore, specify a meaningful name for each REST Service.

Field	Description
	<p>The name of a REST Service asset must be NCName-conformant, meaning that:</p> <ul style="list-style-type: none"> ■ The name must begin with a letter or the underscore character (<code>_</code>). ■ The remainder of the name can contain any combination of letters, digits, or the following characters: <code>. - _</code> (that is, period, dash, or underscore). It can also contain combining characters and extender characters (for example, diacriticals). ■ The name cannot contain any spaces. ■ Furthermore, if the REST Service name contains any non-conformant character, upon publishing the REST Service to any gateway, the non-conformant character is simply replaced with the underscore character (<code>_</code>) in Mediator. However, in ContraSite the REST Service name defined by you is displayed. <p>For more information about the NCName type, see http://www.w3.org/TR/xmlschema-2/#NCName</p> <p>If you have not specified a Name for the REST Service, ContraSite automatically maps this field with the <code>Service Title</code> property, that is defined in the Swagger specification.</p>
Type	The asset type, REST Service .
Organization	The organization to which you want to add the REST Service. (The Organization list only displays organizations for which you have the Manage Assets permission or at least the Create Assets permission.)
Version	<p>(Optional). The version identifier for the REST Service. You can enter any string in this field, that is, the version identifier does not need to be numeric. You can also leave the field blank. You can later create new versions of the REST Service. The default is <code>1.0</code>.</p> <p>Examples:</p> <pre data-bbox="396 1457 1365 1583">0.0a 1.0.0 (beta) Pre-release 001 V1-2007.04.30</pre> <p>If you have not specified a Version for the REST Service, ContraSite automatically maps this field with the <code>Service Version</code> property, which is defined in the Swagger specification.</p>
Description	(Optional). The description for the REST Service.
	<p>Note:</p>

Field	Description						
	<p>This is the description information that users will see when they view this REST Service asset in the CentraSite User Interfaces. Therefore, specify a meaningful description for each REST Service.</p> <p>If you have not specified a Description for the REST Service, CentraSite automatically maps this field with the <code>Description</code> property, that is defined in the Swagger specification.</p>						
Import From Specification File	<p>Select the specification file type, Swagger-2.0.</p> <table border="1"> <thead> <tr> <th>Option</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>URL</td> <td>If the Swagger file you are importing resides on the network, you can specify its URL.</td> </tr> <tr> <td>File</td> <td>If the specification resides in your local file system, specify the file name. You can use the Browse button to navigate to the required folder.</td> </tr> </tbody> </table>	Option	Description	URL	If the Swagger file you are importing resides on the network, you can specify its URL.	File	If the specification resides in your local file system, specify the file name. You can use the Browse button to navigate to the required folder.
Option	Description						
URL	If the Swagger file you are importing resides on the network, you can specify its URL.						
File	If the specification resides in your local file system, specify the file name. You can use the Browse button to navigate to the required folder.						

- Click the **Advanced Settings** chevron to expand the additional options that are available for the Service asset. Provide the required information for each of the displayed fields:

Field	Description
Credentials	If you have specified a URL and the site you want to access through the URL requires user authentication, type a username and password for authentication at the URL site.
Resolution	<p>Select a resolution strategy, which will allow you to specify how an already existing imported and included file is handled. For each of the imported and included files you have one of these options:</p> <ul style="list-style-type: none"> ■ Overwrite the importing file with new content. ■ Create a new version of the file with the new content (if, for example, you want to modify a Swagger file but want to retain its previous version). <p>Note: Currently, CentraSite does not support this option for a REST Service with Swagger specification.</p>

- Click **Next**.

You cannot navigate to the next screen unless all of its required attributes have been set.

- In the **Preview** panel, review the basic information for the REST Service before you actually add to the CentraSite registry.

6. Click **Save**.

A REST Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the REST Service asset that you just created is displayed.

7. Configure the extended attributes of the REST Service as described later in this topic.

Tip:

If you had previously imported a Swagger file that has an associated schema file and you now re-import just the schema file with modifications, your browser may not display the updated contents of the schema file. This can happen if the browser cache is not being updated automatically. To rectify the problem, you can change your browser settings so that pages are always updated on every visit.

Adding REST Service using an Archive File

Pre-requisites:

To add a REST Service asset to an organization's asset catalog, you must belong to a role that has the Create Assets or Manage Assets permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

Before you import a REST Service asset to the catalog, you must have the archive file (.zip file). This file must reside on the file system of the computer where your browser is running.

You can import a REST Service asset using the archive file (.zip file) to which the REST Service was previously exported. You can import REST Services into the same CentraSite registry from which they were originally exported or to a different CentraSite registry.

➤ **To add a REST Service asset using an archived asset**

1. In the CentraSite Business UI activity bar, click **Create Asset**.

This opens the **Create New Asset** wizard. The bottom panel of the **Create New Asset** wizard displays the option to import an archive file.

2. To import an archive file, click **Choose** to navigate to the folder where the exported archive file of a REST Service asset resides, and select the file.

When you choose a file to import, the fields in the area labeled **Basic Information** cannot be modified.

3. Click **Next**.

The **Create New Asset** wizard displays a list of the referenced objects for the REST Service to import.

4. The check box next to each referenced object indicates whether the object should be imported. By default, all objects displayed are included in the import set. To exclude any referenced object from the import set, clear the corresponding check box.
5. Click the **Advanced Settings** chevron to expand the additional options that are available for the REST Service asset. Provide values for each of the displayed options:

Option	Description
Change Owner	<p>The imported REST Service can be assigned to the same owner as in the source CentraSite registry, or you can assign a new owner.</p> <p>The Change Owner field is type-ahead field. As you type characters in this field, the dialog box lists the user names that match the characters you specify.</p>
Change Organization	<p>When you import a REST Service, you can import it into the same organization in the target CentraSite registry as in the source CentraSite registry from which they were exported, or you can assign a new organization.</p> <p>The Change Organization field is type-ahead field. As you type characters in this field, the dialog box lists the organization names that match the characters you specify.</p>
Retain lifecycle state	<p>This option determines whether the lifecycle state of the imported REST Service is preserved. Enable the option to retain the lifecycle state of the REST Service which is imported.</p>
Overwrite existing entities	<p>This option specifies that an existing REST Service with the same uuid in the target CentraSite registry will be overwritten, even if the REST Service in the archive is older than the one in the target CentraSite registry. Enable the option to overwrite the existing REST Service.</p>
Import groups that the user belongs to	<p>This option determines that when you import a user, whether you want to import the groups that the user belongs to. This applies only to user-defined groups. System-defined groups are not imported. Enable the option to import the groups.</p>
Ignore API keys and OAuth2 tokens	<p>This option determines whether the API keys and OAuth2 tokens of the imported assets are to be imported into the target CentraSite registry. Enable the option to ignore the API keys and OAuth2 tokens during the import process.</p>

6. Click **Import** to import the REST Service.

After the import operation completes, the import wizard informs you if the import was successful or if there were any errors and warnings.

7. Click **Download Import Log** to view the import logs.

The import log lists the status of all the objects stating whether they were successfully imported or if there were errors and warnings.

8. Click **OK** to terminate the import wizard.

A REST Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the REST Service asset that you just created is displayed.

Adding REST Service from Scratch

Pre-requisites:

To add a REST Service asset to an organization's asset catalog, you must belong to a role that has the Create Assets or Manage Assets permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

Before you add a REST Service asset to the catalog, you must have the RAML or Swagger specification file that you want to import. This file can reside on the file system of the computer where your browser is running or it can reside anywhere on the network, as long as its location is addressable through a URL.

Now that we are familiar with the RESTful paradigm, let us focus on documenting a REST Service using the enhanced REST data model in CentraSite Business UI.

If you are documenting our sample online phone store application as a REST Service by capturing the metadata collected from online phone store application. The metadata to manage the latest phone details online will include the following:

- A list of resources. For example, phones.
- A list of HTTP methods the Service will support for each individual resource phones. For example, GET, POST, PUT, PATCH, and DELETE.
- A list of parameters that will best describe the resource phones. For example, `features=androidosv4.3&cameraresolution=8MP`
- A list of sample HTTP request and response messages.

The sequence of documenting a REST Service using the CentraSite Business user interface can be best understood with the following illustration:

- Start by defining basic details of a REST Service and then define the base URL, schemas, and parameters.
- Easily capture resources and methods and then add the required details as you want.
- Add parameters and other details (content types, status codes) specific to each call.

- Specify samples for requests specific to each call and then the corresponding samples for expected responses.

➤ **To add a REST Service asset from scratch**

1. In the CentraSite Business UI activity bar, click **Create Asset**.

This opens the **Create Asset** wizard.

2. In the **Basic Information** profile, provide the required information for each of the displayed data fields.

Field	Description
Name	(Optional). Name of the REST Service.
Type	The asset type, REST Service .
Organization	The organization to which you want to add the REST Service. (The Organization list only displays organizations for which you have the Manage Assets permission or at least the Create Assets permission.)

Note:

This is the name that users will see when they view this REST Service asset in the CentraSite User Interfaces. Therefore, specify a meaningful name for each REST Service.

The name of a REST Service asset must be NCName-conformant, meaning that:

- The name must begin with a letter or the underscore character (_).
- The remainder of the name can contain any combination of letters, digits, or the following characters: . - _ (that is, period, dash, or underscore). It can also contain combining characters and extender characters (for example, diacriticals).
- The name cannot contain any spaces.
- Furthermore, if the REST Service name contains any non-conformant character, upon publishing the REST Service to any gateway, the non-conformant character is simply replaced with the underscore character (_) in Mediator. However, in CentraSite the REST Service name defined by you is displayed.

For more information about the NCName type, see <http://www.w3.org/TR/xmlschema-2/#NCName>

If you have not specified a **Name** for the REST Service, CentraSite automatically maps this field with the `Service Title` property, that is defined in the Swagger specification.

Field	Description
Version	(Optional). The version identifier for the REST Service. You can enter any string in this field, that is, the version identifier does not need to be numeric. You can also leave the field blank. You can later create new versions of the REST Service. The default is 1.0.

Examples:

```
0.0a
1.0.0 (beta)
Pre-release 001
V1-2007.04.30
```

If you have not specified a **Version** for the REST Service, CentraSite automatically maps this field with the `Service Version` property, which is defined in the Swagger specification.

Description (Optional). The description for the REST Service.

Note:

This is the description information that users will see when they view this REST Service asset in the CentraSite User Interfaces. Therefore, specify a meaningful description for each REST Service.

If you have not specified a **Description** for the REST Service, CentraSite automatically maps this field with the `Description` property, that is defined in the Swagger specification.

Import from The input file for the REST Service.

a Specification File

Option	Description
URL	If the input file you are importing resides on the network, you can specify its URL.
File	If the specification resides in your local file system, specify the file name. You can use the Choose button to navigate to the required folder.

- Click the **Advanced Settings** chevron to expand the additional options that are available for the Web Service asset. Provide the required information for each of the displayed fields:

Field	Description
Credentials	If you have specified a URL and the site you want to access through the URL requires user authentication, type a username and password for authentication at the URL site.

Field	Description
Resolution	<p>Select a resolution strategy, which will allow you to specify how an already existing imported and included file is handled. For each of the imported and included files you have one of these options:</p> <ul style="list-style-type: none"> ■ Create new version: Creates a new version of the file with the new content (if, for example, you want to modify a WSDL file but want to retain its previous version). ■ Always overwrite: Overwrites the importing file with new content.

4. Click **Next**.

You cannot navigate to the next screen unless all of its required attributes have been set.

5. In the **Preview** panel, review the basic information for the REST Service before you actually add to the CentraSite registry.
6. Click **Save**.

A REST Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the REST Service asset that you just created is displayed.

7. Configure the extended attributes of the REST Service asset as described later in this topic.

Adding Base URL to REST Service

After you define the REST Service, you then expose the REST Service's base URLs. A base URL path includes the hostname of the server where the REST Service is actually hosted.

When you configure the global details for a REST Service, keep the following points in mind:

- Base URL is the core design element that serves as the only way to access the identified REST Service.

CentraSite allows you to configure multiple base URL paths for the REST Service. For example, you can configure a sandbox URL for testing purposes and a production URL for accessing real-world data.

For our sample Phone Store Service, here are the sandbox and production base URLs:

Sandbox URL

```
https://www.sandbox.phonestore.com/Service/v2
```

Production URL

```
https://www.phonestore.com/Service/v2
```

For example, a base URL for our sample Phone Store Service would look like:

```
https://www.phonestore.in/Service/v2
```

A complete URL is formed by combining the resource path with the base URL.

For example, here is the URL you would use in a request to get the list of phones:

```
GET https://www.phonestore.in/Service/v2/phones
```

Or, retrieve a phone with product code is 412456

```
GET https://www.phonestore.in/Service/v2/phones/phone-412456
```

Where,

GET - HTTP request method

https://www.phonestore.in/Service/v2 - URL

phones - resource URI

412456 - path parameter

- REST Service parameter is an expression that represents a value that the client passes to the REST Service specified in the client call. Since the parameters are defined at the REST Service level, the parameters will be available for all child resources and methods below the REST Service in the hierarchy.

Configure the base URLs for the REST Service using which users would traverse to any of the REST Service's resources. In order to execute this task, you must know the URL of the server that is hosting the REST Service you intend to model.

> To add a base URL

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, REST Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and perform the following:

- a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.

b. In the list of asset types, select **REST Service**.

c. Click **OK**.

A list of defined REST Service assets is displayed in the Search Results page.

5. Click the REST Service you want to add the base URL.

This opens the REST Service details page. Also, the Actions bar displays a set of actions that are available for working with the REST Service.

6. On the Actions bar of the REST Service details page, click **Edit**.

7. Select the **Technical Details** profile. Provide the required information for each of the displayed data fields.

Field	Description
Base URL	<p>(Optional). The server base URL for the REST Service.</p> <p>Note: If you are specifying multiple base URLs for a REST Service, it must be unique among all URLs in the REST Service.</p> <p>To specify additional base URLs, use the plus button (+) next to the text box to create a new base URL input field, and type another URL.</p> <p>To remove a base URL, use the minus button (-).</p>
Sandbox	<p>(Optional). The sandbox category using which you want to classify base URL for the REST Service.</p> <p>a. Click Choose.</p> <p>A list of defined sandbox categories is displayed in the Choose Sandbox Categories dialog box.</p> <p>b. Click the chevron next to a Sandbox taxonomy to expand the categorization tree.</p> <p>c. Select the check box of the category you want to use for classifying the base URL.</p> <p>d. Click OK.</p> <p>CentraSite includes a set of predefined categories for the taxonomy node Sandbox, especially for classifying base URLs of REST Services. By default, the base URLs can be classified into the following predefined categories - Development, Production, Test.</p>

Field	Description
	<p>For information on the Sandbox categories that CentraSite supports out-of-the-box, in CentraSite Control, go to Administration > Taxonomies. In the Taxonomies page, navigate to Sandbox in the list of taxonomies.</p> <p>If you want to use sandbox categories that are not supported by CentraSite, you can define your custom categories.</p> <p>Note: Although it is possible to define subcategories for the predefined and custom categories within the Sandbox taxonomy, you cannot use these subcategories to classify the base URLs. CentraSite only displays the names of the top-level categories (that is, categories that are defined for the Sandbox taxonomy) for the classification.</p>
Namespace	The namespace for request or response message.
Request Content-Type	The content format for request message. (The Request Content-Type displays a list of the supported content formats for the REST Service.) By default, this field shows an empty value.
Response Content-Type	The content format for response message. (The Response Content-Type list displays a list of the supported content formats for the REST Service.) By default, this field shows an empty value.
Parameters - Add Parameter	<p>(Optional). One or multiple request parameters at the REST Service (API) level. The supported parameters types are:</p> <ul style="list-style-type: none"> ■ Query-String ■ Header ■ Form <p>Although CentraSite allows you to define parameters of the type, Form, at the REST Service level, these parameters are not supported at run-time. Only parameters of the type - Query-String and Header, are supported at run-time.</p> <p>Note: You cannot add more than one parameter with the same name and the same type at the REST Service level.</p> <ol style="list-style-type: none"> a. Click the Add Parameter link. This opens the Add Parameter dialog box. b. In the Add Parameter dialog box, specify an input parameter for the REST Service.

Field	Description
	To specify multiple parameters, click the Add Parameter link to add each new parameter.
	The newly created parameter is added to the Technical Details profile.
	To further update the new parameter, hover over the required parameter, and click the Edit icon. Repeat for each parameter for which you want to modify the details.

8. Click **Save**.

Adding Resource to REST Service

Resources are the basic components of a REST Service.

After you have exposed the base URIs for accessing the REST Service, you must identify and first define the resources for it. By identifying the resources in the REST Service, you can make the REST Service more useful and easier to develop.

Each resource has its own unique URI. Defining URI patterns is important because URIs enable clients to directly access a resource.

For example, consider the case of our sample Phone Store Service. Assume this Service exposes a database that defines a list of phones and the features and specifications of each phone; wherein the list of phones is represented with the collection URI and the features of each phones is represented as an individual URI.

Collection resource URI: /phones

Unique resource URI: /phones/412456

Software AG recommends that you follow these simple tips and guidelines for structuring the resource path (URI):

- **Short name URIs** as much as possible - for example, prefer /phones/412456 than /phones/phone.php?phone_id=412456
- **Straightforward and meaningful URIs** to ease use of the resource - /phones/412456
- **Consistent and predictable URIs** patterns - /phones/412456/features
- **Simple and hierarchical URIs** to represent the relationships - for example,
 - /phones
 - /phones/412456
 - /phones/412456/features
- **Nouns, not verbs** - for example, plural nouns to represent list of things - /phones; singular nouns to represent a particular thing - /phones/412456

- **Hyphens, avoid spaces or underlines** to improve and enhance aesthetic interaction with the resource - for example, prefer `/phones/412456` than `/phones/412456`
- **Lower case, avoid mixed case and upper case** to improve readability - `/phones/412456` than `/Phones/412456`

Here are some common examples for our sample resource `/phones`:

- `/phones`
- `/phones/412456`
- `/phones/412456?fields=(make,features,bodytype)`
- `/phones/412456/make`
- `/phones/search?q=(make,eq,apple)`
- `/phones/?make=apple&features=3g&price.min=44101`

In this task, you identify the resource of the REST Service and capture the resource URI for the REST Service. Indirectly, a URI address implies the relationship between each of the resources.

> To add a REST Resource

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, REST Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and perform the following:
 - a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.
 - b. In the list of asset types, select **REST Service**.
 - c. Click **OK**.

A list of defined REST Service assets is displayed in the Search Results page.

- Click the REST Service you want to add the REST Resource.

This opens the REST Service details page. Also, the Actions bar displays a set of actions that are available for working with the REST Service.

- On the actions bar of the REST Service details page, click **Edit**.
- Select the **Resource and Methods** profile.

- Click the **Add Resource** link.

This opens the **Add Resource** dialog box.

- In the **Add Resource** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Name	<p>Name of the REST Resource.</p> <div data-bbox="540 871 1461 1045" style="background-color: #f0f0f0; padding: 5px;"> <p>Note: This is the name that users will see when they view this REST Resource in the CentraSite User Interfaces. Therefore, specify a meaningful name for each REST Resource.</p> </div> <p>The name of a REST Resource must be NCName-conformant, meaning that:</p> <ul style="list-style-type: none"> ■ The name must begin with a letter or the underscore character (_). ■ The remainder of the name can contain any combination of letters, digits, or the following characters: . - _ (that is, period, dash, or underscore). It can also contain combining characters and extender characters (for example, diacriticals). ■ The name cannot contain any spaces. <p>For more information about the NCName type, see http://www.w3.org/TR/xmlschema-2/#NCName</p> <div data-bbox="540 1514 1461 1690" style="background-color: #f0f0f0; padding: 5px;"> <p>Note: If you are specifying multiple Resources for a particular REST Service, the name must be unique among all of the resources within that REST Service.</p> </div>
Resource Path	<p>The Resource URI.</p> <p>Make sure that you specify the URI with the simple tips and guidelines described above.</p>

Field	Description
Description	<p>Important: As a best practice, Software AG recommends that you adopt the following conventions when specifying a path parameter in the resource URI:</p> <ul style="list-style-type: none"> ■ Append a path parameter variable within curly {} brackets. ■ Specify a path parameter variable such that it exactly matches the path parameter defined at the Resource level. <p>(Optional). The description for the REST Resource.</p> <p>Note: This is the description information that users will see when they view this REST Resource in the CentraSite User Interfaces. Therefore, specify a meaningful description for each REST Resource.</p>
Upload Schema	(Optional). The XML Schema Definition (XSD) file for the REST Resource.
Upload Files	<p>(Optional). Input files that provide additional information about the REST Resource.</p> <p>You can use the Browse button to navigate to the required folder.</p> <p>To specify additional input files, use the plus button (+) to create a new input field, and then use the Browse button to select another input file.</p>
Parameters - Add Parameter (link)	<p>(Optional). Request Parameters for the REST Resource. The supported parameters types are:</p> <ul style="list-style-type: none"> ■ Path ■ Query-String ■ Header ■ Form <p>Although CentraSite allows you to define parameters of the type, Form, at the Resource level, these parameters are not supported at run-time. Only parameters of the type - Path, Query-String, and Header, are supported at run-time.</p> <p>Note: You cannot add more than one parameter with the same name and the same type for a REST Resource.</p>
	<p>a. Click the Add Parameter link.</p> <p>This opens the Add Parameter dialog box.</p>

Field	Description
	<p>b. In the Add Parameter dialog box, provide values for the REST Parameter.</p> <p>c. Click OK.</p> <p>The newly created parameter is added to the REST Resource.</p> <p>To further update the new parameter, hover over the required parameter, and click the Edit icon. Repeat for each parameter for which you want to modify the details.</p> <p>To specify multiple parameters, click the Add Parameter link, and provide values for the new parameters.</p>
10.	Click the chevron next to the REST Resource for which you want to display the details.
11.	To further update the new REST Resource, hover over the required Resource, and click the Edit icon. Repeat for each Resource for which you want to modify the details.
12.	To specify multiple REST Resources, click the Add Resource link, and provide values for the new REST Resource.
13.	Click Save .

Adding HTTP Method to REST Service

Understand the predefined HTTP methods and their known attributes. See the HTTP method definitions information to learn more about the common set of methods for HTTP.

The HTTP methods passed as part of an HTTP request tell the REST Service what operation needs to be done with the addressed resource.

Clients use HTTP methods to perform certain operations. Multiple methods exist - GET, POST, PUT, DELETE, HEAD, OPTIONS, PATCH, TRACE, and CONNECT.

Important:

During virtualization of a REST Service, CentraSite does not support the following HTTP methods: HEAD, OPTIONS, TRACE, and CONNECT.

This is because, when a Virtual REST Service is published to Mediator gateway, CentraSite only supports the HTTP methods: GET, POST, PUT, PATCH, and DELETE, at run-time.

Let us consider the following scenarios for our sample Phone Store Service:

Resource URI	Supported HTTP Methods	Description
/phones	GET	List all phones.
/phones	POST	Creates a new phone with product code 412456.
/phones/412456	GET	Retrieves details of a phone whose product code is 412456.
/phones/412456	DELETE	Removes a phone whose product code is 412456.
/phones/412456?fields=(make,features,bodytype)	GET	Retrieves additional details (such as Brand, Features, Body Type) of a phone whose product code is 412456.
/phones/412456/make	GET	Identifies the brand of a phone whose product code is 412456.
/phones/search?q=(make,eq,apple)	GET	Retrieves a list of all phones whose brand is Apple.
/phones/412456?make=apple&features=3g	PUT	Updates a phone whose product code is 412456, brand is Apple, and also 3G compatible.

Resource methods can also use parameters to identify or pass additional information. You can capture the sample requests and responses to facilitate clients easily interact with the resources of Service. You can set HTTP status codes to help client quickly and more easily understand the HTTP response messages. In this task, you define the valid operations for the resources. In addition, you can define the resource representation formats and the samples to represent the HTTP requests and responses.

> To add a REST Method

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, REST Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:

- a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.

- b. In the list of asset types, select **REST Service**.

- c. Click **OK**.

A list of defined REST Service assets is displayed in the Search Results page.

5. Click the REST Service you want to add the HTTP Method.

This opens the REST Service details page. Also, the Actions bar displays a set of actions that are available for working with the REST Service.

6. On the Actions bar of the REST Service details page, click **Edit**.

7. Select the **Resource and Methods** profile.

8. Locate the resource for which you want to add the HTTP Method.

9. Click the **Add Method** link.

This opens the **Add Method** dialog box.

10. In the **Add Method** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Name	Name of the REST (HTTP) Method.
Description	(Optional). The description for the REST Method.
HTTP Method	The HTTP operation you want to perform on the REST Resource. (The HTTP Method list displays a list of the supported HTTP Methods.)
Request Content-Type	The content format for HTTP Request message. (The Request Content-Type displays a list of the supported content formats for the REST Service.) By default, this field shows an empty value.
Response Content-Type	The content format for HTTP Response message. (The Response Content-Type displays a list of the supported content formats for the REST Service.) By default, this field shows an empty value.
Deprecated	(Optional). Specify if the REST Method is deprecated.

Field	Description
Parameters - Add Parameter (link)	<p>(Optional). Request Parameters for the REST Method. The supported parameters types are:</p> <ul style="list-style-type: none">■ Query-String■ Header■ Form <p>Although CentraSite allows you to define parameters of the type, Form, at the Resource level, these parameters are not supported at run-time. Only parameters of the type - Query-String, and Header, are supported at run-time.</p> <p>Note: You cannot add more than one parameter with the same name and the same type for a REST Method.</p> <ol style="list-style-type: none">Click the Add Parameter link. This opens the Add Parameter dialog box.In the Add Parameter dialog box, provide values for the REST Parameter.Click OK. <p>The newly created parameter is added to the REST Method.</p> <p>To further update the new parameter, hover over the required parameter, and click the Edit icon. Repeat for each parameter for which you want to modify the details.</p> <p>To specify multiple parameters, click the Add Parameter link, and provide values for the new parameters.</p>
Requests - Add Request (link)	(Optional). The HTTP requests to the Resources of the REST Service.
Responses - Add Response (link)	(Optional). The HTTP responses to the Resources of the REST Service.
Sample Requests and Responses - Add Request and Response (link)	(Optional). The sample requests to the Resources of the REST Service, and the corresponding sample responses from the Resources of the REST Service.

11. Click the chevron next to the REST Method for which you want to display the details.

12. To further update the new REST Method, hover over the required Method, and click the **Edit** icon. Repeat for each Method for which you want to modify the details.
13. To specify multiple REST Methods, click the **Add Method** link, and provide values for the new REST Method.
14. Click **Save**.

Adding Parameter to REST Service

Parameters are used to pass and add additional information to a request. You can use parameters as part of the URL or in the headers or as part of the message body.

Multiple parameter types exist - the most widely used parameters are path and query-string parameters.

- Path parameters, which are integral part of the request URL and correspond to the URL path variable names.

The following example shows the path parameter representations and the result expected:

GET /phones/412456 - Returns the details for a specific phone whose product code is 412456.

In the above snippet, the URL path variable name 412456 is passed as a parameter to the GET method.

Note:

CentraSite allows you to define a path parameter only at the Resource level.

- Query-String parameters, which are passed as the request URL query parameters.

The following examples show the different query-string parameter representations and the results expected:

- GET /phones?make=apple - Returns a list of all the phones that match the specified brand Apple.

- GET /phones/412456?format=JSON - Returns the details for phone whose product code is 412456 in the JSON format.

- Header parameters, which are passed as custom HTTP headers.

The following example shows the header parameter representation:

```
GET /phones?412456
Accept-Encoding: application/json
x-CentraSite-ServiceKey:66f4b263-cc6e-11e3-85a7-a2d064a5bd02
```

- Form parameters, which are encoded as the POST part of the request message body.

Important:

Although CentraSite allows you to define parameters of the type "Form" at the REST Service level, these parameters are not supported at run-time.

In this task, you define input parameters either at the REST Service level, REST Resource level, or the REST Method level. Defining a parameter at the REST Service level (in the **Technical Details** profile) means that it is inherited by all Resources, and by all methods under the individual Resources. Defining a parameter at the Resource level (in the **Add Resource** dialog box) means that it is inherited by all Methods under it. Defining it at the Method level (in the **Add Method** dialog box) only applies the parameters to that particular method; it does not affect either the REST Service level or Resource level.

> **To add a parameter**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, REST Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and perform the following:

a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.

b. In the list of asset types, select **REST Service**.

c. Click **OK**.

A list of defined REST Service assets is displayed in the Search Results page.

5. Click the REST Service you want to add the REST Parameter.

This opens the REST Service details page. Also, the Actions bar displays a set of actions that are available for working with the REST Service.

6. On the Actions bar of the REST Service details page, click **Edit**.

7. Select the **Technical Details** or **Resource and Methods** profile.

8. Click the chevron next to the **Parameters** in the **Technical Details** profile, or the **Add Resource** dialog box, or the **Add Method** dialog box, as required.

- Click the **Add Parameter** link.

This opens the **Add Parameter** dialog box.

- In the **Add Parameter** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Name	Name of the REST Parameter.
Description	(Optional). The description for the REST Parameter.
Parameter Type	The type of REST Parameter. The supported types include: <ul style="list-style-type: none"> ■ Path ■ Query-String ■ Header ■ Form
Data Type	The data type of the REST Parameter.
Required	Specifies if the parameter is mandatory or optional for invoking the REST Service.
Array	Specifies if the parameter can hold a single value or an array of values.
Possible Values	(Optional). A list of possible values for the REST Parameter.
Default Value	(Optional). A default value for the REST Parameter.

The newly created REST Parameter is added to the REST Service, or the REST Resource, or the REST Method.

- Click the chevron next to the REST Parameter for which you want to display the details.
- To further update the new REST Parameter, hover over the required Parameter, and click the **Edit** icon. Repeat for each Parameter for which you want to modify the details.
- To specify multiple REST Parameters, click the **Add Parameter** link, and provide values for the new REST Parameter.
- Click **Save**.

Adding HTTP Request to REST Service

A HTTP request describes the input to a HTTP method (as a collection of parameters) for the addressed resource.

In this task, you define the valid requests for the resources. In addition, you can define the request representation formats, and the schemas and examples to represent the HTTP requests.

➤ **To add a HTTP request**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, REST Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:

a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.

b. In the list of asset types, select **REST Service**.

c. Click **OK**.

A list of defined REST Service assets is displayed in the Search Results page.

5. Click the REST Service you want to add the HTTP Request Message.

This opens the REST Service details page. Also, the Actions bar displays a set of actions that are available for working with the REST Service.

6. On the Actions bar of the REST Service details page, click **Edit**.

7. Select the **Resource and Methods** profile.

8. In the **Add/Modify Method** dialog box, click the chevron to expand the area labeled **Requests**.

9. Click the **Add Request** link.

This opens the **Add Request** dialog box.

10. In the **Add Request** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Name	<p>This is a label that you assign as a meaningful name of the HTTP Request.</p> <p>For example, you may call a HTTP Request of type application/xml, as XML Request or XML Payload.</p> <p>You may also call a HTTP Request based on the data that it holds. For example, you could specify a POST request to create new customer as Create Customer Request.</p>
Description	(Optional). The description for the HTTP Request.
Request Content-Type	The content format for the HTTP Request message. (The Request Content-Type displays a list of the supported content formats for the REST Service.) By default, this field shows an empty value.
Schema	(Optional). The REST Payload (HTTP Request message) using an XML schema or JSON schema.
Inline or External File	<p>Specifies if the schema definition will be read from an inline text (the Inline option) or from an external file or URL (the External File option).</p> <p>To use an inline schema, type the schema definition in the Inline text box.</p> <p>To use an external schema, do the following:</p> <ul style="list-style-type: none"> ■ If the schema definition you are uploading resides on the network, specify its URL. ■ If the schema definition resides in an external file, specify the file name. You can use the Browse button to navigate to the required folder.
Example	<p>(Optional). The REST Payload (HTTP Request message) with suitable examples to demonstrate the usage of a schema. An example is an XML code or JSON code.</p>
Inline or External File	<p>Specifies if the example will be read from an inline example code (the Inline option) or from an external file or URL (the External File option).</p> <p>To use an inline example, type the example code in the Inline text box.</p> <p>To use an external example, do the following:</p> <ul style="list-style-type: none"> ■ If the example you are uploading resides on the network, specify its URL.

Note:

As a best practice, you should use the **Inline** option to include small number of data and use the **External File** option to include large number of data stored in an external file.

Field	Description
	<ul style="list-style-type: none">■ If the example resides in an external file, specify the file name. You can use the Browse button to navigate to the required folder.

Note:

As a best practice, you should use the **Inline** option to include small number of data and use the **External File** option to include large number of data stored in an external file.

The newly created HTTP Request is added to the REST Method.

11. Click the chevron next to the HTTP Request for which you want to display the details.

12. To further update or delete a HTTP Request, hover over a particular HTTP Request.

This displays icons for one or more actions that you can perform on the HTTP Request.

13. Click **Edit** or **Delete**, as required.

14. To specify multiple HTTP Requests, click the **Add Request** link, and provide values for the new HTTP Requests.

15. Click **Save**.

Adding HTTP Response to REST Service

HTTP/1.1 defines all the legal status codes. Examine these codes to determine which are appropriate for your Service.

HTTP response status codes provide information about the status of a HTTP request. The HTTP specification defines several status codes that are typically understood by clients.

In this task, you define the valid responses for the HTTP requests. In addition, you can define the response representation formats, and the schemas and examples to represent the HTTP responses.

A HTTP response indicates the success or failure of a REST Service invocation.

> To add a HTTP response

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, REST Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.
 - b. In the list of asset types, select **REST Service**.
 - c. Click **OK**.

A list of defined REST Service assets is displayed in the Search Results page.

5. Click the REST Service you want to add the HTTP Response Message.

This opens the REST Service details page. Also, the Actions bar displays a set of actions that are available for working with the REST Service.

6. On the Actions bar of the REST Service details page, click **Edit**.
7. Select the **Resource and Methods** profile.
8. In the **Add/Modify Method** dialog box, click the chevron to expand the area labeled **Responses**.
9. Click the **Add Response** link.

This opens the **Add Response** dialog box.

10. In the **Add Response** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Status Code	<p>Select a HTTP Response status code number.</p> <p>Examples</p> <ul style="list-style-type: none"> ■ HTTP 200 OK ■ HTTP 400 Bad Request ■ HTTP Error 404 Not Found ■ HTTP Error 500 Internal Server Error
Name	This is a label that you assign as a meaningful name of the HTTP Response.

Field	Description
	For example, you may call a HTTP 400 Response as <code>Validation Error</code> , instead of the <code>Bad Request</code> .
Description	(Optional). The description for the HTTP Response.
Request Content-Type	The content format for HTTP Response message. (The Response Content-Type displays a list of the supported content formats for the REST Service.) By default, this field shows an empty value.
Schema	(Optional). The REST Payload (HTTP Response message) using an XML schema or JSON schema.
Inline or External File	<p>Specifies if the schema definition will be read from an inline text (the Inline option) or from an external file or URL (the External File option).</p> <p>To use an inline schema, type the schema definition in the Inline text box.</p> <p>To use an external schema, do the following:</p> <ul style="list-style-type: none">■ If the schema definition you are uploading resides on the network, specify its URL.■ If the schema definition resides in an external file, specify the file name. You can use the Browse button to navigate to the required folder. <p>Note: As a best practice, you should use the Inline option to include small number of data and use the External File option to include large number of data stored in an external file.</p>
Example	(Optional). The REST Payload (HTTP Response message) with suitable examples to demonstrate the usage of a schema. An example is an XML code or JSON code.
Inline or External File	<p>Specifies if the example will be read from an inline example code (the Inline option) or from an external file or URL (the External File option).</p> <p>To use an inline example, type the example code in the Inline text box.</p> <p>To use an external example, do the following:</p> <ul style="list-style-type: none">■ If the example you are uploading resides on the network, specify its URL.■ If the example resides in an external file, specify the file name. You can use the Browse button to navigate to the required folder. <p>Note:</p>

Field	Description
-------	-------------

As a best practice, you should use the **Inline** option to include small number of data and use the **External File** option to include large number of data stored in an external file.

The newly created HTTP Response is added to the REST Method.

11. Click the chevron next to the HTTP Response for which you want to display the details.
12. To further update or delete a HTTP Response, hover over a particular HTTP Response.
This displays icons for one or more actions that you can perform on the HTTP Response.
13. Click the **Edit** or **Delete** icon, as required.
14. To specify multiple HTTP Responses, click the **Add Response** link, and provide values for the new HTTP Responses.
15. Click **Save**.

Adding Request and Response Messages to REST Service

REST Services can produce successful response and errors. CentraSite allows you to capture the default error responses when an exception or error occurs.

Note:

Beginning with version 9.8, although CentraSite supports the existing **Sample Requests and Responses** wizard, we enforce you use the **REST Requests and REST Responses** wizards to specify additional details about the REST payload. You may use the **Sample Requests and Responses** wizard if required.

➤ **To add a sample request and response message**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, REST Service, click **Choose**.
This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:

- a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.

- b. In the list of asset types, select **REST Service**.

- c. Click **OK**.

A list of defined REST Service assets is displayed in the Search Results page.

5. Click the REST Service you want to add the add the HTTP Request and Response Messages.

This opens the REST Service details page. Also, the Actions bar displays a set of actions that are available for working with the REST Service.

6. On the Actions bar of the REST Service details page, click **Edit**.

7. Select the **Resource and Methods** profile.

8. In the **Add/Modify Method** dialog box, click the chevron to expand the area labeled **Sample Requests and Responses**.

9. Click the **Add Request and Response** link.

This opens the **Add Sample Request and Response** dialog box.

10. In the **Add Sample Request and Response** dialog box, provide the required information for each of the displayed data fields.

Field...	Description
Request	The HTTP Request message.
	Important: As a best practice, Software AG recommends that you use sample messages that could be sent from the client to the server.
Response/Error	The HTTP Response or error message.

The newly created Sample Request and Response Message is added to the REST Method.

11. To further update or delete a Sample Request and Response Message, hover over a particular Sample Message.

This displays icons for one or more actions that you can perform on the Sample Message.

12. Click the **Edit** or **Delete** icon, as required.
13. To specify multiple Sample Request and Response Messages, click the **Add Sample Request and Response Message** link, and provide values for the new Sample Request and Response Messages.
14. Click **Save**.

Adding Status Code to REST Service

HTTP/1.1 defines all the legal status codes. Examine these codes to determine which are appropriate for your Service.

HTTP response status codes provide information about the status of a HTTP request. The HTTP specification defines several status codes that are typically understood by clients.

In this task, you define individual HTTP response status codes for each method.

HTTP status codes indicate the success or failure of an invocation.

> To add a HTTP status code

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, REST Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.
 - b. In the list of asset types, select **REST Service**.
 - c. Click **OK**.

A list of defined REST Service assets is displayed in the Search Results page.

5. Click the REST Service you want to add the HTTP Status Code.

This opens the REST Service details page. Also, the Actions bar displays a set of actions that are available for working with the REST Service.

6. On the Actions bar of the REST Service details page, click **Edit**.
7. Select the **Resource and Methods** profile.
8. In the **Add/Modify Method** dialog box, click the chevron to expand the area labeled **HTTP Status Code**.
9. Click the **Add Status Code** link.

This opens the **Add HTTP Status Code** dialog box.

10. In the **Add HTTP Status Code** dialog box, provide the required information for each of the displayed data fields.

Field...	Description
Status Code	The HTTP Response status code number. Examples <ul style="list-style-type: none">■ HTTP 200 OK■ HTTP 400 Bad Request■ HTTP Error 404 Not Found■ HTTP Error 500 Internal Server Error
Name	This is a label that you assign as a meaningful name of the HTTP Status Code. For example, you may call a HTTP 400 response as Validation Error, instead of the Bad Request.
Description	(Optional). The description for the HTTP Status Code.

11. To further update the new status code, click the **Edit** icon. Repeat for each code that you want to modify.
12. To specify multiple status codes, click the **Add Status Code** link to add each new status code.
13. If you need to delete a status code, click the **Delete** icon. Repeat for each status code that you want to delete.
14. Click **Save**.

Viewing REST Service List

You use the Search Results page to display the list of REST Service assets.

➤ To view the list of REST Service assets

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, REST Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

A list of scopes that are available to you is displayed. By default, CentraSite contains the following predefined scopes:

Scope	Description
Everything	Displays a list of the currently available assets, organizations, and users in CentraSite registry.
Assets	Displays a list of the currently available assets in CentraSite registry. This is the default scope.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:

- a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.

- b. In the displayed list of asset types, select **REST Service**.

- c. Click **OK**.

A list of defined REST Service assets is displayed in the Search Results page.

5. To filter the list to see just a subset of the available REST Service assets, type a partial string in the **Keyword** text box. Add the specified keyword to the Search Recipe, by clicking the plus symbol next to the text box, or press Enter.

The Search Results page provides the following information about each REST Service asset:

Column	Description
Name	Name of the REST Service asset.
Description	The description for the REST Service.
Asset Type	The asset type, REST Service .
Last Updated Date	The date on which the REST Service was last modified.
Owner	The user who owns the REST Service.
Organization	The organization which owns the REST Service.
Version	The user-assigned version identifier for the REST Service.

To specify the sorting preference, select an attribute from the **Sort by** list.

Note:

Use the **View** menu to display the additional attributes.

Viewing REST Service Details

Pre-requisites:

Make sure that you have the details page of a REST Service in the View mode to examine its details.

You use the details page of a REST Service to examine its RAML/Swagger documentation.

The REST Service asset type has a unique set of profiles. However, your administrator can configure the REST Service asset type to display a customized set of profiles and attributes.

The following general guidelines apply when examining the details of a REST Service in CentraSite Business UI:

- If you are not the owner of a REST Service, you cannot view the details page of the REST Service unless you have a View permission on the REST Service (granted through either a role-based permission or instance-level permission).
- You will only see profiles of the REST Service for which you have an instance-level View permission.
- You can toggle the **Resources | Methods** menu to display the details of a REST Service in either the **Resources** view or the **Methods** view.
- In addition to examining the Resources and Methods of a REST Service, you can choose to delete one or more of the top level REST components - REST Resources or the REST Methods.

REST Service Compatibility

Beginning with version 9.7, CentraSite supports the enhanced interface for REST Services (in contrast, earlier versions of CentraSite supported a standardized interface for REST Service).

Documentation of the prior REST Service interface is available to CentraSite customers who have a current maintenance contract in Empower Product Support website.

- **If you Migrate REST Services from a Pre-9.7 Release:** If you have REST Services that were created prior to version 9.7, these REST Services will continue to hold the old version's metadata in the enhanced REST Service interface implemented by version 9.7 of CentraSite.
- **If you Migrate REST Services from a 9.7 Release:** If you have REST Services that were created in the CentraSite version 9.7 using the CentraSite Business UI, these REST Services will again continue to hold the version's metadata in the enhanced REST Service interface implemented by current version of CentraSite. However, you will find the following information in the migrated REST Service:
 - The sample request and response messages that were shown under the REST Method display without changes.
 - The status codes that were shown under the REST Method will now display under the REST Response.

CentraSite Business UI offers the Resource-Centric and Method-Centric views for examining the details of a REST Service.

Using the Method-Centric View

You can access the Method-Centric view by clicking on the **Methods** menu in the **Resources and Methods** profile.

The Method-Centric view displays the available HTTP methods for a REST Service. In a displayed REST Service, if there are HTTP methods defined at the Resource level, the Method-Centric view displays the list of all HTTP methods defined at the various Resource levels for that service. This view provides you a consolidated list of the supported HTTP methods for any given resource path URI. The default display is Method-Centric view.

In short, a Method-Centric view displays the details of a REST Service in the following hierarchical pattern:

- ..> HTTP Methods...
- ... > Resource Path, Name...
 - ... > Method Description, Request Content Type, Response Content-Type
 - ... > Method Parameters > Name, Type, Description...
 - ... > HTTP Requests > Name, Description, Request Content-Type, Schema, Example...
 - ... > HTTP Responses > Status Code, Name, Description, Response Content-Type, Schema, Example...
 - ... > Sample Requests and Responses > Sample 1 > Request, Response / Error...

Using the Resource-Centric View

You can access the Resource-Centric view by clicking on the **Resources** menu in the **Resources and Methods** profile.

The Resource-Centric view displays the available resources for a REST Service. In a displayed REST Service, if there are multiple resources, and each resource defined with multiple HTTP methods, the Resource-Centric view displays all the REST components - Resources, Methods that apply to the selected Resource, and the Parameters, HTTP Requests, HTTP Responses, and Sample Requests and Responses that are defined at the various Method and Resource levels in that service. This view provides you a consolidated list of available resources for the displayed service.

In short, a Resource-Centric view displays the details of a REST Service in the following hierarchical pattern:

- ...> Resources ...
 - ... > Description, Resource Path, Resource Parameters, Documents, Schema...
 - ... > HTTP Methods, Resource Path, Resource Name...
 - ... > Description, Request Content Type, Response Content Type
 - ... > Method Parameters > Name, Type, Description...
 - ... > HTTP Requests > Name, Description, Request Content-Type, Schema, Example...
 - ... > HTTP Responses > Status Code, Name, Description, Response Content-Type, Schema, Example...
 - ... > Sample Requests and Responses > Sample 1 > Request, Response / Error...

In this task you examine the basic and type-specific attributes that are associated with a REST Service asset. You can view the resources, methods, and parameters in the **Resources** view and the **Methods** view. In addition to examining the attributes, you can delete the existing REST Parameters, REST Resources, and REST Methods.

➤ To view the details of a REST Service asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, REST Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.

b. In the displayed list of asset types, select **REST Service**.

c. Click **OK**.

A list of defined REST Service assets is displayed in the Search Results page.

5. Click the REST Service asset you want to examine the attributes.

This opens the REST Service details page. Also, the actions bar displays a set of actions that are available for working with the REST Service.

You can hover over the **info** symbol next to an attribute to display the tooltip text, which describes the purpose of the attribute. The tooltip text displays the values of the attribute's Name and Description fields that are contained in the REST Service type definition.

6. Examine the generic attributes that are displayed in the **Basic Information** profile.

7. To examine the extended attributes that are displayed in the individual profiles, follow these steps:

a. Select the profile that contains the attribute(s) you want to display.

b. Examine the attributes on the profile as required.

c. Repeat steps 7.a and 7.b for each profile for which you want to display the details.

8. To examine the extended type-specific attributes of the REST Service - Resources, Methods, Parameters, Status Codes, and the Request and Response Messages, click the **Resources and Methods** profile.

This profile displays multiple attributes which are dependent on the Resource Centric view or Method Centric view.

9. To select the Resource-Centric view or the Method-Centric view, click the **Resources | Methods** menu on the upper right-side. Depending on the view selected, the profile displays a list of REST Resources, or REST Methods.

CentraSite Business UI displays a list of the currently defined Resources, or Methods based on the Resource-Centric view or Method-Centric view that you have selected.

10. Click the chevron next to the REST Resource's name to examine its details.

11. Click the REST Method's Name button to examine its details.

12. Drill down to different levels of the REST Parameters, HTTP Requests, HTTP Responses, and the Sample Requests and Responses to examine the details of each of them.

13. Click the REST Parameter for which you want to display the details.
14. If you have made changes to a REST Parameter, click **Save**.

Modifying REST Service Details

Pre-requisites:

You use the details page of a REST Service asset to examine and modify the RAML or Swagger specification.

The asset type **REST Service** has a unique set of profiles. However, an administrator can configure this asset type to display a customized set of profiles and attributes.

The following general guidelines apply when modifying the details of a REST Service in CentraSite Business UI:

- If you are not the owner of the REST Service asset, you cannot modify the details of the REST Service, unless you have the View or Modify permission on the REST Service (granted through either a role-based permission or an instance-level permission).
- When you view the details page of a REST Service asset, you will only be able to modify the attributes of the profiles on which you have the Modify permission.
- When you hover over an attribute, CentraSite displays the tooltip text that provides a short description of the attribute's purpose.
- Some attributes accept only specific types of information. For example, if the REST Service asset type includes a URL type attribute, you must supply a URL when you modify that attribute. Other attribute types that require a specific type of value include the Date and Email attributes.
- Some attributes are designed to be read-only. You cannot modify the read-only attributes even if you have the CentraSite Administrator role.
- You can toggle the **Resources | Methods** menu to display the details of a REST Service in either the **Resources** view or the **Methods** view. The default display is Method-Centric view.
- When you are examining the Resources and Methods of a REST Service, you can choose to delete one or more of the top level REST components - REST Resources and the REST Methods.
- When you are modifying the Resources and Methods of a REST Service, you can choose to delete the other REST components - REST Parameters, HTTP Requests, HTTP Responses, and Sample Requests and Responses.
- When you view the REST Service details page in an Edit mode, you will only see an editable user interface of the Resource-Centric view. There is no Method-Centric view in the Edit mode.
- In addition to modifying the REST components of a REST Service, you can choose to delete one or more of the REST components - REST Resources, REST Methods, REST Parameters, HTTP Requests, HTTP Responses, and the Sample Requests and Responses.

- Currently, CentraSite supports only specific properties of the RAML and Swagger specifications. For example, if the Swagger specification includes a `swagger` version property, you will not be able to define the swagger's version in the REST Service details page.

Modifying Basic Details of REST Service

You use the REST Service details page to examine and modify the RAML/Swagger specifications.

In this task you examine and change the various basic and type-specific attributes associated with the REST Service. In addition, you can examine and change the various components of the REST Service - Resources, HTTP methods, Parameters, HTTP Requests and Responses, and the Sample Messages in the **Resources** view and the **Methods** view, as applicable. You can also delete the existing REST components - Resource Parameters, Method Parameters, HTTP Requests and Responses, and the Sample Messages.

➤ To modify the basic details of a REST Service

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, REST Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:
 - a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.

- b. In the list of asset types, select **REST Service**.
- c. Click **OK**.

A list of defined REST Service assets is displayed in the Search Results page.

5. Click the REST Service you want to examine and modify the attributes.

This opens the REST Service details page. Also, the actions bar displays a set of actions that are available for working with the REST Service.

You can hover over the **info** symbol next to an attribute to display the tooltip text, which describes the purpose of the attribute. The tooltip text displays the values of the attribute's Name, and Description fields that are contained in the REST Service type definition.

6. To modify the generic attributes of the REST Service that are displayed in the **Basic Information** profile, on the actions bar, click **Edit**. Change values of the attributes in the respective data fields as required.
7. Click **Save**.

Modifying Extended Details of REST Service

You use the **Resources and Methods** profile of a REST Service to view, modify, and delete the extended REST components - Resources and Methods. CentraSite Business UI displays a list of the currently defined Resources, or Methods based on the Resource-Centric view or Method-Centric view that you select.

If you have multiple Resources and Methods defined for a REST Service, you would modify these Resources and Methods using the corresponding **Edit** icons in the **Resources and Methods** profile.

> To modify the details of resources and methods

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, REST Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

A list of scopes that are available to you is displayed. By default, CentraSite contains the following predefined scopes:

Scope	Description
Everything	Displays a list of the currently available assets, organizations, and users in CentraSite registry.
Assets	Displays a list of the currently available assets in CentraSite registry. This is the default scope.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:

- a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.

- b. In the list of asset types, select **REST Service**.

- c. Click **OK**.

A list of defined REST Service assets is displayed in the Search Results page.

5. Click the REST Service you want to examine and modify the attributes.

This opens the REST Service details page. Also, the actions bar displays a set of actions that are available for working with the REST Service.

You can hover over the **info** symbol next to an attribute to display the tooltip text, which describes the purpose of the attribute. The tooltip text displays the values of the attribute's Name, and Description fields that are contained in the REST Service type definition.

6. To modify the REST Service details that are displayed in the **Resource and Methods** profile, on the actions bar, click **Edit**.

7. Select the **Resource and Methods** profile. Add or modify the REST information at the Resource level and at the Method level, as required.

- Resource level details include the basic information for a REST Resource, and its Request Parameters.
- Method level details include the basic information for a REST Method, its Request parameters, Content Types, Status Codes, and HTTP Messages that are defined for the REST Method.

8. To modify the details of a REST Resource, follow these steps:

- a. In the list of Resources, hover over the Resource for which you want to modify the attributes.

This displays icons for one or more actions that you can perform on the Resource.

- b. Click the **Edit** icon.

This opens the **Edit Resource** dialog box

- c. In the **Edit Resource** dialog box, provide the required information for each of the displayed data fields:

Field	Description
Name	Name of the REST Resource. Make sure the name you specify in this field is a valid value for NCName.
Resource Path	The Resource URI.
Description	(Optional). The description for the REST Resource.
Upload Schema	(Optional). The XML Schema Definition (XSD) file for the REST Resource. Note: If you have a REST Service that uses XML as content, then you can optionally upload an XML schema document.
Upload Files	(Optional). Input files that provide additional information about the REST Resource.
Parameters	(Optional). Request Parameters for the REST Resource. Modify an Existing Parameter <ol style="list-style-type: none">In the list of Parameters, hover over the Parameter for which you want to modify the attributes. This displays icons for one or more actions that you can perform on the Parameter.Click the Edit icon. This opens the Edit Parameter dialog box.In the Edit Parameter dialog box, provide new values for the fields.Click OK.Repeat for each Parameter that you want to modify in the REST Resource. Delete an Existing Parameter <ol style="list-style-type: none">In the list of Parameters, hover over the Parameter you want to delete. This displays icons for one or more actions that you can perform on the Parameter.Click the Delete icon.Repeat for each Parameter that you want to delete from the REST Resource. Add a New Parameter <ol style="list-style-type: none">Click the Add Parameter link.

- | Field | Description |
|-------|---|
| | This opens the Add Parameter dialog box. |
| | b. In the Add Parameter dialog box, provide values for the REST Parameter. |
| | To specify multiple parameters, click the Add Parameter link, and provide values for the new parameters. |
| | c. Click OK . |
| | d. Repeat for each Parameter that you want to add to the REST Resource. |
| 9. | To modify the details of a REST Method, follow these steps: |
| | a. In the list of Resources, hover over the Resource for which you want to modify the attributes. |
| | This displays icons for one or more actions that you can perform on the Resource. |
| | b. Click the Edit icon. |
| | This opens the Edit Resource dialog box |
| | c. In the Edit Method dialog, provide the required information for each of the displayed data fields: |

Field	Description
Name	Name of the REST (HTTP) Method.
Description	(Optional). The description for the REST Method.
HTTP Method	The HTTP operation you want to perform on the REST Resource.
Request Content-Type	The content format for HTTP Request message.
Response Content-Type	The content format for HTTP Response message.
Parameters	(Optional). Request Parameters for the REST Method.
	Modify an Existing Parameter
	a. In the list of Parameters, hover over the Parameter for which you want to modify the attributes.
	This displays icons for one or more actions that you can perform on the Parameter.
	b. Click the Edit icon.

Field**Description**

This opens the **Edit Parameter** dialog box.

- c. In the **Edit Parameter** dialog box, provide new values for the fields.
- d. Click **OK**.
- e. Repeat for each Parameter that you want to modify in the REST Method.

Delete an Existing Parameter

- a. In the list of Parameters, hover over the Parameter you want to delete.

This displays icons for one or more actions that you can perform on the Parameter.
- b. Click the **Delete** icon.
- c. Repeat for each Parameter that you want to delete from the REST Method.

Add a New Parameter

- a. Click the **Add Parameter** link.

This opens the **Add Parameter** dialog box.
- b. In the **Add Parameter** dialog box, provide values for the REST Parameter.

To specify multiple parameters, click the **Add Parameter** link, and provide values for the new parameters.
- c. Click **OK**.
- d. Repeat for each Parameter that you want to add to the REST Method.

Requests

(Optional). HTTP Requests indicating the operation that could be performed with the addressed REST Resource.

Modify an Existing Request

- a. In the list of HTTP Requests, hover over the Request for which you want to modify the attributes.

This displays icons for one or more actions that you can perform on the Request.
- b. Click the **Edit** icon.

This opens the **Edit Request** dialog box.
- c. In the **Edit Request** dialog box, provide new values for the fields.

Field	Description
	<ul style="list-style-type: none"> d. Click OK. e. Repeat for each Request that you want to modify in the REST Method. <p>Delete an Existing Request</p> <ul style="list-style-type: none"> a. In the list of HTTP Requests, hover over the Request you want to delete. <ul style="list-style-type: none"> This displays icons for one or more actions that you can perform on the Request. b. Click the Delete icon. c. Repeat for each Request that you want to delete from the REST Method. <p>Add a New Request</p> <ul style="list-style-type: none"> a. Click the Add Request link. <ul style="list-style-type: none"> This opens the Add Request dialog box. b. In the Add Request dialog box, provide values for the HTTP Request. <ul style="list-style-type: none"> To specify multiple requests, click the Add Request link, and provide values for the new requests. c. Click OK. d. Repeat for each Request that you want to add to the REST Method.
Responses	<p>(Optional). HTTP Responses indicating the success or failure of a request invocation.</p> <p>Modify an Existing Response</p> <ul style="list-style-type: none"> a. In the list of HTTP Responses, hover over the Response for which you want to modify the attributes. <ul style="list-style-type: none"> This displays icons for one or more actions that you can perform on the Response. b. Click the Edit icon. <ul style="list-style-type: none"> This opens the Edit Response dialog box. c. In the Edit Response dialog box, provide new values for the fields. d. Click OK. e. Repeat for each Response that you want to modify in the REST Method.

Field	Description
	<p data-bbox="451 254 818 285">Delete an Existing Request</p> <ol data-bbox="451 317 1385 625" style="list-style-type: none"><li data-bbox="451 317 1385 380">a. In the list of HTTP Responses, hover over the Response you want to delete. This displays icons for one or more actions that you can perform on the Response.<li data-bbox="451 506 786 537">b. Click the Delete icon.<li data-bbox="451 562 1338 625">c. Repeat for each Response that you want to delete from the REST Method. <p data-bbox="451 657 745 688">Add a New Response</p> <ol data-bbox="451 720 1385 1119" style="list-style-type: none"><li data-bbox="451 720 1057 814">a. Click the Add Response link. This opens the Add Response dialog box.<li data-bbox="451 835 1321 1003">b. In the Add Response dialog box, provide values for the HTTP Response. To specify multiple requests, click the Add Response link, and provide values for the new requests.<li data-bbox="451 1024 630 1056">c. Click OK.<li data-bbox="451 1087 1385 1119">d. Repeat for each Response that you want to add to the REST Method.
Sample Requests and Responses	(Optional). Sample Requests to the Resources of the REST Service, and the corresponding Sample Responses from the REST Service.
	<p data-bbox="451 1234 824 1266">Modify an Existing Sample</p> <ol data-bbox="451 1297 1385 1791" style="list-style-type: none"><li data-bbox="451 1297 1385 1465">a. In the list of Sample Requests and Responses, hover over the Sample for which you want to modify the attributes. This displays icons for one or more actions that you can perform on the Sample.<li data-bbox="451 1486 1354 1581">b. Click the Edit icon. This opens the Edit Sample Request and Response dialog box.<li data-bbox="451 1602 1354 1675">c. In the Edit Sample Request and Response dialog box, provide new values for the fields.<li data-bbox="451 1696 630 1728">d. Click OK.<li data-bbox="451 1759 1385 1791">e. Repeat for each Sample that you want to modify in the REST Method. <p data-bbox="451 1812 812 1843">Delete an Existing Sample</p>

Field	Description
	<ol style="list-style-type: none"> a. In the list of Sample Requests and Responses, hover over the Sample you want to delete. This displays icons for one or more actions that you can perform on the Sample. b. Click the Delete icon. c. Repeat for each Sample that you want to delete from the REST Method.
	<p>Add a New Sample</p> <ol style="list-style-type: none"> a. Click the Add Request and Response link. This opens the Add Sample Request and Response dialog box. b. In the Add Sample Request and Response dialog box, provide values for the Sample. To specify multiple samples, click the Add Response link, and provide values for the new samples. c. Click OK. d. Repeat for each Sample that you want to add to the REST Method. <p>10. Click Save.</p>

Generating Swagger 2.0-Compliant File for a REST Service

CentraSite Business UI enables you to generate Swagger 2.0-compliant file for Swagger REST Services. The generated Swagger file helps to integrate CentraSite with other third-party providers following the Swagger 2.0 specifications.

You would generate a Swagger 2.0-compliant file for a REST Service to:

- convert a RAML 0.8 specification of a REST Service into a Swagger 2.0 file.
- capture complete and up-to-date information of a REST Service that was created with a RAML 0.8 or Swagger 2.0 specification into a Swagger 2.0-compliant file.
- create a Swagger 2.0-compliant file for a REST Service just created from scratch.

You can generate a Swagger 2.0-compliant file for the REST Service in the following ways:

- **Using the Download Documents action:** When you execute the **Download Documents** action for a REST Service in order to have an up-to-date information of the REST Service, CentraSite generates the Swagger 2.0-compliant file with the full and up-to-date information of the REST Service.

- **By Publishing/Republishing REST Service to API Portal gateways:** When you re-publish a REST Service to an API Portal gateway, CentraSite generates the Swagger 2.0-compliant file with the complete and up-to-date information of the REST Service.
- **Execution of predefined Swagger Generator policy:** Whenever you modify the information of a REST Service, CentraSite triggers the predefined Swagger Generator policy and generates the Swagger 2.0-compliant file with the complete and up-to-date information of the REST Service.

The Swagger Generator policy containing the built-in Generate Swagger 2.0 File policy action initiates the generation of Swagger v2.0-compliant file, while it is in the Productive state. By default, the Swagger Generator policy is installed in the New (inactive) state. To activate the policy, you must change the policy's lifecycle state to the Productive (active) state.

In order for the generated Swagger 2.0-compliant file to be valid and referenced correctly, the following preconditions must all be met:

- Name of a HTTP method must be unique within the REST Service. If a name is not provided for the HTTP method, CentraSite uses the HTTP method in combination with the resource path.
- The HTTP request and response messages with a valid JSON schema definition is only mapped properly in the Swagger 2.0-compliant file. If the HTTP request or response message includes an XML schema definition, then CentraSite ignores the XSD.
- If one or more schemas are referred by another schema in the input specification file, then such referenced schemas are mapped as inline schemas to each HTTP method. The resulting schema definition in the generated Swagger 2.0-compliant file does not contain any reference from one schema to the other.

➤ To generate Swagger 2.0-compliant file for REST Service

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link that is located in the upper-left corner of the menu bar.
- Click the **Search** icon that is located next to the **Scope** list.

2. In the **Additional Search Criteria** list, select **Asset Types**. Do one of the following:

1. To search for the assets of type, REST Service, click **Choose**. This opens the **Choose Asset Types** dialog box.

A list of scopes that are available to you is displayed. By default, CentraSite contains the following predefined scopes:

Scope	Description
Everything	Displays a list of the currently available assets, organizations, and users in CentraSite registry.

Scope	Description
Assets	Displays a list of the currently available assets in CentraSite registry. This is the default scope.

2. In the **Choose Asset Types** dialog box, select the **Assets** option button.
 - a. Click the chevron to expand the list of asset types.
 - b. Select one or more asset types, and then click **OK**.
3. Click the chevron to expand the list of asset types that are available to you.
4. Select **REST Service**.

A list of currently defined REST Service assets is displayed in the Search Results page.

3. Click the REST Service you want to generate the Swagger 2.0-compliant file.

The REST Service details page is displayed. Also, the Actions bar displays a set of actions that are available for working with the displayed Service.

4. Click **Download Documents**.

The **Download Documents** dialog box contains the **Include all applicable supporting documents** option, which allows you to include the associated documents (from the Supporting Document Library) in the archive file. By default, the supporting documents are included in the archive file.

5. Click **Download**.

This initiates the creation of an archive file. Once the archive file is created, CentraSite appends the generated Swagger 2.0-compliant file to the **Documents** attribute in the **Specification** profile. The **Specification** profile will be enabled in the details page of a REST API, only if the Swagger Generator policy is in the *Productive* state.

The Swagger 2.0-compliant file is prefixed with `Generated Swagger File:` and represented in a specific format: `<Name of the REST Service>.json`.

Example: `Generated Swagger File: Weather Service.json`

Note:

The generated Swagger 2.0-compliant file exhibits the Swagger mappings defined for the REST Service in CentraSite. You can easily compare the generated Swagger 2.0-compliant file with the standard Swagger 2.0 specification and see if there is any discrepancy.

Structure of the Archive File

The archive file is organized as a directory that holds a collection of downloaded files. If any of the names of the downloaded files are not unique; then such files are stored with consecutive numbers (for example, `SwaggerA_1.json`, `SwaggerA_2.json`, and so on.).

When you generate a Swagger 2.0-compliant file for the REST Service, for example, `Weather Service`, that was previously imported into CentraSite using a RAML specification, `Weather.raml`, and has an associated document in the Supporting Document Library, `Weather Data.docx`, the resulting archive file expands into a folder with the following files:

- `Weather.raml`
- `Weather Service.json`
- `Weather Data.docx`

In a later stage, when you generate the Swagger 2.0-compliant file for the same `Weather Service`, which already contains a Swagger 2.0-compliant file in the archive file, CentraSite overwrites the existing Swagger 2.0-compliant file.

Deleting REST Services

If you are not the owner of a REST Service asset, you cannot delete the REST Service unless you have Full permission on the REST Service (granted through either a role-based permission or instance-level permission).

The following general guidelines apply when deleting a REST Service in CentraSite Business UI:

- Deleting a REST Service permanently removes it from the catalog.
- A REST Service can only be deleted if it is not the target of an association from another registry object.
- When you delete a REST Service, CentraSite removes the catalog entry for the REST Service (that is, it removes the instance of the REST Service from CentraSite's object database). Also note that:
 - The performance metrics and event information of the REST Service are also deleted.

Note:

When you delete the REST Service, this information is deleted by the built-in action **Delete RuntimeEvents and RuntimeMetrics of Service**, which is installed with CentraSite.

- When you delete a composite REST Service, all of its nonshared components are also deleted.
- Deleting a REST Service will *not* remove:
 - Other assets to which the REST Service refers (unless the reference is to an asset that is a nonshared component of the REST Service you are deleting). For example, if you are deleting a REST Service with a Consumes or Consumed By relationship with other services in the registry, the related services will not be deleted.
 - Supporting documents that are attached to the REST Service.
 - Earlier versions of the REST Service. Only the latest version of the REST Service can be deleted; to remove earlier versions, they must be purged.

- You cannot delete a REST Service if:
 - The REST Service is in a pending state (for example, awaiting approval).
 - Any user in your CentraSite registry is currently modifying the REST Service.
 - The REST Service is associated with access tokens such as API key or OAuth token.

Note:

In such cases, you can delete the REST Service, only when you revoke and then delete the access tokens associated with it. For more information on revoking the access tokens, refer to ["Revoking Access Tokens as API Consumer" on page 1449](#) or ["Revoking Access Tokens as API Provider" on page 1448](#). Similarly, for deleting the access tokens, refer to ["Deleting Access Tokens" on page 1450](#).

> To delete REST Service assets

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.This displays a list of assets in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, REST Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:
 - a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.
 - b. In the displayed list of asset types, select **REST Service**.
 - c. Click **OK**.A list of defined REST Services is displayed in the Search Results page.
5. Select one or multiple REST Services you want to delete.
6. On the actions bar of the Search Results page, click **Delete**.

You can also delete a single REST Service from the actions bar of its details page.
7. When you are prompted to confirm the delete operation, click **Yes**.

Note:

If you have selected a set of REST Services, where one or multiple REST Services are in a pending state (for example, awaiting approval), CentraSite ignores the pending list of REST Services, and deletes any remaining REST Services for which you have the required permission.

OData Service Management

This section describes operations you can perform to manage OData services through CentraSite Business UI.

About OData Service Assets

Open Data Protocol (OData) enables the creation of REST-based services, which allow resources to be exposed as endpoints and identified using the Uniform Resource Identifiers (URIs). In general, OData is represented by an abstract data model called Entity Data Model (EDM). This Entity Data Model allows Web clients to publish and edit REST services and their resources using simple HTTP messages.

OData leverages the principles of HTTP, REST and ATOM, and combines the simplicity of REST and SOAP metadata definitions to describe service interfaces, data models, and semantics.

The descriptions in this topic are based on a sample OData service, TripPinService.

Instructions throughout this guide use the term *OData Services* when referring to OData Service assets.

The Entity Data Model (EDM)

This section provides a high-level description of the EDM, which is the underlying abstract data model used by OData services.

An OData Service Metadata Document describes its data in EDM terms using an XML language. The remainder of this section provides a brief description of the Entity Data Model and defines how EDM constructs are mapped to the resources of the CentraSite OData model.

The following table lists the OData EDM components that are mapped to an OData service in CentraSite:

OData (EDM) Components	Description
Service	The simple OData Service which implements the Open Data Protocol (OData).
Schema	The Schema(s) exposed by the OData service. The schema in XML language describes the service's data in EDM terms. For example: <Schema xmlns="http://docs.oasis-open.org/odata/ns/edm">

OData (EDM) Components	Description
Namespace	<p>The Namespace used by OData services when representing data in XML-based formats. The URI identifying the namespace is <Schema xmlns="http://docs.oasis-open.org/odata/ns/edm" Namespace="services.odata.org.TripPin"></p>
Version	<p>The Version of the OData protocol required to consume the service. The supported OData protocol versions are 2.0 and 4.0. For example,</p> <pre data-bbox="505 554 1422 617"><edm:Edmx xmlns:edm="http://docs.oasis-open.org/odata/ns/edm" Version="4.0"></pre>
EnumType	<p>Enumeration Types (for example, PersonGender) represent a series of related values. Enumeration types expose the related values as members of the enumeration.</p> <p>The following example shows a simple enum for our sample TripPinService:</p> <pre data-bbox="505 821 1013 968"><EnumType Name="PersonGender"> <Member Name="Male" Value="0"/> <Member Name="Female" Value="1"/> <Member Name="Unknown" Value="2"/> </EnumType></pre>
ComplexType	<p>Complex Types are structured types (for example, City, Location, Airport Location and so on) consisting of a list of properties (for example, CountryRegion, Name, Address, and so on) but with no key, and thus can only exist as a property of a containing entity or as a temporary value.</p> <p>The following example shows a simple complex type for our sample TripPinService:</p> <pre data-bbox="505 1255 1214 1486"><ComplexType Name="City"> <Property Name="CountryRegion" Type="Edm.String" Nullable="false"/> <Property Name="Name" Type="Edm.String" Nullable="false"/> <Property Name="Region" Type="Edm.String" Nullable="false"/> </ComplexType></pre>
EntityType	<p>Entity Types (for example, Person, Airline and so on) are structured records consisting of named and typed properties and key properties whose values (for example, UserName, AirlineCode and so on) uniquely identify one instance from another.</p> <p>The following example shows a simple entity type for our sample TripPinService:</p> <pre data-bbox="505 1728 1143 1875"><EntityType Name="Person" OpenType="true"> <Key> <PropertyRef Name="UserName"/> </Key> <Property Name="UserName" Type="Edm.String"</pre>

**OData (EDM)
Components****Description**

```
    Nullable="false"/>
  <Property Name="FirstName" Type="Edm.String"
    Nullable="false"/>
  <Property Name="LastName" Type="Edm.String"
    Nullable="false"/>
  <Property Name="Emails" Type="Collection
    (Edm.String)"/>
</EntityType>
```

Property

The Property element allows the construction of structural types from a single value or a collection of values.

```
<Property Name="UserName" Type="Edm.String"
  Nullable="false"/>
<Property Name="Emails" Type="Collection
  (Edm.String)"/>
```

NavigationProperty The Navigation Property allows navigation from an entity to related entities.

In the following example, the Person entity type has the navigation properties, for example, Friends, Trips, and Photo:

```
<EntityType Name="Person" OpenType="true">
...
  <NavigationProperty Name="Friends" Type=
    "Collection(services.odata.org.TripPin.Person)"/>
  <NavigationProperty Name="Trips" Type=
    "Collection(services.odata.org.TripPin.Trip)"
    ContainsTarget="true"/>
  <NavigationProperty Name="Photo" Type=
    "services.odata.org.TripPin.Photo"/>
</EntityType>
```

EntityContainer

An Entity Container corresponds to a logical data store and contains zero or more entity sets and function imports.

A full example of an entity container is as follows:

```
<EntityContainer Name="DefaultContainer">
  <EntitySet Name="Photos"
    EntityType="services.odata.org.TripPin.Photo"/>
  <EntitySet Name="People"
    EntityType="services.odata.org.TripPin.Person"/>
  <EntitySet Name="Airlines"
    EntityType="services.odata.org.TripPin.Airline"/>
  <FunctionImport Name="GetNearestAirport"
    Function="services.odata.org.TripPin.GetNearestAirport"
  <EntitySet="Airports"
    IncludeInServiceDocument="true">
  </FunctionImport>
</EntityContainer>
```

EntitySet

An Entity Set element represents a single entity or a collection of entities of a specific entity type in the data model.

OData (EDM) Components	Description
	<p>For example, the entity set identified by the URI <code>http://services.odata.org/V4/TripPinService/People('scottketchum')/Friends</code> or the collection of entities identified by the "Friends" navigation property in <code>http://services.odata.org/V4/TripPinService/People('scottketchum')/Friends('russellwhyte')/Trips</code> identifies a feed of entries exposed by the OData service.</p>
Singleton	<p>Singletons are single entities which are accessed as children of the entity container.</p>
	<p>A simple example of a singleton is as follows:</p>
	<pre><Singleton Name="Me" Type="services.odata.org.TripPin.Person"> <NavigationPropertyBinding Path="Friends" Target="People"/> <NavigationPropertyBinding Path="Photo" Target="Photos"/> </Singleton></pre>
FunctionImport	<p>The Function Import element represents a Function in an entity model.</p>
	<p>A simple example of a Function Import is as follows:</p>
	<pre><FunctionImport Name="GetNearestAirport" Function="services.odata.org.TripPin.GetNearestAirport" EntitySet="Airports" IncludeInServiceDocument="true"> <Annotation Term="Org.OData.Core.V1.ResourcePath" String="services.odata.org.TripPin.GetNearestAirport"/> </FunctionImport></pre>
Function IsBound	<p>The Function IsBound element denotes if the function is bound to a specific entity type in an entity model.</p>
	<p>A simple example of a Function IsBound is as follows:</p>
	<pre><Function Name="GetFavoriteAirline" IsBound="true" EntitySetPath="person/Trips/PlanItems/ Services.odata.org.TripPin.Flight/Airline" IsComposable="true"> <Parameter Name="person" Type="services.odata.org.TripPin.Person" Nullable="false"/> <ReturnType Type= "services.odata.org.TripPin.Airline" Nullable="false"/> </Function></pre>
Function Parameter	<p>The Function Parameter element represents a parameter to the function.</p>
	<p>The following example demonstrates a Function that contains two parameters:</p>
	<pre><Function Name="GetNearestAirport" IsComposable="true"> <Parameter Name="lat" Type="Edm.Double"</pre>

OData (EDM) Components**Description**

```
    Nullable="false"/>
  <Parameter Name="lon" Type="Edm.Double"
    Nullable="false"/>
  <ReturnType Type="services.odata.org.TripPin.Airport"
    Nullable="false"/>
</Function>
```

ActionImport

The Action Import element represents an Action in an entity model.

A simple example of an Action Import is as follows:

```
<ActionImport Name="ResetDataSource"
  Action="services.odata.org.TripPin.ResetDataSource"/>
```

Action IsBound

The Action IsBound element denotes if the action is bound to a specific entity type in an entity model.

A simple example of an Action IsBound is as follows:

```
<Action Name="ShareTrip" IsBound="true">
  <Parameter Name="person"
    Type="services.odata.org.TripPin.Person"
    Nullable="false"/>
  <Parameter Name="userName" Type="Edm.String"
    Nullable="false"/>
  <Parameter Name="tripId" Type="Edm.Int32"
    Nullable="false"/>
</Action>
```

Metadata for OData Services

An OData service is best described using two types of OData Service Metadata documents:

- **Service Document** - The service document serves as the entry point for navigating to the OData service resources. This document lists all of the resources, entity sets, functions and singletons. The service document is typically available at the Base URL in the ATOM or JSON format. For example, this URI <http://services.odata.org/V4/TripPinService> identifies the service document for a sample OData service.
- **Metadata Document** - The metadata document describes the Entity Data Model (that is, the structure and organization of the OData service resources) exposed as HTTP endpoints by that particular service. This document describes the entity types, entity sets, functions and actions. For example, this URI [http://services.odata.org/V4/TripPinService/\\$metadata](http://services.odata.org/V4/TripPinService/$metadata) identifies the metadata document for a sample OData service.

The CentraSite OData Model

The CentraSite OData Model provides the resource-oriented interface for an OData service ensuring that the representation of OData services is in-line with the representation of REST services.

An OData service is represented in CentraSite by an asset instance of the “OData Service” type, which is one of the predefined types installed with CentraSite. An OData service instance in CentraSite precisely describes a collection of OData Resources that represent the resource structure of an OData EDM.

This section describes how the EDM objects and properties (as described in the Service Document URL or the Metadata Document) are represented in the CentraSite OData model.

Note:

CentraSite supports OData versions 2.0 and 4.0.

EDMX to CentraSite OData Service Mappings

The following table lists the corresponding representations of OData Entity Data Model in the Resource Oriented Architecture (ROA) of the Business UI.

This EDM Component...	Is Referenced Using the OData Service Field...
Service	<p>OData Service</p> <p>The OData service is represented by an asset instance of the type OData Service in the CentraSite registry.</p>
Version	<p>Version</p> <p>The OData version (for example, 2.0 or 4.0) is represented by the Version field in the OData API's Technical Details profile.</p>
EntityType	<p>Entity Types</p> <p>The OData entity type (for example, Person) is represented by the Entity Types field in the Entity Sets. The Entity Types field is contained in the OData Resources profile.</p>
NavigationProperty	<p>Navigation Properties</p> <p>The OData navigation property (for example, Friends, Trips, Photo) is represented as an OData Resource and denoted as OData Navigation Properties inside the OData Resources profile.</p>
EntityContainer	<p>OData Service</p> <p>The OData entity container (for example, DefaultContainer) is represented by a collection of Attributes which hold data that is specific to the OData service.</p>
EntitySet	<p>Entity Sets</p> <p>The OData entity set (for example, photos, people, airlines and so on) is represented as an OData Resource and denoted as OData Entity Sets inside the OData Resources profile.</p>

This EDM Component... Is Referenced Using the OData Service Field...**Singleton****Singletons**

The OData singleton (for example, Me) is represented as an OData Resource and denoted as OData **Singletons** inside the **OData Resources** profile.

FunctionImport**Function Imports**

The OData function import (for example, GetNearestAirport) is represented as an OData Resource and denoted as OData **Function Imports** inside the **OData Resources** profile.

Function Parameter**Resource Parameters**

The OData function parameter or action parameter (for example, person, trip) is represented as an OData Parameter and denoted as OData **Parameter** of the type **Path** inside the **OData Resources** profile.

ActionImport**Action Imports**

The OData action import (for example, ResetDataSource) is represented as an OData Resource and denoted as OData **Action Imports** inside the **OData Resources** profile.

The EDM components currently not supported in CentraSite are as follows:

- Schema
- Namespace
- EnumType
- ComplexType
- Property
- Function IsBound
- Action IsBound

Registry and Repository Entries for an OData Service

The OData components are represented in CentraSite by a set of related entries in the registry and in the repository. When an OData service is imported, the appropriate entries are created. Some of the entries are visible in the detail pages of the OData service asset; others are not displayed in the detail pages but are displayed when you use the asset navigator feature.

Supported HTTP Methods for OData Resources

The following table lists the HTTP methods supported for OData Resources. Not all of the HTTP methods are supported for an OData Resource.

OData Resource Supported HTTP Methods

Entity Sets Resource (collection)	■ GET
	■ POST
Entity Sets Resource (single)	■ GET
	■ PUT
	■ PATCH
	■ DELETE
Singletons Resource	■ GET
	■ PUT
	■ PATCH
Function Imports Resource	■ GET
Action Imports Resource	■ POST
Navigation Properties Resource (collection)	■ GET
	■ POST
Navigation Properties Resource (single)	■ GET
	■ PUT
	■ PATCH
	■ DELETE

Sample OData Resource URLs

- **Resource URL:** [http://services.odata.org/V4/TripPinService/People\('scottketchum'\)/Friends](http://services.odata.org/V4/TripPinService/People('scottketchum')/Friends)
Description: To fetch the friends of 'scottketchum'.
- **Resource URL:** [http://services.odata.org/V4/TripPinService/People\('scottketchum'\)/Trips](http://services.odata.org/V4/TripPinService/People('scottketchum')/Trips)
Description: To fetch the details on trips of 'scottketchum'.
- **Resource URL:**
[http://services.odata.org/V4/TripPinService/People\('scottketchum'\)/Friends\('russellwhyte'\)/Trips](http://services.odata.org/V4/TripPinService/People('scottketchum')/Friends('russellwhyte')/Trips)

Description: To find out if 'russellwhyte' is a friend of 'scottketchum' and fetch the details of his trips.

- **Resource URL:** [http://services.odata.org/V4/TripPinService/GetNearestAirport\(lat = 33, lon = -118\)](http://services.odata.org/V4/TripPinService/GetNearestAirport(lat = 33, lon = -118))

Description: To find the nearest Airport.

- **Resource URL:**
[http://services.odata.org/V4/TripPinService/People\(russellwhyte\)/Trips\(0\)/Microsoft.OData.Sample.Service.Models.TripPin.GetInvolvedPeople](http://services.odata.org/V4/TripPinService/People(russellwhyte)/Trips(0)/Microsoft.OData.Sample.Service.Models.TripPin.GetInvolvedPeople)

Description: To find involved people for a trip (Calling a bound function requires a fully qualified function name).

Adding OData Service using Importer

To add assets to an organization's asset catalog, you must belong to a role that has the Create Assets or Manage Assets permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

An importer in CentraSite is a utility that takes in the specification file of a particular format as the input and then creates an asset of the particular metadata format in the registry. For example, the CentraSite importer for OData Service reads an EDMX specification and from it, creates an OData Service asset that best describes the OData Service with EDMX specification. The importer also uploads the input file to the CentraSite repository and links the file to the OData Service. When you import a OData Service using an EDMX file, for example, the importer copies the EDMX file into the repository and then links the file to the OData Service.

Adding OData Service using an EDMX File

Pre-requisites:

To add an OData Service asset to an organization's asset catalog, you must belong to a role that has the Create Assets or Manage Assets permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

Before you add an OData Service asset to the catalog, you must have the EDMX specification file that you want to import. This file can reside on the file system of the computer where your browser is running or it can reside anywhere on the network, as long as its location is addressable through a URL.

> To add an OData Service asset using EDMX specification

1. In the CentraSite Business UI activity bar, click **Create Asset**.

This opens the **Create New Asset** wizard.

- In the **Basic Information** profile, provide the required information for each of the displayed data fields.

Field	Description		
Name	<p>(Optional). Name of the OData Service.</p> <p>Note: This is the name that users will see when they view this OData Service asset in the CentraSite User Interfaces. Therefore, we recommend that you specify a meaningful name for OData Service.</p> <p>An OData Service asset name can contain any characters (including spaces), and must be unique within an organization.</p> <p>If you have not specified a name for the OData service, CentraSite automatically populates the Name field with the data extracted from the OData Service URL or the EDMX file name.</p>		
Type	The asset type, OData Service .		
Organization	The organization to which you want to add the OData Service. (The Organization list only displays organizations for which you have the Manage Assets permission or at least the Create Assets permission.)		
Version	<p>(Optional). The version identifier for the OData Service. You can enter any string in this field, that is, the version identifier does not need to be numeric. You can also leave the field blank. You can later create new versions of the OData Service. The default is 1.0.</p> <p>Examples:</p> <pre>0.0a 1.0.0 (beta) Pre-release 001 V1-2007.04.30</pre>		
Description	<p>(Optional). The description for the OData Service.</p> <p>Note: This is the description information that users will see when they view this OData Service asset in the CentraSite User Interfaces. Therefore, we recommend that you specify a meaningful description for each OData Service.</p>		
Import a File	The input EDMX file for the OData Service. You may want to read the input EDMX file from a URL-addressable location on the network (the URL option) or from your local file system (the File option).		
	<table border="1"> <thead> <tr> <th>Option</th> <th>Description</th> </tr> </thead> <tbody> </tbody> </table>	Option	Description
Option	Description		

Field	Description
URL	If the EDMX file you are importing resides on the network, you can specify its URL.
File	If the specification resides in your local file system, specify the file name. You can use the Browse button to navigate to the required folder.

3. Click the **Advanced Settings** chevron to expand the additional options that are available for the Service asset. Provide the required information for each of the displayed fields:

Field	Description
Credentials	If you have specified a URL and the site you want to access through the URL requires user authentication, type a username and password for authentication at the URL site.
Resolution	Select a resolution strategy, which will allow you to specify how an already existing imported and included file is handled. For each of the imported and included files you have one of these options: <ul style="list-style-type: none">■ Create new version: Creates a new version of the file with the new content (if, for example, you want to modify an EDMX file but want to retain its previous version).■ Always overwrite: Overwrites the importing file with new content.

4. Click **Next**.

You will not be allowed to move to the next screen unless all of its required attributes have been set.

5. In the **Preview** panel, review the basic information for the OData Service before you actually add to the CentraSite registry.

6. Click **Save**.

An OData Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the OData Service asset that you just created is displayed.

7. Configure the extended attributes of the OData Service as described later in this topic.

Tip:

If you had previously imported an EDMX file that has an associated schema file and you now re-import just the schema file with modifications, your browser may not display the updated contents of the schema file. This can happen if the browser cache is not being updated

automatically. To rectify the problem, you can change your browser settings so that pages are always updated on every visit.

Adding OData Service using an Archive

Pre-requisites:

To add an OData Service asset to an organization's asset catalog, you must belong to a role that has the Create Assets or Manage Assets permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

Before you import an OData Service asset to the catalog, you must have the archive file (.zip file). This file must reside on the file system of the computer where your browser is running.

You can import an OData service using the archive file (.zip file) to which the OData service was previously exported. You can import OData services into the same CentraSite registry from which they were originally exported or to a different CentraSite registry.

> To add an OData Service using an archive

1. In the CentraSite Business UI activity bar, click **Create Asset**.

This opens the **Create New Asset** wizard. The bottom panel of the **Create New Asset** wizard displays the option to import an archive file.

2. To import an archive file, click **Choose** to navigate to the folder where the exported archive file of an OData Service asset resides, and choose the file.

When you choose a file to import, the fields in the area labeled **Basic Information** cannot be modified.

3. Click **Next**.

The **Create New Asset** wizard displays a list of the referenced objects for the Service to import. The check box next to each referenced object indicates whether the object should be imported. By default, all objects displayed are included in the import set.

4. To exclude any referenced object from the import set, clear the corresponding check box.
5. Click the **Advanced Settings** chevron to expand the additional options that are available for the OData Service asset. Provide values for each of the displayed options:

Option	Description
Change Owner	The imported OData Service can be assigned to the same owner as in the source CentraSite registry, or you can assign a new owner.

Option	Description
	The Change Owner field is type-ahead field. As you type characters in this field, the dialog box lists the user names that match the characters you specify.
Change Organization	When you import an OData Service, you can import it into the same organization in the target CentraSite registry as in the source CentraSite registry from which they were exported, or you can assign a new organization. The Change Organization field is type-ahead field. As you type characters in this field, the dialog box lists the organization names that match the characters you specify.
Retain lifecycle state	This option determines whether the lifecycle state of the imported OData Service is preserved. Enable the option to retain the lifecycle state of the OData Service which is imported.
Overwrite existing entities	This option specifies that an existing OData Service with the same uuid in the target CentraSite registry will be overwritten, even if the OData Service in the archive is older than the one in the target CentraSite registry. Enable the option to overwrite the existing OData Service.
Import groups that the user belongs to	This option determines that when you import a user, whether you want to import the groups that the user belongs to. This applies only to user-defined groups. System-defined groups are not imported. Enable the option to import the groups.
Ignore API keys and OAuth2 tokens	This option determines whether the API keys and OAuth2 tokens of the imported assets are to be imported into the target CentraSite registry. Enable the option to ignore the API keys and OAuth2 tokens during the import process.

6. Click **Import** to import the OData Service.

After the import operation completes, the import wizard informs you if the import was successful or if there were any errors and warnings.

7. Click **Download Import Log** to view the import logs.

The import log lists the status of all the objects stating whether they were successfully imported or if there were errors and warnings.

8. Click **OK** to terminate the import wizard.

An OData Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the OData Service asset that you just created is displayed.

Viewing OData Service List

You use the Search Results page to display the list of OData Service assets.

➤ **To view the list of OData Service assets**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, OData Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

A list of scopes that are available to you is displayed. By default, CentraSite contains the following predefined scopes:

Scope	Description
Everything	Displays a list of the currently available assets, organizations, and users in CentraSite registry.
Assets	Displays a list of the currently available assets in CentraSite registry. This is the default scope.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:

- a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.

- b. In the displayed list of asset types, select **OData Service**.

- c. Click **OK**.

A list of defined OData Service assets is displayed in the Search Results page.

5. To filter the list to see just a subset of the available OData Service assets, type a partial string in the **Keyword** text box. Add the specified keyword to the Search Recipe, by clicking the plus symbol next to the text box, or press Enter.

The Search Results page provides the following information about each OData Service asset:

Column	Description
Name	Name of the OData Service asset.
Description	The description for the OData Service.
Asset Type	The asset type, OData Service .
Last Updated Date	The date on which the OData Service was last modified.
Owner	The user who owns the OData Service.
Organization	The organization which owns the OData Service.
Version	The user-assigned version identifier for the OData Service.

To specify the sorting preference, select an attribute from the **Sort by** list.

Note:

Use the **View** menu to display the additional attributes.

Viewing OData Service Details

You use the OData Service details page to examine the EDMX documentation.

The following general guidelines apply when examining the details of an OData Service in CentraSite Business UI:

- If you are not the owner of an OData Service, you cannot view the details page of the OData Service unless you have a View permission on the OData Service (granted through either a role-based permission or instance-level permission).
- You will only see profiles of the OData Service for which you have an instance-level View permission.

In this task you examine the basic and type-specific attributes that are associated with an OData Service asset.

➤ To view the details of an OData Service asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, OData Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.

A list of currently defined asset types in CentraSite is displayed.
 - b. In the displayed list of asset types, select **OData Service**.
 - c. Click **OK**.

A list of currently defined OData Services is displayed in the Search Results page.

5. Click the OData Service you want to examine the attributes.

This opens the OData Service details page. Also, the actions bar displays a set of actions that are available for working with the OData Service.

You can hover over the **info** symbol next to an attribute to display the tooltip text, which describes the purpose of the attribute. The tooltip text displays the values of the attribute's Name, and Description fields that are contained in the OData Service type definition.

6. To examine the extended type-specific attributes of the OData Service, click the appropriate profiles.

Modifying OData Service Details

You use the details page of a Virtual OData Service asset to examine and modify the OData Service details.

Note:

CentraSite Business UI does not provide any EDMX editor to edit OData Service resources.

The asset type **OData Service** has a unique set of profiles. However, an administrator can configure this asset type to display a customized set of profiles and attributes.

The following general guidelines apply when modifying the details of an OData Service asset in CentraSite Business UI:

- If you are not the owner of the OData Service asset, you cannot examine or modify the details of the OData Service, unless you have the View or Modify permission on the OData Service (granted though either a role-based permission or an instance-level permission).
- When you view the details page of an OData Service asset, you will only see profiles for which you have the profile-level View permission. You will only be able to modify the attributes of the profiles on which you have the Modify permission.
- When you hover over an attribute, CentraSite displays the tooltip text that provides a short description of the attribute's purpose.

- Some attributes accept only specific types of information. For example, if the OData Service asset type includes a URL type attribute, you must supply a URL when you modify that attribute. Other attribute types that require a specific type of value include the Date and Email attributes.
- Some attributes are designed to be read-only. You cannot modify the read-only attributes even if you have the CentraSite Administrator role.

In this task you modify the basic and type-specific attributes that are associated with an OData Service asset.

➤ **To modify the details of an OData Service asset**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, OData Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:

a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.

b. In the list of asset types, select **OData Service**.

c. Click **OK**.

A list of defined OData Service assets is displayed in the Search Results page.

5. Click the OData Service you want to examine and modify the attributes.

This opens the OData Service details page. Also, the actions bar displays a set of actions that are available for working with the OData Service.

You can hover over the **info** symbol next to an attribute to display the tooltip text, which describes the purpose of the attribute. The tooltip text displays the values of the attribute's Name, and Description fields that are contained in the OData Service type definition.

6. To modify the generic attributes of the OData Service that are displayed in the **Basic Information** profile, on the actions bar, click **Edit**. Change values of the attributes in the respective data fields as required.
7. To modify the extended attributes of the OData Service that are displayed in the individual profiles, follow these steps:
 - a. Select the profile that contains the attribute(s) you want to modify.
 - b. Change values of the attributes in the respective data fields as necessary.
 - c. Repeat steps 7.a and 7.b for each profile for which you want to modify the attributes.
8. Click **Save**.

Deleting OData Services

If you are not the owner of an OData Service asset, you cannot delete the OData Service unless you have Full permission on the OData Service (granted through either a role-based permission or instance-level permission).

The following general guidelines apply when deleting an OData Service asset in CentraSite Business UI:

- Deleting an OData Service asset permanently removes the OData Service from the catalog.
- An OData Service asset can only be deleted if it is not the target of an association from another registry object.
- When you delete an OData Service asset, CentraSite removes the catalog entry for the OData Service (that is, it removes the instance of the OData Service asset from CentraSite's object database). Also note that:

- The performance metrics and event information of the OData Service are also deleted.

Note:

When you delete the OData Service asset, this information is deleted by the built-in action **Delete RuntimeEvents and RuntimeMetrics of Service**, which is installed with CentraSite.

- When you delete a composite OData Service asset, all of its nonshared components are also deleted.
- Deleting an OData Service asset will *not* remove:
 - Other assets to which the OData Service asset refers (unless the reference is to an asset that is a nonshared component of the asset you are deleting). For example, if you are deleting an OData Service asset with a Consumes or Consumed By relationship with other services in the registry, the related services will not be deleted.

- Supporting documents that are attached to the OData Service asset.
- Earlier versions of the OData Service asset. Only the latest version of an asset can be deleted; to remove earlier versions, they must be purged.
- You cannot delete an OData Service asset if:
 - The OData Service is in a pending state (for example, awaiting approval).
 - Any user in your CentraSite registry is currently modifying the OData Service.
 - The OData Service is associated with access tokens such as API key or OAuth token.

Note:

In such cases, you can delete the OData Service, only when you revoke and then delete the access tokens associated with it. For more information on revoking the access tokens, refer to ["Revoking Access Tokens as API Consumer" on page 1449](#) or ["Revoking Access Tokens as API Provider" on page 1448](#). Similarly, for deleting the access tokens, refer to ["Deleting Access Tokens" on page 1450](#).

> To delete OData Service assets

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.This displays a list of assets in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, OData Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:
 - a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.
 - b. In the list of asset types, select **OData Service**.
 - c. Click **OK**.A list of defined OData Service assets is displayed in the Search Results page.
5. Select one or multiple OData Service assets you want to delete.
6. On the actions bar of the Search Results page, click **Delete**.

You can also delete a single OData Service asset from the actions bar of its details page.

7. When you are prompted to confirm the delete operation, click **Yes**.

Note:

If you have selected a set of OData Services, where one or multiple OData Services are in a pending state (for example, awaiting approval), CentraSite ignores the pending list of OData Services, and deletes any remaining OData Services for which you have the required permission.

Application Management

This section describes operations you can perform to manage Application assets through CentraSite Business UI.

Adding an Application Asset to the Catalog

Pre-requisites:

To add assets to an organization's asset catalog, you must belong to a role that has the `Create Assets` or `Manage Assets` permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

You create an **Application** asset to specify the consumer applications that are authorized to consume a particular Service, BPEL Process, or XML Schema.

The Application asset type is one of the predefined asset types installed with CentraSite. Application assets are used by the policy-enforcement point (PEP) to determine from which consumer application a request for an asset originated. An Application asset defines the precise characteristics by which the PEP can identify or authenticate messages from a specific consumer application at run-time.

In CentraSite Control, you can add a **Application** asset to the catalog in one of the following ways:

- *You can create an Application asset from scratch*, meaning that you create the Application asset (and set its attributes) manually.
- *You can create an Application asset on-the-fly*, meaning that you create the Application asset on-the-fly from the details page of the asset you want to consume using the **Consume Asset** dialog.

When defining Application assets, keep the following points in mind:

- Any user who has permission to publish an asset to CentraSite can define an Application asset. However, not all users are generally qualified to create an asset of this type. Defining applications is a critical task that should be performed only by an administrator who is familiar with the Mediator(s), virtual services, and run-time policies in your environment.

- An Application asset becomes available to Mediator only when you synchronize the consumer application in CentraSite with the Mediator.
- Treat Application assets as global objects and make them available to all organizations. Be sure that your registry contains only one Application asset per consumer application (that is, a consumer application should be represented by *one and only one Application asset* in the registry).
- Be sure that the identifiers that you assign to an Application asset are unique to that Application asset. If multiple Application assets have the same identifier, Mediator associates the identifier with the first matching application it finds in its local list of Application assets at run time.
- If you control access to virtual services based on consumer applications (that is, you use run-time policies that include the Authorize User action), consider:
 - Including an approval step in your consumer-registration policy that requires a security administrator to review and approve the registration event.
 - Giving only a small group of knowledgeable administrators permission to modify an Application asset after it is registered to a virtual service. This prevents users from adding unauthorized identifiers to an existing Application asset and thus, allowing unauthorized consumer applications to access the virtual service.

➤ **To add an Application asset to the catalog**

1. In the CentraSite Business UI activity bar, click **Create Asset**.

This opens the **Create Asset** wizard.

2. In the **Basic Information** profile, provide the required information for each of the displayed data fields.

In this field... **Specify...**

Name Name of the Application asset.

Note:

This is the name that users will see when they view this Application asset in the CentraSite User Interfaces. Therefore, we recommend that you specify a meaningful name for Application Service.

An Application asset name can contain any characters (including spaces), and must be unique within an organization.

Type The asset type, **Application**.

Organization The organization to which you want to add the Application. (The **Organization** list only displays organizations for which you have the Manage Assets permission or at least the Create Assets permission.)

In this field...	Specify...
Initial Version	<p>(Optional). The version identifier for the Application asset. You can enter any string in this field, that is, the version identifier does not need to be numeric. You can also leave the field blank. You can later create new versions of the Application asset. The default is 1.0.</p> <p>Examples:</p> <pre>0.0a 1.0.0 (beta) Pre-release 001 V1-2007.04.30</pre>
Description	(Optional).. A comment or descriptive information about the new application asset.

3. Click **Next**.

You cannot navigate to the next screen unless all the required attributes have been set.

4. In the **Preview** panel, review the basic information for the Application before you actually add to the CentraSite registry.
5. Click **Save**.

An Application asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the Application asset that you just created is displayed.

6. Configure the extended attributes of the Application asset as described later in this topic.

Viewing Application Asset List

You use the Search Results page to display the list of Application assets.

➤ To view the list of Application assets

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Application, click **Choose**.

This opens the **Choose Asset Types** dialog box.

A list of scopes that are available to you is displayed. By default, CentraSite contains the following predefined scopes:

Scope	Description
Everything	Displays a list of the currently available assets, organizations, and users in CentraSite registry.
Assets	Displays a list of the currently available assets in CentraSite registry. This is the default scope.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and perform the following:

a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.

b. In the displayed list of asset types, select **Application**.

c. Click **OK**.

A list of defined Application assets is displayed in the Search Results page.

5. To filter the list to see just a subset of the available Application assets, type a partial string in the **Keyword** text box. Add the specified keyword to the Search Recipe, by clicking the plus symbol (+) next to the text box, or press **Enter**.

The Search Results page provides the following information about each Application asset:

Column	Description
Name	Name of the Application asset.
Description	The description for the Application.
Asset Type	The asset type, Application .
Last Updated Date	The date on which the Application was last modified.
Owner	The user who owns the Application.
Organization	The organization which owns the Application.
Version	The user-assigned version identifier for the Application.

To specify the sorting preference, select an attribute from the **Sort by** list.

Note:

Use the **View** menu to display the additional attributes.

Viewing Application Asset Details

The following general guidelines apply when examining the details of an Application asset in CentraSite Business UI:

- If you are not the owner of an Application, you cannot view the details page of the Application unless you have a View permission on the Application (granted through either a role-based permission or instance-level permission).
- You will only see profiles of the Application for which you have an instance-level View permission.

In this task you examine the basic and type-specific attributes that are associated with an Application asset.

➤ To view the details of an Application asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Application, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and perform the following:

- a. Click the chevron next to **Assets** option button.

A list of currently defined asset types in CentraSite is displayed.

- b. In the displayed list of asset types, select **Application**.

- c. Click **OK**.

A list of currently defined Application assets is displayed in the Search Results page.

5. Click the Application asset you want to examine the attributes.

This opens the Application details page. Also, the actions bar displays a set of actions that are available for working with the Application asset.

You can hover over the **info** symbol next to an attribute to display the tooltip text, which describes the purpose of the attribute. The tooltip text displays the values of the attribute's Name, and Description fields that are contained in the Application type definition.

- To examine the extended type-specific attributes of the Application asset, click the appropriate profiles.

The details page of an Application asset includes the following additional information:

Identification Profile (for Application Assets)

In this profile, you specify the precise values for the consumer identifier token(s) that you want to use for identifying and authorizing the consumers for a particular virtual API. (Alternatively, you may configure this profile to allow unrestricted access.)

For example, if you configure the Identification profile to identify and authorize consumers by IP address, the PEP extracts the IP address from a request's HTTP header at run time and searches its list of consumers for the virtual API that is defined by that IP address.

Note:

- If you want to authenticate consumers, make sure that your policy enforcement point is configured to enable authentication. For information, see *Administering webMethods Mediator*.
- For reasons of legibility some of the examples below contain break lines and may not work when pasted into applications or command line tools.

Field	Description
IPv4 Address	<p>Use this field to identify consumers based on their originating 4-byte IP address range.</p> <p>Specify a range of IPv4 addresses. Type the lowest IP address in the From field and the highest IP address in the To field. For example, 192.168.0.0 and 192.168.0.10</p> <p>The virtual API will then identify and authorize only those requests that originate from the specified IP address.</p> <p>If you need to specify additional IP addresses, use the plus button to add more rows.</p>
IPv6 Address	<p>Use this field to identify consumers based on their originating 128-bit IPv6 address.</p> <p>Specify a IPv6 address. For example, fdda:5cc1:23:4::1f</p> <p>The virtual API will then identify and authorize only those requests that originate from an IP address that lies between the specified ranges.</p>

Field	Description
	<p>If you need to specify additional IP addresses, use the plus button to add more rows.</p>
Hostname	<p>Use this field to identify consumers based on a specified host name.</p> <p>Specify the hostname. For example, <code>pcmachine.ab.com</code></p> <p>The virtual API will then identify and authorize only those requests that originate from the specified host name.</p> <p>If you need to specify additional host names, use the plus button to add more rows.</p>
HTTP Token	<p>Use this field to authenticate consumers based on the user name that is transmitted in an HTTP authentication user token.</p> <p>Specify one or more HTTP user names. For example, <code>SAGUser123</code></p> <p>The virtual API will then identify and authorize only those requests that contain the specified user name encoded and passed in the HTTP authentication user token.</p> <p>If you need to specify additional tokens, use the plus button to add more rows.</p>
WS-Security Token	<p>Use this field to authenticate consumers based on the user name that is transmitted in the SOAP or XML message header (HTTP body).</p> <p>Specify the WSS username token. For example, <code>userwss</code></p> <p>The virtual API will then identify and authorize only those requests that contain the specified user name passed in the SOAP or XML message header.</p> <p>If you need to specify additional tokens, use the plus button to add more rows.</p>
XPath Token	<p>Use this field to identify consumers based on the result of applying an XPath expression on the SOAP or XML message or request.</p> <pre data-bbox="552 1480 1453 1585"> //*[local-name()='Envelope']/* [local-name()='Body']/* [local-name()='echoInt']/* [local-name()='echoIntInput='] [.='2'] </pre> <p>The virtual API will then identify and authorize only those requests that contain the XPath and the consumers.</p> <p>If you need to specify additional tokens, use the plus button to add more rows.</p>
Consumer Certificate	<p>Use this field to identify consumers based on information in an X.509 v3 certificate.</p>

Field	Description
	Click Upload to locate and select the certificate (.cer) file.
	The virtual API will then identify and authorize only those requests that contain the specified X.509 v3 certificate in the SOAP or XML header.

Identification Profile (for Assets with Key-based Authentication)

Field	Description
API Key String	<i>Read-only. String.</i> The confidential secret key used to securely authenticate the client. This field is visible only to a consumer who requested the API key.
Expiry Date	<i>Read-only. String.</i> An expiration date for the API key.

Identification Profile (for Assets with OAuth-based Authentication)

Field	Description
Client Id	<i>Read-only. String.</i> The unique identifier that is used by the client to fetch access tokens for the virtual API.
Client Secret	<i>Read-only. String.</i> The secret key value that is used with the client identifier, serves as a password to fetch access tokens for the virtual API.
Client Name	<i>Read-only. String.</i> The name of the client (consumer application) that is attempting to get access to the virtual API.
Scope	<i>Read-only. String.</i> The scope value is the name of the virtual API. If the scope value is valid, Mediator obtains the access token. If no scope value is provided, Mediator provides the access token to the scope in which the client is allowed and adds the scope to the response.
Refresh Token	<i>Read-only. String.</i> The unique identifier used by the client to obtain a new access token when the current access token becomes invalid or expires.

API Key Scope Profile

Field	Description
API Service	<i>Read-only. String.</i> The name of the virtual API that is associated with the API key. To view details of the virtual API, click its hyperlinked name.

Modifying Application Assets

The Application asset type has a unique set of profiles. However, an administrator can configure the Application asset type to display a customized set of profiles and attributes. You can modify the basic and type-specific attributes that are associated with an Application asset.

The following general guidelines apply when modifying the details of an Application asset in CentraSite Business UI:

- If you are not the owner of the Application asset, you cannot examine or modify the details of the Application, unless you have the View or Modify permission on the Application (granted though either a role-based permission or an instance-level permission).
- When you view the details page of an Application asset, you will only see profiles for which you have the profile-level View permission. You will only be able to modify the attributes of the profiles on which you have the Modify permission.
- When you hover over an attribute, CentraSite displays the tooltip text that provides a short description of the attribute's purpose.
- Some attributes accept only specific types of information. For example, if the Application asset type includes a URL type attribute, you must supply a URL when you modify that attribute. Other attribute types that require a specific type of value include the Date and Email attributes.
- Some attributes are designed to be read-only. You cannot modify the read-only attributes even if you have the CentraSite Administrator role.

➤ To modify the details of an Application asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Application, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and perform the following:

- a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.

- b. In the list of asset types, select **Application**.

- c. Click **OK**.

A list of defined Application assets is displayed in the Search Results page.

5. Click the Application asset you want to examine and modify the attributes.

This opens the Application details page. Also, the actions bar displays a set of actions that are available for working with the Application.

You can hover over the **info** symbol next to an attribute to display the tooltip text, which describes the purpose of the attribute. The tooltip text displays the values of the attribute's Name, and Description fields that are contained in the Application type definition.

6. To modify the generic attributes that are displayed in the **Basic Information** profile, on the actions bar, click **Edit**. Change values of the attributes in the respective data fields as required.
7. To modify the extended attributes that are displayed in the individual profiles, follow these steps:
 - a. Select the profile that contains the attribute(s) you want to modify.
 - b. Change values of the attributes in the respective data fields as necessary.
 - c. Repeat steps 7.a and 7.b for each profile for which you want to modify the attributes.
8. Click **Save**.

Defining Consumer Identifiers for Application Asset

In the **Identification** profile, specify the precise values for the consumer identifiers that you have specified in the **Evaluate <name>** action.

Note:

The following general guidelines apply when you define the consumer identifiers for an Application asset:

- If you specify *multiple* identifiers, the system evaluates them with the identifier that is defined in the **Evaluate <name>** action.
- If you want to authenticate consumers, make sure that your Policy Enforcement Point (PEP) is configured to enable authentication. For information, see the webMethods Mediator documentation or the documentation for your third-party PEP.

➤ To define the consumer identifiers

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.This displays a list of assets in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Application, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and perform the following:
 - a. Click the chevron next to **Assets** option button.
 A list of defined asset types in CentraSite is displayed.
 - b. In the list of asset types, select **Application**.
 - c. Click **OK**.
5. Click the Application asset you want to define consumer identifiers.

This opens the Application details page. Also, the actions bar displays a set of actions that are available for working with the Application.

6. In the **Identification** profile, specify values for one or more consumer identifier tokens.

Note:

The value(s) that you specify in the Identification profile depend on how the run-time policy's **Evaluate <name>** actions are configured. For example, if an **Evaluate IP Address** action is configured to identify and validate consumers by their IP address, you should specify the consumer IP addresses here.

Note:

For reasons of legibility some of the examples below contain break lines and may not work when pasted into applications or command line tools.

Field	Description
Identification Token	<p>Identifies and authenticates consumers based on one or more of the following kinds of identification tokens:</p> <p>Use this field when the Evaluate <name> action is configured to identify and authenticate consumer applications by host name, HTTP user name, WSS user name or a custom token.</p> <ul style="list-style-type: none"> ■ Host Name—To identify consumers based on a specified host name, type the host name (for example, pcmachine.ab.com) in the Name field. The application asset will identify only those requests that originate from the specified host name. ■ HTTP Authentication Token—To identify and authenticate consumers based on the user name that is transmitted in an HTTP authentication user token, type the user name (for example, testuser123) in the Name field. The application asset

Field	Description
	<p>will identify only the requests that contain the specified user name encoded and passed in the HTTP authentication user token. Authentication is handled by LDAP or another external authentication mechanism. You can specify the kinds of HTTP headers that Mediator will pass from requests to consumer applications. The default is the Authorization header. To configure Mediator to pass other kinds of HTTP headers, see the Mediator documentation.</p> <ul style="list-style-type: none"><li data-bbox="454 546 1289 840">■ WS-Security Authentication Token—To identify and authenticate consumers based on the user name that is transmitted in the SOAP or XML message header (HTTP body), type the user name (for example, userwss) in the Name field. The application asset will identify only the requests that contain the specified user name passed in the SOAP or XML message header. Authentication is handled by LDAP or another external authentication mechanism.<li data-bbox="454 861 1289 1134">■ Custom identification token (XPath)—To identify consumers based on the result of applying an XPath expression on the SOAP or XML message or request, enter the XPath expression in the Name field. For example, typing <pre data-bbox="503 1008 1289 1071">//*[local-name()='Envelope']/*[local-name()='Body'] /*[local-name()='echoInt']/*[local-name()='echoIntInput']["='2'"]</pre>in the Name field will identify the requests that contain the XPath and the consumers. <p data-bbox="454 1155 1289 1239">If you need to specify additional tokens, use the plus button to add more rows.</p>
From IP-V4 Address	<p>Identifies consumers based on their originating 4-byte IP address.</p> <p>Use this field when the Evaluate IP Address action is configured to identify consumer applications based on their originating IP addresses.</p> <p>To specify an individual IP address, type the address in the From IP-V4 Address field. The application asset will identify only those requests that originate from the specified IP address. Example: 192.168.0.0</p>
To IP-V4 Address	<p>Identifies consumers based on their 4-byte IP address range.</p> <p>Use this field when the Evaluate IP Address action is configured to identify consumer applications based on their 4-byte IP address range.</p> <p>To specify a range of IP addresses, type the lowest IP address in the From IP-V4 Address field and the highest IP address in the</p>

Field	Description
	<p>To IP-V4 Address field. For example, the values 192.168.0.0 and 192.168.0.10 indicates that requests originating from any IP address that lies between the specified range will be identified by the application asset.</p>
From IP-V6 Address	<p>Identifies consumers based on their originating 6-byte IP address.</p> <p>Use this field when the Evaluate IP Address action is configured to identify consumer applications based on their originating IP addresses.</p> <p>As for IPv4 Address, but using the 128-bit IPv6 format. Example: 1234:5678:9ABC:DEF0:1234:5678:9ABC:DEF0</p>
To IP-V6 Address	<p>Identifies consumers based on their 6-byte IP address range.</p> <p>Use this field when the Evaluate IP Address action is configured to identify consumer applications based on their 6-byte IP address range.</p> <p>As for IPv4 Address, but using the 128-bit IPv6 format. For example: 1234:5678:9ABC:DEF0:1234:5678:9ABC:DEF0</p>
Consumer Certificate	<p>Identifies consumers using the X.509 certificate that is passed in the SOAP message.</p> <p>Click Upload, and select the certificate (.cer) file.</p>
Partner ID	<p>Specifies the trading partner ID.</p> <p>When a consumer application is identified in Mediator and the event logging is enabled, you can locate the partner ID in the event data of the identified consumer. You can leverage the event data for partner based analytics.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: The Partner ID attribute is introduced for integration scenarios based on webMethods Trading Networks.</p> </div>

Deleting Application Assets

If you are not the owner of an Application asset, you cannot delete the Application unless you have Full permission on the Application (granted though either a role-based permission or instance-level permission).

The following general guidelines apply when deleting an Application asset in CentraSite Business UI:

- Deleting an Application asset permanently removes the Application from the catalog.

- An Application asset can only be deleted if it is not the target of an association from another registry object.
- When you delete an Application asset, CentraSite removes the catalog entry for the Application (that is, it removes the instance of the Application asset from CentraSite's object database). Also note that:

- The performance metrics and event information of the Application are also deleted.

Note:

When you delete the Application asset, this information is deleted by the built-in action **Delete RuntimeEvents and RuntimeMetrics of Service**, which is installed with CentraSite.

- When you delete a composite Application asset, all of its nonshared components are also deleted.
- Deleting an Application asset will *not* remove:
 - Other assets to which the Application asset refers (unless the reference is to an asset that is a nonshared component of the asset you are deleting). For example, if you are deleting is an Application asset with a Consumes or Consumed By relationship with other services in the registry, the related services will not be deleted.
 - Supporting documents that are attached to the Application asset.
 - Earlier versions of the Application asset. Only the latest version of an asset can be deleted; to remove earlier versions, they must be purged.
- You cannot delete an Application asset if:
 - The Application is in a pending state (for example, awaiting approval).
 - Any user in your CentraSite registry is currently modifying the Application.

➤ **To delete Application assets**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.This displays a list of assets in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Application, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and perform the following:

- a. Click the chevron next to **Assets** option button.

A list of defined asset types in CentraSite is displayed.

- b. In the list of asset types, select **Application**.

- c. Click **OK**.

A list of defined Application assets is displayed in the Search Results page.

5. Select one or multiple Application assets you want to delete.

6. On the actions bar of the Search Results page, click **Delete**.

You can also delete a single Application asset from the actions bar of its details page.

7. Click **Yes** in the confirmation dialog box.

Note:

If you have selected a set of Applications, where one or multiple Applications are in a pending state (for example, awaiting approval), CentraSite ignores the pending list of Applications, and deletes any remaining Applications for which you have the required permission.

General Procedures across Assets

This section outlines the general procedures across assets performed through CentraSite Business UI.

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

Attaching Supporting Documents to Asset

Assets include attributes that allow you to associate documents such as a WSDL or schema and supporting documents such as programming guides, sample code and project plan with the asset.

For example, Service assets include the **Specification** profile. This profile contains several file-related attributes representing external documents such as Functional Requirements, Error Messages, Release Notes, and so forth.

You can attach a document to an asset instance in the following ways:

- You can attach any document on the network that is accessible via a URL.
- You can attach a document from the computer's file system.

- You can attach a document from your organization's supporting document library. The supporting document library is a collection of shareable documents that members of your organization have uploaded to CentraSite's document repository. For more information about the supporting document library, see the *CentraSite User's Guide*.

When attaching a document to an asset, keep the following points in mind:

- If you are not the owner of the asset, you cannot attach a document to the asset unless you have Modify permission on the asset (granted though either a role-based permission or an instance-level permission).
- CentraSite relies on file extensions to determine a file's type. When you upload a file from your local machine to the asset, be sure the name of the file on your local machine includes a file extension so that CentraSite can determine the file's type and mark it correctly in the repository.
- When you attach a document to the asset using a URL, CentraSite retrieves the document from the URL and place it in its document repository.
- Make sure that you attach the appropriate type-specific files.

Attaching Asset Definition Files

An asset definition (input) file is type-specific and depends on the type of asset to which it will be attached. The following table lists the asset types installed with CentraSite and identifies the types of files they require as input.

For this asset...	You must supply this type of file...
Web Service (including Abstract Service)	Web Service Definition Language (WSDL) file.
XML Schema	XML Schema Definition (XSD) file.
Process	XML Process Definition Language (XPDL) file.

You can also choose the option **Resolution**, which will allow you to specify how already existing imported/included files (further WSDL or schema) are handled. For each of the imported/included files you have one of these options:

- Overwrite the file with new content.
- Create a new version of the file with the new content (if, for example, you want to modify a schema but want to retain its previous version).
- Reuse any version of the file (if, for example, an intermediate version of a schema is currently referred to by a WSDL, you can redirect it to the newest version).

You can attach an input file to an asset instance in the following ways:

- You can attach a document from your file system.
- You can attach any document on the network that is accessible via a URL.

When you attach a WSDL file, keep the following points in mind:

- **The Service asset has no WSDL already attached.** In this case the WSDL will just be attached to the asset.

You have the choice between just reusing any existing version of the WSDL/schema file or uploading a new version.

- **Select an existing version.** An existing version of the WSDL/schema file is attached to the asset. The asset name will be changed to the WSDL's file name.
- **Upload a new version.** A new version of the WSDL/schema file is attached and uploaded in the repository. The asset name will be changed to the WSDL's file name.

The WSDL can only be attached if no WSDL with the same name and namespace already exists in the CentraSite repository.

- **The Service asset already has an attached WSDL file.** In this case you can attach any WSDL file to it, even with a different file name and/or different namespace.

Moreover you have the choice between just overwriting the Service asset or creating a new version.

- **Always overwrite.** The WSDL content in the repository gets replaced by the new content. The asset's name and the classifications Local-Name and namespace are modified according to the new information.
- **Always create new versions.** A new asset will be established with the information from the WSDL file which will be a new version of the attached one. The original WSDL asset will not be modified.

Limitations: The following restrictions apply for attaching a WSDL file to a Service asset:

- If a WSDL W1 is already referred to by another WSDL W2, WSDL W1 cannot be replaced by a new WSDL that has a modified file name or a modified namespace. This restriction does not apply if the create new version option is used.
- When you use the **Attach** button, you can only attach a WSDL file to the most recent version of a Service asset.
- If you attach a WSDL with a different file name and/or a different namespace, then there must not be another existing WSDL with same name and namespace.
- Consider the Service asset has a Name Validate Policy defined that enforces unique name. In this case; the WSDL file is attached only if the Service name in the WSDL is the same as the Name pattern defined in the validation policy.

Note:

The same restrictions apply when attaching an XML schema file to XML Schema asset or and XPD file to Process asset.

➤ **To attach an input file to asset**

Prerequisite: Before you begin, you must have the input file that you want to attach. This file can reside on the file system of the computer where your browser is running, or it can reside anywhere on the network, as long as its location is addressable through a URL.

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Click the asset you want to attach an input file.

This opens the Asset Details page. Also, the actions bar displays a set of actions that are available for working with the displayed asset.

3. On the actions bar of the Asset Details page, click **Attach**.

4. In the **Attach Document** dialog, specify whether the input file will be read from a URL-addressable location on the network (the **URL** option) or from your local file system (the **File** option). Do one of the following as appropriate:

- If the file you are attaching resides on the network, enable the **URL** option. Specify its URL.
- If the file resides in your local file system, enable the **File** option. Specify the file name. You can use the **Choose File** button to navigate to the required folder.

Note:

To ensure that CentraSite sets the file type correctly in the repository, the name of the file should include an extension that indicates the type of data it contains.

5. Expand **Advanced Settings** and complete the following steps as necessary.

- a. If you have specified a URL and the site you want to access via the URL requires user authentication, enter a username and password for authentication at the URL site.
- b. You can also use the **Resolution** option, which will allow you to specify how an already existing imported/included file (further WSDL or schema) is handled. For each of the imported/included files you have one of these options:

Options	Usage
Always ask	Reuse the existing WSDL/schema files referred to in the main WSDL/schema file, or replace the existing WSDL/schema files by uploading new files.
Always overwrite	Overwrite the existing WSDL/schema files in the registry with the new ones specified by the input file.
Always create new versions	Create a new version of the file with the new content.

If you choose the **Always ask** option and there is more than one input file with the same name and namespace, you can choose between these. To allow you to select the required file, the dialog **Reuse Existing Asset** lists the available assets that match the given file name, and also their available versions. You can view the description of any asset in the list; this is useful if you want to check that you have selected the correct asset's file from the list.

6. Choose **Attach**.

Attaching Documents Using URL

You can attach a document from your organization's *supporting document library*. The supporting document library is a collection of shareable documents that members of your organization have uploaded to CentraSite's document repository.

> To attach document using an URL

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Click the asset you want to attach a document.

This opens the Asset Details page.

3. Locate the attribute and click its **Attach** button. (If the attribute has existing attachments, be sure to click the bottom-most **Attach** button. If you click an **Attach** button that belongs to an existing attachment, you will replace that attachment. If you do not see an available **Attach** button, use the plus icon to display one.)

The **Attach to...** dialog is displayed.

4. Enable the **URL** option and type the document's URL into the URL text box.
5. In the **Display Name** text box, specify a name that users will see when the document is attached to a *File* attribute.
6. Click **Attach**.

Note that the **Attach** button will be disabled until a URL is specified.

7. Repeat steps 3 to 6 for each URL that you want to attach to the attribute.
8. Click **Save**.

Attaching Documents from Computer's File System

If the document that you want to attach to the asset is not already in the supporting document library, use the following procedure to attach a document from the file system of the computer where your browser is running.

Note that this procedure uploads a document to the supporting document library and then creates a link to the asset's attribute.

> To attach document using the file system

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Click the asset you want to attach a document.

This opens the Asset Details page.

3. Select the asset's profile that contains the attribute to which you want to attach the document.

4. Locate the attribute and click its **Attach** button. (If the attribute has existing attachments, be sure to click the bottom-most **Attach** button. If you click an **Attach** button that belongs to an existing attachment, you will replace that attachment. If you do not see an available **Attach** button, use the plus button to display one.)

The **Attach to...** dialog is displayed.

5. Enable the **Upload document** option and type the document's URL into the URL text box.

6. In the **Display Name** text box, specify a name that users will see when the document is attached to a *File* attribute. This is also the name by which the document will be identified in the library.

7. In the **File** text box, specify the full pathname within your operating system environment of the file that you want to upload to the supporting document library. You can use the **Choose File** button to navigate to the required file.

To ensure that CentraSite sets the file type correctly in the supporting document library, the name of the file should include an extension that indicates the type of data it contains.

8. In the **Select a Folder** text box, specify a folder in the supporting document library where the new document will be stored. Note that a type-ahead feature is provided in this text box. You can use the **Browse** button to select the required folder.

9. Click **Attach**.

Note that the **Attach** button will be disabled until a URL is specified.

10. Repeat steps 3 to 8 for each document that you want to attach to the attribute.
11. Click **Save**.

Attaching Documents from Supporting Document Library

If the document that you want to attach to the asset is not already in the supporting document library, use the following procedure to attach a document from the file system of the computer where your browser is running.

Note that this procedure uploads a document to the supporting document library and then creates a link to the asset's attribute.

» To attach document from the supporting document library

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.
2. Click the asset you want to attach a document.

This opens the Asset Details page.
3. Select the asset's profile that contains the attribute to which you want to attach the document.
4. Locate the attribute and click its **Attach** button. (If the attribute has existing attachments, be sure to click the bottom-most **Attach** button. If you click an **Attach** button that belongs to an existing attachment, you will replace that attachment. If you do not see an available **Attach** button, use the plus button to display one.)

The **Attach to...** dialog is displayed.
5. Enable the **Reuse existing document** option.
6. In the **Select a File** text box, specify the file that you want to attach to the asset from the supporting document library. You can use the **Browse** button to navigate to the required file.
7. In the **Display Name** text box, specify a name that users will see when the document is attached to a *File* attribute.
8. Click **Attach**.

Note that the **Attach** button will be disabled until a URL is specified.

9. Repeat steps 3 to 7 for each document that you want to attach to the attribute.
10. Click **Save**.

Removing Supporting Documents from Asset

You can remove a document that is attached to an asset.

> To remove supporting document from asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.This displays a list of assets in the Search Results page.
2. Click the asset whose supporting document you want to remove.
This opens the Asset Details page.
3. Select the asset's profile that contains the required File attribute.
4. Locate the document you want to remove and click the minus sign (-) next to the name. Repeat for each document that you want to remove.
5. Click **Save**.

Changing Lifecycle State of Asset

If an asset has an associated lifecycle model, you might need to switch the lifecycle state of an asset.

When changing the lifecycle state of an asset, keep the following points in mind:

- For any given lifecycle model, a list of names of users and/or groups who are allowed to move assets to new states is maintained within the definition of the lifecycle model. For each user or group, the permission to move assets to new states can be restricted to a subset of the available states in the model. When the lifecycle model is assigned to an asset and a state has users or groups defined for it, only a user who is one of the defined users or groups can make the transition of the asset into that state. If no users or groups are defined for a particular state, any user who has Modify permission on the asset can change the lifecycle state for that asset.
- Users with the Manage Lifecycle Models permission can define the list of users and groups who are allowed to enter new states in a lifecycle model.

Note that this function can be performed on a single asset or a set of assets.

> To change the lifecycle state of assets

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Click the asset whose lifecycle state you want to change.

This opens the Asset Details page. Also, the actions bar displays a set of actions that are available for working with the displayed asset.

3. On the actions bar for the asset, click **Edit**.
4. In the **Lifecycle State** box, select the state to which you want to switch the asset. (The list will contain only the states that you are permitted to assign to the asset.)
5. When you have finished making your selection, click **Save**.
6. When you are prompted to confirm the save operation, click **Yes**.

If the state change requires approval, CentraSite Business UI will initiate an approval workflow and your request for a state change will be submitted to the appropriate approvers. While the request is awaiting approval, the asset will appear in the pending mode.

Setting Permissions on Asset

To set permissions on an asset, you must have the Manage Assets permission or have the Full instance-level permission on the asset itself.

By default, everyone in your organization is permitted to view the assets that you publish. However, only you (as the owner of the asset) and users who belong to a role with the Manage Assets permission for your organization are allowed to view, edit, and delete these assets. To enable other users to view, edit, and delete an asset that you have published, you must modify the asset's permission settings.

When setting permissions on assets, keep the following points in mind:

- To set permissions on an asset, you must belong to a role that has the Manage Assets permission or have the Full instance-level permission on the asset itself.
- You can assign permissions to any individual user or group defined in CentraSite.
- The groups to which you can assign permissions include the following system-defined groups:

Group Name	Description
Users	All users within a specified organization.
Members	All users within a specified organization and its child organizations.
Everyone	All users of CentraSite <i>including guest users</i> .

- If a user is affected by multiple permission assignments, the user receives the union of all the assignments. For example, if group ABC has Modify permission on an asset and group XYZ has Full permission on the same asset, users that belong to both groups will, in effect, receive Full permission on the asset.

The same principle applies to users who have both role-based permissions and instance-level permissions on the same asset. In this case, users receive the union of the role-based permission and the instance-level permission on the asset.

- If you intend to give users in other organizations access to the asset and the asset includes supporting documents that you want those users to be able to view, make sure you give those users permission to view the supporting documents as well as the asset itself.

You can set the permissions on an asset in two ways:

- **Using the Permissions profile in the user interface**

You can use the **Permissions** action in CentraSite Business UI.

- **Using the Set Instance and Profile Permissions policy action**

You can use the **Set Instance and Profile Permissions** policy action in a design/change-time policy to automatically assign permissions to an asset during any of the following events:

- PostCreate
- PreStateChange
- PostStateChange
- OnTrigger

Restricting Access to Summary Profile

CentraSite allows you to set permissions on individual profiles within an asset. This feature enables you to specify which of the available profiles can be viewed or edited by users when they display the asset in CentraSite Business UI. For any given asset, you can define different profile permissions for different users. For example, if an asset includes a profile called Source Control that displays links to your source control systems, you might want to restrict the visibility of that profile to authorized developers.

You define the user-specific or group-specific profile permissions of an asset using the asset's **Permissions** action. For procedures, see:

- [Setting Instance Level Permissions on Asset](#)

■ Setting Instance Level Profile Permissions on Asset

The profile permissions that can be set on a given asset for any user or group are:

Permission	Description
View	Enables the specified user or group to see the profile when they view the asset.
Modify	Enables the specified user or group to modify the attribute settings in the profile when they view the asset.

Note that the individual profiles do not include the Full permission because users cannot delete a profile from an individual asset.

Setting Instance Level Permissions on Asset

➤ To assign instance-level permissions on an asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Click the asset whose permissions you want to modify.

This opens the Asset Details page. Also, the actions bar displays a set of actions that are available for working with the displayed asset.

3. On the actions bar of the Asset Details page, click **Permissions**.

Note:

If you do not see the **Permissions** action, it is probably because you do not have the Modify permission on that asset.

4. In the **Assign Permissions** dialog box, select the users or groups to which you want to assign permissions.

- To select the users or groups using the typeahead option, see [Selecting Users or Groups Using Search Option](#).
- To select the users or groups using the browse option, see [Selecting Users or Groups Using Browse Option](#).

5. Use the View, Modify, and Full check boxes to assign specific permissions to each user and/or group in the **User/Group Permissions** list as follows:

Permission	Allows the selected user or group to...
 View	View the asset.
 Modify	View and edit the asset.
 Full	View, edit, and delete the asset. This permission also allows the selected user or group to assign instance-level permissions to the asset.

- When you assign instance-level permissions on an asset, the related objects (for example, bindings, operations, interfaces, and so on) receive the same permissions that are assigned on the asset.
- If you want to ensure that the asset's dependent assets (for example, a WSDL or schema) receive the same permissions, expand the **Advanced Settings** section and mark the checkbox **Propagate permissions**. If you do not mark this checkbox, the permissions of the dependent assets will not be modified.

In addition, you can ensure that the dependent assets of the same object type receive the same profile permissions. To do this, mark the checkbox **Propagate profile permissions**.

- If at any time, you wish to remove one or more users' or groups' permissions, click the **Delete** icon.
- Click **Ok**.
- Click **Save**.

Setting Instance Level Profile Permissions on Asset

> To assign instance-level permissions on an asset's profiles

- In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

- Click the asset whose permissions you want to modify.

This opens the Asset Details page. Also, the actions bar displays a set of actions that are available for working with the displayed asset.

- On the actions bar of the Asset Details page, click **Permissions**.

Note:

If you do not see the **Permissions** action, it is probably because you do not have the Modify permission on that asset.

4. Locate the user or group for which you wish to set profile permissions. Then click the chevron next to the user or group name to open the profile permission list.
5. Use the check boxes to indicate which profiles the user or group is permitted to view or modify.
6. Click **Ok**.
7. Click **Save**.

Propagation of Permissions

An asset can have one or more dependent objects. For example, a Service asset can refer to a WSDL which in turn can refer to one or more XML Schema assets. You can optionally choose whether the permissions assigned to an asset instance should be automatically propagated to the asset instance's dependent objects.

In the context of CentraSite Business UI, propagation of permissions means that the new permissions completely replace the old permissions; the new permissions are not merged with the old permissions. As an alternative, you can use a change-time policy containing the action Set Instance and Profile Permissions. With this action, you can choose whether the new permissions will be merged with the old permissions or will replace the old permissions. For details, see the *CentraSite Developer's Guide*.

Propagation of Instance Level Permissions

By default, the access level permissions that are assigned on an asset are implicitly propagated to these dependent assets. This behavior is activated when you mark the checkbox **Propagate asset permissions** in the asset's **Advanced Settings**. For example, assigning Modify permission on a Service asset propagates the Modify permission to the asset's WSDL, schemas, and so on.

If you do not have permission to assign instance-level permissions to a dependent object, the dependent object will not be modified and a warning message will be issued.

Propagation of Profile Permissions

In addition to propagating permissions that control the access to an asset instance (as described above), it is also possible to propagate permissions that control the access to the asset instance's profiles. This means that the profile permissions that you define for an asset instance can be propagated to the asset's dependent assets. However, this is only possible if the dependent object is of the same asset type as the first object; this restriction arises because different asset types can have different sets of profiles.

This behavior is activated when you mark the checkbox **Propagate profile permissions** in the asset's **Advanced Settings**.

Selecting Users or Groups Using Search Option

You can use CentraSite's typeahead search feature to search for users and groups in the user database.

When performing a search for the users and groups, keep the following points in mind:

- You must type a search string to retrieve the desired list of users or groups.
- CentraSite treats the text you enter as a partial string. For example, if you enter "ali", then "Alison", "Calient" and "Salie" all fit the search criteria.
- The search starts with the specified number of offset characters counted from the beginning of the string.
- Search strings are not case-sensitive. Example: A search for "alison" will return the same results as a search for "Alison" or "ALISON".
- Search strings are not accent-sensitive.
- When you are searching the user list, CentraSite searches the user name attribute, not the user ID attribute. Thus, if a user has the name "John Smith" and the user ID "MyDomain\AdminUser01", a search for will find the user "John", whereas a search for "Admin" will not.

> To select users or groups by typeahead search

1. In the **Add User or Group** text box, type a search string that specifies the characters contained in the user or group name.

As you enter the search string, CentraSite returns the top n assets that meet your search text.

By default, the result set is ordered alphabetically.

2. Press the Up Arrow and Down Arrow keys to scroll through one user or group at a time.
3. Locate the user or group to whom you wish to set permissions for the asset.
4. Click **Add**.
5. Repeat steps 1 to 5 until you have all the required users and/or groups.
6. Click the **Ok** button to add the chosen users and/or groups to the **User/Group Permissions**.
7. Assign specific permissions to each user and/or group as mentioned above.

Selecting Users or Groups Using Browse Option

You can use CentraSite's browse feature to search for users and groups in the user database.

When performing a search for the users and groups, keep the following points in mind:

- The **Choose** option opens to the **Choose Users and Groups** dialog. You must type a search string to retrieve the desired list of users or groups.
- CentraSite treats the text you enter as a partial string. For example, if you enter "ali", then "Alison", "Calient" and "Salie" all fit the search criteria.
- The search starts with the specified number of offset characters counted from the beginning of the string.
- Searches are not case sensitive nor accent sensitive.
- CentraSite performs a search based on the *Name* attribute.

➤ **To select users or groups by browse option**

1. In the **Add User or Group** text box, type a search string that specifies the characters contained in the user or group name.

By default, the result set is ordered alphabetically.

2. Refine the search result set by choosing one of the options from the drop-down **Sort** (*Name* or *Organization*).
3. Select the checkbox next to the name of the users or groups whom you wish to set permissions for the asset.
4. Click the **Ok** button.
5. Repeat steps 1 to 4 until you have all the required users and/or groups.
6. Click the **Ok** button to add the chosen users and/or groups to the **User/Group Permissions**.
7. Assign specific permissions to each user and/or group as mentioned above.

Publishing and Unpublishing Services to and from API Portal through API Gateway

Starting from 10.5, CentraSite allows you to publish and unpublish virtual services to and from API Portal through API Gateway .

To publish and unpublish virtual services to and from an API Portal gateway, you must have the following roles or permissions:

- CentraSite Administrator
- Organization Administrator
- API Portal Publisher
- Instance-level Modify permission for API Portal gateway

- If you have the CentraSite Administrator role, you can publish and unpublish virtual service to and from API Portals within any organization.
- If you have the Organization Administrator role or API Portal Publisher role for an organization, you have the ability to publish and unpublish virtual service to and from the API Portal within the specific organization.

Publishing a Service to API Portal gateway refers to the process you use to deploy virtual services to API Gateway and then to API Portal on which they are to be exposed for testing and user consumption.

Doing this involves the following high-level steps:

1. Make sure the virtual service is published to the API Gateway, before the user initiates the **Publish** action.
2. The user initiates the **Publish** action and specifies the API Portal to which they are to be published. The user also needs to specify the API Gateway endpoints through which the virtual service can be published to the API Portal .
3. You can only publish virtual service to the API Portal for which you have the required roles or the Modify instance-level permission.
4. CentraSite publishes the virtual service to the specified API Portal.

Publishing Service Assets to API Portal

» To publish a service asset to API Portal

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.This displays a list of assets in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** button, and follow these steps:
 - a. Click the chevron next to **Assets** button.
 - b. In the list of asset types, select any of the following types: **Virtual Service**, **Virtual REST Service**, **Virtual OData Service**.
 - c. Click **OK**.

5. Click the service you want to publish to the gateway.

This opens the Service Details page. Also, the actions bar displays a set of actions that are available for working with the displayed service.

6. On the actions bar of the Service Details page, click **Publish**.

This opens the **Publish** dialog box.

7. Select *API Portal* from **Gateway type** drop-down list.

The default is set to **API Gateway**.

8. (Optional). In the **Sandbox** list, select a sandbox category that classifies the endpoint of the service for publishing to API Portal :

The **Sandbox** list includes:

If you choose...	CentraSite displays...
Development	Instances of a specific gateway whose endpoints are classified by the Development category.
Production	Instances of a specific gateway whose endpoints are classified by the Production category.
Test	Instances of a specific gateway whose endpoints are classified by the Test category.
All Sandboxes	Instances of a specific gateway whose endpoints are classified by any of the above mentioned categories.

The default is set to **All Sandboxes**.

9. In the **Select an API Portal and communities to publish** column, select a API Portal instance and its corresponding communities to which you want to publish the virtual services.

- a. Select one or multiple communities available for a particular API Portal.

The default is set to **Public Community**. Any new community assigned to the service overwrites the existing community assignments.

10. In the **Select an API Gateway and endpoints to expose** column, select one or more API Gateway instances and its corresponding endpoints through which you want to expose the virtual services in API Portal.

Note:

The **Select an API Gateway and endpoints to expose** column displays the API Gateway instances to which you have already published the service.

11. Click **Publish**.

A **Publish Inprogress** popup displays the progress state of publishing the service to selected API Portal.

If the publish process logs failures, identify and correct the failure and then try publishing the service again.

Unpublishing Service Assets from API Portal

> To unpublish a Service asset from API Portal

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, service, click **Choose**.

4. In the **Choose Asset Types** dialog box, select the **Assets** button, and follow these steps:

- a. Click the chevron next to **Assets** button.
- b. In the list of asset types, select any of the following types: **Virtual Service, Virtual REST Service, Virtual OData Service**.
- c. Click **OK**.

5. Click the service you want to unpublish from the gateways.

This opens the Service Details page. Also, the actions bar displays a set of actions that are available for working with the displayed service.

6. On the actions bar of the Service Details page, click **Unpublish**.

This opens the **Unpublish** dialog box.

7. Select *API Portal* from **Gateway type** drop-down list.

The default is set to **API Gateway**.

This opens the **Unpublish** dialog box.

8. (Optional). In the **Sandbox** list, do the following:
- Select one or multiple sandbox categories that classify the endpoint of the services for unpublishing from API Portal gateways.

The **Sandbox** list includes:

If you choose...	CentraSite displays...
Development	Instances of a specific gateway whose endpoints are classified by the Development category.
Production	Instances of a specific gateway whose endpoints are classified by the Production category.
Test	Instances of a specific gateway whose endpoints are classified by the Test category.
All Sandboxes	Instances of a specific gateway whose endpoints are classified by any of the above mentioned categories.

The default is set to **All Sandboxes**.

9. In the **Gateway** list, select one or more API Portal instances from which you want to unpublish the virtual services.

The **Gateway** list only displays the gateways for which you have the Modify permission.

10. Click **Unpublish**.

The **Unpublish Inprogress** popup displays the progress state of unpublishing the service from the selected API Portals.

If the unpublish process logs failures, identify, and correct the failure, and try unpublishing the service again.

Legacy Portal Publish

Starting from 10.5, CentraSite allows you to publish mediator published virtual services to API Portal using the **Legacy Portal Publish** action.

Note:

You cannot publish a native service through the **Legacy Portal Publish** action.

Doing this involves the following high-level steps:

- Make sure the virtual service is published to the Mediator, before you initiate the **Legacy Publish** process.
- The user initiates the **Legacy Portal Publish** action and specifies the API Portal to which they are to be published.

3. CentraSite publishes the virtual services to each of the specified API Portals.
4. The publishing process continues even if CentraSite encounters a failure with one of the API Portal.
5. CentraSite logs information about the virtual services that fail at the end of the publish process. If the process of publishing returns specific data about the Service, its metadata is updated as required in the API Portal registry.

Before you execute a **Legacy Portal Publish** action through the Service Details page, you have to publish the virtual service asset to Mediator gateway. Only then **Legacy Portal Publish** icon appears on the action bar of the Service Details page. Then you can publish the virtual service to API Portal.

➤ **To perform a legacy portal publish**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.This displays a list of assets in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** button, and follow these steps:
 - a. Click the chevron next to **Assets** button.
 - b. In the list of asset types, select any of the following types: **Virtual Service, Virtual REST Service, Virtual OData Service**.
 - c. Click **OK**.
5. Click the service you want to publish to the gateways.

This opens the Service Details page. Also, the actions bar displays a set of actions that are available for working with the displayed service.
6. On the actions bar of the Service Details page, click **Publish**.

This opens the **Publish** dialog box.
7. Select *Mediator* from **Gateway type** drop-down list.

The default is set to **API Gateway**.

8. (Optional). In the **Sandbox** list, select a sandbox categories that classify the endpoint of the service for publishing to Mediator gateways:

The **Sandbox** list includes:

If you choose...	CentraSite displays...
Development	Instances of a specific gateway whose endpoints are classified by the Development category.
Production	Instances of a specific gateway whose endpoints are classified by the Production category.
Test	Instances of a specific gateway whose endpoints are classified by the Test category.
All Sandboxes	Instances of a specific gateway whose endpoints are classified by any of the above mentioned categories.

The default is set to **All Sandboxes**.

This lists Mediator instances and its corresponding sandbox.

9. In the **Name** column, select one or more Mediator instances to which you want to publish the virtual services.
10. Click **Publish**.

A **Publish Inprogress** popup displays the progress state of publishing the service to selected Mediator gateways.

If the publish process logs failures, identify and correct the failure and then try publishing the service again.

Note:

Once the service gets published to the Mediators , the **Legacy Portal Publish** icon displays on the action bar of the Service Details page.

11. Click **Legacy Portal Publish**

This lists API Portal gateways and its corresponding sandbox.

12. In the **Name** column, select one or more API Portal instances to which you want to publish the virtual services.
13. Click **Publish**.

A **Publish Inprogress** popup displays the progress state of publishing the service to selected API Portals.

If the publish process logs failures, identify and correct the failure and then try publishing the service again.

Versioning an Asset

To version an asset, you must have the Manage Assets permission for the organization to which the asset belongs.

You can use the versioning feature in CentraSite to add an updated version of an asset to the catalog. For example, if you make significant changes to a Service asset (such as adding operations to the service or modifying the data types that it uses), you can use the versioning feature to add the new version of the service to the catalog.

Versioning can be active or inactive for any given asset type. The method for activating versioning for an asset type is included in the CentraSite. Note also the restrictions for activating versioning, described in [“Considerations for Asset Types of the Suite” on page 536](#).

When you generate a new version of an asset, CentraSite adds a new asset of the same type to the catalog. The new asset will have the same name and description as the one from which it was versioned. It will have an updated version number. The new version is related to the old version by a Supersedes association from the new version to the old version. In cases where the detail page of an asset has a **Summary** profile, the association is displayed under the **Summary** profile.

Note:

Depending on the type of asset you version, some of the attributes are cloned from the original asset and others are not. For example, when you version a Web service, the settings on the **Classifications** profile are cloned. After you version an asset, you should always examine the attribute settings for the new version and set them appropriately.

The metrics and event information that was collected for the old version of the asset will remain unchanged in the registry/repository. The old version's metrics and event information will not be copied to the new version. CentraSite will begin collecting metrics and event information for the new version of the asset.

CentraSite maintains two sets of version numbers for an asset. One set is maintained for CentraSite's own internal use. CentraSite automatically assigns this version number when you create a new version of an asset. You cannot modify it. The version numbers assigned by CentraSite have the format *<MajorVersion>.<Revision>* and are always sequentially numbered starting from 1.0 (for example, 1.0, 2.0, 3.0). If the revision feature is enabled, the revision number is incremented automatically each time you modify the current version of the asset. For detailed information, see [“Managing Asset Revisions” on page 601](#).

Each version of an asset also has a separate user-defined version identifier. This is the public version number that CentraSite shows to users when it displays the catalog. The user-defined version identifier does not need to be numeric. For example, you might use a value such as V2.a (beta) to identify a version.

Creating New Version of Asset

When you version an asset, you become the owner of the new version of the asset. Ownership is not carried forward from the previous version.

The new version of the asset will belong to the same organization as its previous version.

➤ **To create new version of an asset**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Click the asset you want to create a new version.

This opens the Asset Details page. Also, the actions bar displays a set of actions that are available for working with the displayed asset.

3. On the actions bar of the asset, click **New Version**.

4. In the **Add Version** dialog box, provide the required details as follows:

Field	Description
Namespace	<p>The namespace associated with this new version.</p> <p>This is of specific relevance for web service assets. The namespace given here reflects the target namespace defined in the associated WSDL file. A change of the namespace can be a differentiating factor between versions. Note that if you supply a new namespace here, you should ensure that the WSDL associated with this asset also reflects the new namespace.</p>
New User Version	<p>An identifier for the new version. You can use any versioning scheme you choose. The version identifier does not need to be numeric.</p> <p>Examples:</p> <pre>0.0a 1.0.0 (beta) Pre-release 001 V1-2007.04.30</pre>
Organization	<p>Specify the organization to which this new version will be added.</p> <p>Note: The Organization list contains the names of all organizations for which you have Manage Assets permission.</p> <p>Important: Choose the organization with care. You cannot change the organization assignment after the service is versioned. You can, however, export a versioned service from one organization and import it to another.</p>

Field	Description
Change Log	<i>Optional.</i> A comment or other descriptive information about the new version.
Propagate versions to dependent objects	<i>(CentraSite only processes this checkbox for assets of type Service.)</i> Mark this checkbox if you wish to automatically create new versions of all of the service's dependent schemas. The schemas will only be updated if you have permissions to modify them.

5. Click **OK**.
6. Click **Save**.

Locating Other Versions of Asset

The **Basic Information** profile for an asset displays the list of all the asset's versions. To locate other versions of an asset, simply display the asset and examine its **Basic Information** profile.

➤ To locate other versions of an asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.This displays a list of assets in the Search Results page.
2. Click the asset to view a list of its versions.
This opens the Asset Details page.
3. To display the details for one of the listed versions, click the name of that version.

Considerations for Asset Types of the Suite

If you are using CentraSite in conjunction with other components of the webMethods Suite, the versioning capability for the asset types defined by these components is by default not activated. Unless the documentation for the webMethods product suite components states otherwise, do not activate the versioning for these asset types.

Changing Ownership of Asset

To change the ownership of an asset, you must belong to the CentraSite Administrator role.

In CentraSite, there are two concepts of ownership. An asset belongs to a particular *user* (known as the asset's *owner*) and it also belongs to a particular *organization*. The owner of an asset has

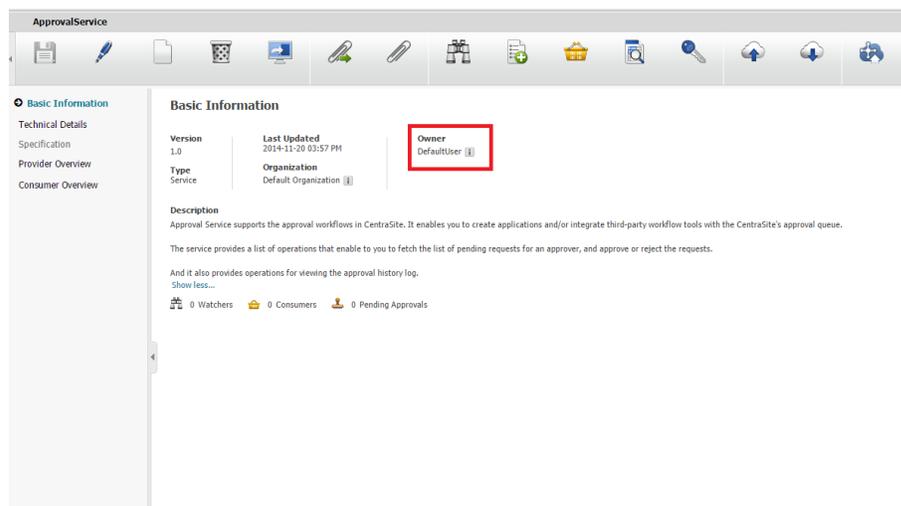
special access rights to the asset and serves as the asset's main point of contact. The asset's organization determines whose rules of governance apply to the asset.

After an asset is created, it is sometimes necessary to change its ownership. For example:

- You may want to transfer an asset to another user if the original owner leaves the company, transfers to another position, or is otherwise unable to continue serving as the owner of an asset.
- You may want to transfer ownership of an asset to another organization when the asset reaches a point in its lifecycle where it is managed by a different group of users. When a service moves into production, for example, you might want to transfer it to your operations organization.

User Ownership

The user who adds an asset to the catalog automatically becomes the asset's owner. User ownership is specified by the asset's **Owner** attribute, which appears on the details page in CentraSite Business UI.



The owner of an asset automatically receives Full permission on the asset. The owner also participates in various processes and policies that affect the asset. For example, the owner of an asset is responsible for reviewing and approving all consumer-registration requests that users submit against the asset.

When you change ownership of an asset, you transfer all of the permissions and responsibilities associated with ownership of the asset to another user.

Note:

Certain predefined assets that are installed with CentraSite are owned by an internal user known as the *default user*. You cannot transfer assets to or from this user.

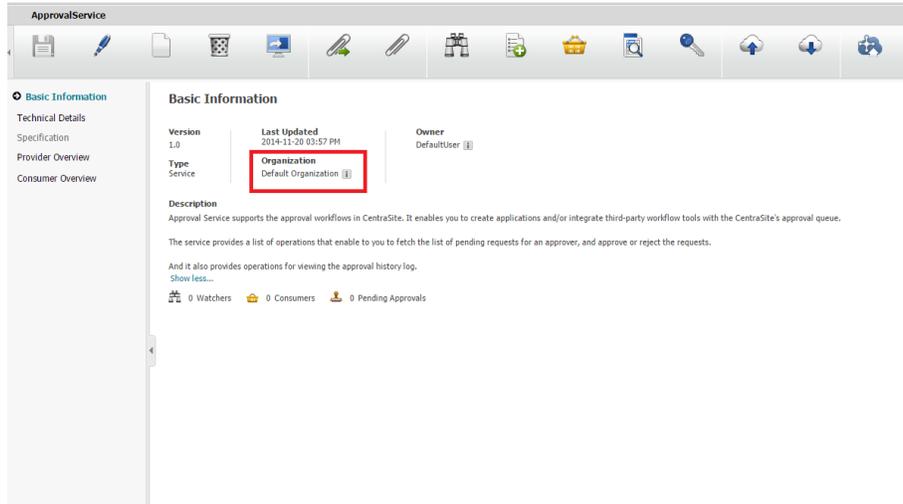
Organizational Ownership

The organizational ownership for an asset is specified by the asset's **Organization** attribute. The organization to which an asset belongs determines which policies apply to the asset, which lifecycle model it follows, and which group of users have implicit permission to view the asset. In other words, it determines whose rules of governance apply to the asset. Consequently, when you change

an asset's organizational ownership, you are in effect placing the asset under the governance of a different organization.

An asset's **Organization** attribute is specified when a user adds the asset to the catalog. Users can add assets to any organization for which they have Create Assets permission. (Most users only have permission to create assets in their own organization, so most assets in the registry belong to the same organization as their owner.)

The organization to which an asset belongs is shown in the **Organization** attribute on the asset's details page.



When changing the ownership of an asset, keep the following points in mind:

- The asset must not belong to the default user (nor can you move an asset to the default user).
- The asset must not be in a pending state (for example, awaiting approval) or have a consumer registration request pending for it.
- The asset must not be a component of a composite asset. If an asset is a component of another asset, you can move it only by moving the root asset to which it belongs.
- The asset must not be an instance of the asset type: REST Service, OData Service, Virtual REST Service, Virtual OData Service.
- The asset cannot be moved to an inactive user.

This section provides procedures for transferring assets to a different user or a different organization.

Note:

If you want to transfer an asset to a different user and a different organization at the same time, use the procedures for changing user ownership. These procedures allow you to optionally change the asset's organization in addition to its owner.

Changing User Ownership of Asset

To transfer asset ownership of a user, you must have the Manage Organizations permission or at least the Manage Users permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

> To change the user ownership of assets

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Click the asset whose ownership you want to change.

This opens the Asset Details page. Also, the actions bar displays a set of actions that are available for working with the displayed asset.

3. On the actions bar of the Asset Details page, click **Edit**.

4. In the **Basic Information** profile, locate the **Owner** attribute.

5. In the adjacent text box, type a partial string to search for the user. You can use one or more wildcards to specify the user.

6. Select the user to whom you want to transfer ownership of the asset.

7. Click **Save**.

8. When you are prompted to confirm the save operation, click **Yes**.

Changing Organization Ownership of Asset

To transfer asset ownership of an organization, you must have the Manage Organizations permission.

> To change the organization ownership of assets

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.

- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Click the asset whose ownership you want to change.

This opens the Asset Details page. Also, the actions bar displays a set of actions that are available for working with the displayed asset.

3. On the actions bar of the asset, click **Edit**.
4. In the **Basic Information** profile, locate the **Organization** attribute.
5. In the adjacent text box, type a partial string to search for the organization. You can use one or more wildcards to specify the organization.
6. Select the organization to which you want to transfer ownership of the asset.
7. Click **Save**.
8. When you are prompted to confirm the save operation, click **Yes**.

Watching and Unwatching Asset

The watch feature enables you to watch information on an asset. When you watch an asset, you receive a notification when changes are made to that asset's information. You can also unwatch changes that you have been watching.

When there is a change in the asset's information, CentraSite will automatically send notification of the changes through an email notification and/or **Inbox** notification (depending on how notification is configured in the User Preferences page).

Assets can be watched (added to your watch list) and unwatched (removed from your watch list). However, you can only add or remove assets to or from your own watch list.

When watching or unwatching an asset, keep the following points in mind:

- If you are not the owner of the asset, you cannot use the **Watch** or **Unwatch** feature unless you have Modify permission on the asset (granted through either a role-based permission or an instance-level permission).
- If you do not have the notification selected in the User Preferences page, the **Watch** or **Unwatch** icon will not be visible in the user interface (this functionality will still be available through the API).
- When you attempt to watch an asset for the first time, the **Watch** icon will be displayed in the user interface. However, if you are already watching the asset, then the **Unwatch** icon will be displayed.

Note:

The Watch and Unwatch functionality are not available to Guest users.

Watching Asset

➤ To watch an asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Click the asset you want to watch.

This opens the Asset Details page. Also, the actions bar displays a set of actions that are available for working with the displayed asset.

3. On the actions bar of the Asset Details page, click **Watch**.

CentraSite automatically changes the watcher count in the asset's **Basic Information** profile.

Unwatching Asset

➤ To unwatch an asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Click the asset you want to stop watching.

This opens the Asset Details page. Also, the actions bar displays a set of actions that are available for working with the displayed asset.

3. On the actions bar of the Asset Details page, click **Unwatch**.

If you still are still able to see the **Watch** action, it is probably because you are not registered to receive notifications for the selected asset.

CentraSite automatically changes the watcher count in the asset's **Basic Information** profile.

Viewing List of Watchers for Asset

The watch list shows the number of users who are watching the asset.

The number of users who are watching the asset is displayed with icons (representing the *Watchers*) in the description area of the **Basic Information** profile in the asset details page, for example, 8 watchers. When you are watching the asset, this is displayed as You and 8 Watchers. If no users are watching the asset, this is displayed as 0 Watchers.

Clicking on this watch list displays the basic information about the watchers.

Downloading Asset

CentraSite Business UI offers two methods of retrieving the source files of CentraSite assets, namely exporting and downloading. The source file is the file that was imported into CentraSite in order to create the registry entry for the asset. For example, the source file for a web service asset is the service's WSDL file. The source file for an XML schema asset is its schema file. The difference between exporting and downloading is as follows:

- The *export* feature creates a zip file containing one or more assets from the repository, as well as all associated registry objects.
- The *download* feature creates a zip file containing just the source file of a single asset from the repository, without any of the associated registry objects. If the source file refers to other source files in the repository (for example, a WSDL file can reference XML schema files), the referenced files will also be included in the zip file. If the asset refers to files in the Supporting Document Library, these can optionally be included in the zip file.

If an asset was not created by an importer, but was instead created from scratch without using a source file, the download feature can still be activated. In this case, however, the downloaded zip file does not contain an asset source file but instead only contains files from the Supporting Document Library that are attached to the asset.

When downloading an asset, keep the following points in mind:

- To download any given asset, you must belong to a role that has the Manage Assets permission for the organization in which the asset resides.
- If you use the download feature to create a zip file, it contains only the files that you have permission to view. The default location to which the zip file is downloaded is My Documents\Downloads.
- *This is of specific relevance to REST and XML based service assets.* Beginning version 9.7, CentraSite supports the enhanced interface for REST services (in contrast, earlier versions of CentraSite supported a standardized interface for REST services). Documentation of the prior REST and XML service interface is available to CentraSite customers who have a current maintenance contract in CentraSite, CentraSite's global extranet (<http://empower.softwareag.com/>).
- If you have REST services that were created prior to version 9.7 or if you are using the current version of CentraSite Business UI, you will only be able to view details of these services in CentraSite Control. You cannot download schemas from REST services using the CentraSite Control user interface (not even if you belong to the CentraSite Administrator role). This is

because; CentraSite Control does not support the enhanced REST interface. You will only be able to download schemas from REST services using the CentraSite Business UI.

Performing Zip Download

The asset that you want to download must belong to an asset type for which there is an importer. The importer can be either one of the predefined importers or a user-defined importer.

When attaching documents to an asset, keep the following points in mind:

- If you use the download feature to create a zip file, it contains only the files that you have permission to view. The default location to which the zip file is downloaded is `My Documents\Downloads`.
- The asset that you want to download must belong to an asset type for which there is an importer. The importer can be either one of the predefined importers or a user-defined importer.

➤ To download asset and its associated files

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Click the asset you want to download.

This opens the Asset Details page. Also, the actions bar displays a set of actions that are available for working with the displayed asset.

3. On the actions bar of the asset, click **Download Documents**.
4. In the **Download Documents** dialog box, select **Include all Supporting Documents** to include the attached documents from Supporting Document Library.
5. Click **OK**.

This starts the creation of the zip file.

Note:

The default location to which the zip file is downloaded is `My Documents\Downloads`.

Downloading a Single Document from Supporting Document Library

➤ To download an attached document from the SDL

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Click the asset to which the supporting document is attached.

This opens the Asset Details page.

3. Select **Specification** or **Technical Details**, as required.

This displays the files that are attached to the asset.

4. Click the **Download** button next to the document you want to download.

5. Specify a location in the file system to store the supporting document, and then click **OK**.

Downloading WSDL or XSD Document from Service or XML Schema Asset

➤ To download a WSDL or XSD document from Service or XML Schema asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Click the asset which the supporting document is attached.

This opens the Asset Details page.

3. Select **Technical Details**.

4. Click on the WSDL / URL hyperlink.

5. Click the **Download** button next to the document you want to download.

6. Specify a location in the file system to store the supporting document, and then click **OK**.

If the WSDL or schema file includes a reference to another file (usually a relative address) in the repository, then this reference will be changed to an absolute repository address.

Downloading XSD document from REST or XML Service Asset

➤ To download a XSD document from REST or XML Service Asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Click the asset you which the supporting document is attached.

This opens the Asset Details page.

3. Select **Technical Details**.

4. Click the **Download** button next to **Schema Name** of the document you want to download.

5. Specify a location in the file system to store the supporting document, and then click **OK**.

If the XSD file includes a reference to another file (usually a relative address) in the repository, then this reference will be changed to an absolute repository address.

Exporting and Importing Assets

You use the export and import features to export an asset from one instance of CentraSite and import it into another.

Before you export and import an asset in the CentraSite Business UI, review the following information. Software AG also recommends that you review the general information about exporting and importing registry objects in the *CentraSite Administrator's Guide*.

- If the asset or any referenced object in the archive already exists in the gateway instance of CentraSite, the existing object will be overwritten.
- To export an asset, you must have the instance-level View permission on the asset.
- To import an asset successfully, you must belong to a role that includes the Manage Assets permission for the organization in which the selected asset resides.
- The export process does not export the selected asset's instance-level permissions.
- The export operation creates an archive file on the file system. The archive file contains a copy of the assets that you have exported. The archive file can be imported afterwards into the same CentraSite registry or into a new registry.
- When you import the asset on the gateway instance, CentraSite assigns instance-level permissions to the imported asset just as though you created the asset manually. (In other words, the imported asset receives the same permission settings as the assets you create from scratch.)
- When an imported asset *replaces* (updates) an existing asset in the target registry, all of the asset's properties, except for its permission settings, are updated according to the asset object in the archive. This includes the asset's organizational scope and its lifecycle state. If the

referenced organization and/or lifecycle model does not already exist on the target registry, the import process will fail.

- If the archive file contains a reference to an object that is not already present in the target registry or is not included in the archive file itself, the asset will not be imported.
- If design/change-time policies exist for the events that the import process initiates (for example, creation of an asset), those policies will be triggered.
- The archive you wish to import must reside in the file system of the computer where your browser is running.
- *This is of specific relevance to REST and XML based service assets.* Beginning with version 9.7, CentraSite supports the enhanced interface for REST services (in contrast, earlier versions of CentraSite supported a standardized interface for REST services). Documentation of the prior REST service interface is available to Software AG customers who have a current maintenance contract in Empower, Software AG's global extranet (<https://empower.softwareag.com/>).
- Starting with this version 9.7, you cannot import a REST service implemented by current version of CentraSite to previous versions of CentraSite. This is because, CentraSite prior to version 9.7 do not support the enhanced REST interface.

If You Migrate REST Services from a Pre-9.7 Release

If you have REST services that were created prior to version 9.7, these REST services will continue to hold the old version's metadata in the enhanced REST service interface implemented by current version of CentraSite. Examine their property settings in the current version and set them appropriately.

Exporting Asset

The export operation creates an archive file on the file system. The archive file contains a copy of the assets that you have exported. The archive file can be imported afterwards into the same CentraSite registry or into a new registry.

> To export an asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. Click the asset you want to export.

This opens the Asset Details page. Also, the actions bar displays a set of actions that are available for working with the displayed asset.

3. On the actions bar of the asset, click **Export**.

The **Export** dialog box displays the selected asset.

4. Expand the **Advanced Settings** to display a list of the additional export options.
5. Specify the options as required.
6. After you have selected the export options, click **Apply Settings**.

This opens the Export Preview page. The Export Preview page displays the list of selected objects and its dependent objects. The checkbox beside each object indicates whether or not the object should be included in the export set. By default, all displayed objects are included in the export set.

To remove an object from the export set, clear the corresponding checkbox. This removes the object and all of its dependent objects (if any) from the export set.

7. Click **Export** to start the export operation.

If none of the object is selected for export, the **Export** button is disabled.

The **Export Progress** popup displays the export progress bar.

8. Click **Download** to download the export archive file.

The **Download** button is disabled until the completion of the entire export operation.

This starts the creation of the archive file. The default location to which the archive file is downloaded is My Documents\Downloads.

Importing Asset

You can import an asset by importing the archive file (zip file) to which the asset was previously exported. You can import assets into the same CentraSite registry from which they were originally exported or to a different CentraSite registry.

Predefined asset types that are exported from older versions of CentraSite may not get imported. The asset instances will get imported only if they conform to the asset type schema in the target CentraSite registry.

> To import an asset

1. In the CentraSite Business UI activity bar, click **Create Asset**.

This opens the **Create New Asset** wizard. The bottom panel of the **Create New Asset** wizard displays the option to import an archive file.

2. To import an archive file, click **Choose** to navigate to the folder where the exported archive file resides, and choose the file. When you choose a file to import, the fields in the **Basic Information** panel cannot be edited.

3. Click **Next**.
4. The **Create New Asset** wizard displays the list of objects to import. The checkbox beside each object indicates whether the object should be imported. By default, all objects displayed are included in the import set.

To exclude any object from the import set, unmark its checkbox.

If you are importing an archive file that was generated prior to CentraSite version 9.0, the wizard does not display the list of objects. However, the objects are imported.

5. Expand **Advanced Settings** to display a set of additional import options. These settings are optional.

Option	Description
Change Owner	<p>The imported assets can be assigned to the same owner as in the source CentraSite registry, or you can assign a new owner.</p> <p>The Change Owner field is type-ahead field. As you enter characters in this field, the dialog will list the usernames that match the characters you enter.</p>
Change Organization	<p>When you import assets, you can import them into the same organization in the target CentraSite registry as in the source CentraSite registry from which they were exported, or you can assign a new organization.</p> <p>The Change Organization field is type-ahead field. As you enter characters in this field, the dialog will list the organization names that match the characters you enter.</p>
Retain lifecycle state	<p>This option determines whether the lifecycle state of the imported assets is preserved. Enable the option to retain the lifecycle state of the assets being imported.</p>
Overwrite existing entities	<p>This option specifies that existing assets with the same uuid in the target CentraSite registry will be overwritten, even if the asset in the archive is older than the one in the target CentraSite registry. Enable the option to overwrite the existing assets.</p>
Import groups that the user belongs to	<p>This option determines that when you import a user, whether you want to import the groups that the user belongs to. This applies only to user-defined groups. System-defined groups are not imported. Enable the option to import the groups.</p>

6. Click **Import** to import the assets.

7. If an asset has an attribute that is required in the target CentraSite registry but not in the source CentraSite registry, CentraSite displays intermediate screens to provide values for each required attribute, before importing the asset.

This happens when an asset type definition in the source CentraSite registry is different from an asset type definition in the target CentraSite registry. For example, the asset type in the target CentraSite registry represents an updated version of the asset type with different attribute definitions.

8. When the import operation completes, the import wizard informs you if the import was successful or if there were any errors/warnings. Click **Download Import Log** to view the import logs. When you click this link, the import log lists the status of all the objects stating whether they were successfully imported or if there were errors/warnings.
9. Click **OK** to terminate the import wizard.

Managing Assets through CentraSite Control

This section describes operations you can perform to manage assets such as, web services, REST services, OData services, and Applications through CentraSite Control.

Searching and Browsing the Asset Catalog

The set of assets available to you when you search or browse the asset catalog are the assets on which you have the View permission. You can obtain View permission on an asset in the following ways:

- By belonging to a role that includes any of the following permissions.

This permission...	Allows you to...
View Assets	View all assets within a specified organization.
Modify Assets	View and edit all assets within a specified organization.
Manage Assets	View, edit, and delete all assets within a specified organization, and set instance-level permissions on those assets. This permission also allows you to create assets.
Create Assets	Add new assets to a specified organization. You automatically receive Full permission (which implies Modify and View permission) on all assets that you create.

- By having View, Modify, or Full instance-level permissions on an asset.

By default, all CentraSite users belong to the Asset Consumer role. This role includes the View Assets permission for the organization to which a user belongs.

Having the Asset Consumer role gives you implicit view permission on all the assets in your organization. You can view assets from other organizations only if you are given permission to do so through the assignment of additional role-based or instance-level permissions.

Note:

In rare instances, an administrator might not grant view permissions to all of the users in an organization. If the administrator of your organization has done this, you will need instance-level permissions on an asset in order to view it.

Using Search Metacharacters in the Keyword Search

Certain characters have a special function when used in the keyword search:

- Wildcard characters allow you to search for keywords that match a string pattern.
- The quote character (") is used to group keywords into phrases.
- To force the keyword search to treat these metacharacters as normal characters, precede the character with a backslash (\). If you want to include the backslash character itself in the search, type two backslashes.

Using Keywords

You can define the input for the keyword search in the following ways:

- A keyword search consists of 1-n search keywords. Multiple keywords are space separated. If multiple keywords are given, a logical disjunction (OR) is implied.
- A keyword is treated as partial text which can occur at the beginning of the searched strings. The `starts with` semantics are implied.

Example: If the keyword is `customer`, then the following matches are returned: `A sample svc for customers` as well as `customerservice`.

- As multiple keywords are OR combined, the keywords can match a single phrase (for example, in the description) or individual keywords can occur in different attributes.

Example: If a search is conducted for `customer service`, then `customer` could be matched in the description and `service` in an object specific attribute.

- If quotes (" ") exist around a phrase, then a search is performed on the exact phrase within the quotes. A space within a quoted phrase is considered as a space character and not as a logical operation.
- You can mix and match any number of words and quoted phrases within the keyword field.
- The search is neither case nor accent sensitive, even within a quoted phrase.

Example: A search for `abc` will return the same results as a search for `ABC` or `Abc`.

- If you type a string that contains an odd number of double-quote characters, then the last double-quote character is ignored when the search is performed.

- If the keyword search input field is empty when the search is executed in CentraSite Control, the search returns all available assets.
- The simple search can include wildcard characters.

Using Wildcards

The available wildcard characters are as follows:

Character	Usage
* or %	If you use the percent symbol (%) or the asterisk (*), CentraSite replaces the wildcard symbol with as many characters as necessary to find a match. For example, an entry of A%n returns both Amazon and American. If you type *al, then CalcService, Calendar and AustralianPostCode all fit the search criteria.
? or _	If you use the question mark (?) or the underscore (_), CentraSite replaces the wildcard symbol with a single character in order to find a match. Example: CustomerSVC?Request matches any character for ?.

You can use a wildcard character at any point in the keyword text and multiple times throughout the keyword text. If you type a wildcard character in the middle of a string, for example `cat*dog`, then at least one of the searched attributes must contain the string in order for the asset or supporting document to be included in the result set.

If a wildcard character between two words is surrounded by spaces, such as `word1 * word2`, the wildcard will match one word.

Note:

Here are some general guidelines:

- Certain non-alphanumeric characters that can appear in the name of an asset are currently ignored by CentraSite's wildcard mechanism when you include them in a keyword search. In particular, the hyphen (-) is ignored. Thus, if you have created the assets `asset-1` and `asset_1`, the wildcard search for `asset?1` will find `asset_1` but not `asset-1`.
- The percent (%) character acts as a word delimiter when it appears in the text to be searched. Thus, for example, if the description field of an asset contains the text `abc%def` (the characters a, b, c, %, d, e, f), this is treated by the search mechanism as two adjacent words `abc` and `def`. A wildcard search such as `abc*def` looks for a single word beginning with `abc` and ending with `def`, so the search will not find this asset.

Searchable Attributes

You can specify generic attributes (that is, attributes common to all asset types) and type-specific attributes as search criteria.

Generic Attributes

The generic attributes that can be used as search criteria in CentraSite Control are described in the following table:

Search Attribute	Usage
Name	<p>Use this attribute to search for assets whose name matches a specified text string.</p> <p>You can specify a substring or expression that can be combined with a <code>contains word</code> (default option), <code>starts with</code>, <code>equals</code>, or <code>not equals</code> expression. The search is neither case nor accent sensitive. If <code>starts with</code> is used, no wildcard is necessary as a postfix. If <code>contains word</code> is used, the word given is treated as a partial string with implicit wildcards. If <code>equals</code> or <code>not equals</code> is used, no wildcards are supported.</p> <p>If multiple substrings have been given the parameters are implicitly quoted. Explicit quotations and wildcards can be used and behave in the same way as for keyword searches.</p>
Description	<p>Use this attribute to search for assets whose description matches a specified text string.</p> <p>Usage is the same as for the <code>Name</code> attribute.</p>
Internal Classification	<p>Use this attribute to search for assets that are classified with the selected classification and the optional category or subtree.</p> <p>The selection of a category is optional to allow searching for all assets where a taxonomy was applied irrespective of the category. If a subtree was selected, then all categories contained in the subtree are considered for search.</p>
External Classification	<p>Use this attribute to search for assets that are classified with the selected classification and the optional category or subtree.</p> <p>Searches for all objects that are classified with the selected taxonomy and the optional value. The input of a value is optional and allows you to search for all assets where a taxonomy was applied irrespective of the category. For the value, wildcards are also allowed and the behavior is the same as for name searches.</p>
LifeCycle State	<p>Use this attribute to search for assets that are in a specified lifecycle state.</p>
Created	<p>Use this attribute to search for assets with a specified creation date.</p> <p>You can select a date and apply a <code>before/after/on/between</code> criterion. If <code>between</code> is used, a second input field allows you to specify the end date.</p> <p>The date input parameters allow year, month, and day input as well as hour and minute. Hour and minute default to 0. The data format is used as specified in the account preferences of a user (defaults to <code>yyyy-mm-dd</code>). No wildcards are supported.</p>
Modified	<p>Use this attribute to search for assets with a specified modification date.</p> <p>Usage is the same as for the <code>Creation Date</code> attribute.</p>

Search Attribute	Usage
Owner	<p>Use this attribute to search for assets belonging to a specified user.</p> <p>Select the user through a Browse selection list.</p>
Organization	<p>Use this attribute to search for assets provided by a specified organization.</p> <p>Select the organization through a Browse selection list.</p>
UDDI Key	<p>Use this attribute to search for an asset that exactly matches the given UDDI V3 key.</p> <p>If no prefix <code>uddi:</code> is given, this is implied automatically. No wildcards are supported.</p>
Object-specific property	<p>Use this attribute to search for assets that match the specified object-specific property.</p> <p>You can specify one or more of the local name, namespace, and value of the attribute. You can for example search just by local name (without value or namespace) to retrieve assets that use the same object specific attribute name.</p> <p>Wildcards are allowed for name and value, and they can be used in the same way as for keyword searches.</p>
Type-specific property	<p>Use this to search for assets, based on the values of type-specific attributes. When you select this criterion, the dialog presents two related fields: one for specifying the name of the type-specific attribute and one for specifying the value to be searched for.</p>
Association	<p>Use this attribute to search for assets that match the specified association.</p> <p>This allows you to retrieve all assets that participate in an association or as a relationship attribute. You can select an association type and select if the assets to be found are: target, source or either case. In addition you can optionally specify the asset types that are the source/target of the association. (Example: find all services that have an outgoing <uses> association to an XML Schema).</p>
Custom Condition	<p>Use this attribute to type a custom XQuery condition to be combined with other given criteria.</p>
Version	<p>Use this attribute to select required versions of assets.</p> <p>If you select <code>show all</code>, the result list contains all versions of an object in which other criteria matched. If you select <code>newest only</code>, only the latest version of an asset is displayed. If you select <code>exact match</code>, you need to type an additional parameter which is treated as version information. The search is conducted across system and user versions.</p>

Search Attribute	Usage
Revision	<p>Use this attribute to search by checkpoint label or checkpoint range (time interval).</p> <p>Use this attribute to search by revision label (user version) and revision range (time interval). You can type a string value as the revision label. For the label <code>startsWith</code>, <code>equals</code>, <code>notEquals</code> and <code>containsWord</code>, operators are available as in search by name, with the same handling of wildcards. In addition a user can select date range to retrieve all revision created in the given data interval. A date based search does not require that a label is given. As search results the revision instances are retrieved (not the objects having that revision in history).</p>
Extension Point Search	<p>Use this attribute to specify a search criterion through a user-defined pluggable UI extension.</p> <p>When you select this criterion, a field appears with a drop-down menu that shows all of the available extension points for a search. Select the required extension point.</p> <p>Click the Modify button. This invokes the user-defined Adapter (layout) screen for typing custom search-related settings.</p> <p>For information on defining an extension point for a search, see the <i>CentraSite Developer's Guide</i>.</p>

Type-Specific Attributes

In addition to the generic attributes listed in the table above, each asset type can have its own type-specific search criteria, based on the type-specific attributes of the asset type.

The type-specific attributes can be selected in two ways in CentraSite Control:

- Select the entry **Type-specific property** in the drop-down list of generic search criteria, as described in the table above.
- Select one of the additional entries in the **Criteria** list as follows:

The type-specific search criteria are shown in the **Criteria** list in the form `<AttributeName>` (`<DataType>`), where `<AttributeName>` is the name of the type-specific attribute and `<DataType>` is the data type of the attribute.

For example, if you select the asset type `Service` in the **Types** field, the **Criteria** drop-down list contains search criteria like `SOAP-Version (String)`, which refers to the service's type-specific attribute `SOAP-Version` which has the data type `String`.

Depending on the data type of the type-specific attribute you select, the **Criteria** section of the dialog changes to reflect the search possibilities for that data type.

Attribute Data Types and Supported Search Operators

Not all attribute data types support the full set of search operators. Some data types execute only with certain operators.

The following tables lists the supported search operators you can use when searching for attributes depending on their data types.

Data Type	Search Operators and Description
String	Equals
	NotEquals
	StartsWith
	Contains
International String	Equals
	NotEquals
	StartsWith
	Contains
Multiline String	Equals
	NotEquals
	StartsWith
	Contains
Email	Equals
	StartsWith
URL/URI	Equals
	StartsWith
Number	Equals
	NotEquals
	Greater
	Smaller
	GreaterorEquals
	SmallerorEquals
Boolean	N/A
Date/Time	Before
	After

Data Type	Search Operators and Description
	Between
	On
Duration	N/A
IP Address	Equals
	Between
File	Equals
	StartsWith
Classification	N/A. Assumed toEquals.
Relationship	N/A. Assumed toEquals.

Searching the Asset Catalog

CentraSite Control supports the following types of search filters:

- Keywords (*Keyword Search*)
- Modifiers (*Advanced Search*)
- Logical operators

The following sections describe how to locate assets in the catalog using these search filters.

Keyword Search

The keyword search is an easy to use search facility in which you can specify arbitrary search patterns.

You can search for all assets that contain one or more specified keywords (that is, text strings) in the asset's string attributes (Name, Description, Attributes, and so on.). The contents of the supporting documents are not searched. It is not possible to restrict the types on which the search is conducted.

By default the result set is ordered by relevance. Relevance is decided as follows:

- If the search criteria contain more than one keyword, the assets that match the most keywords are ranked higher.
- Assets where one or more search criteria match the Name or Description are rated higher than those where other attributes match.

The number of search results is displayed in brackets in the title line of the results area, for example, Assets (43). If no results are found, this is displayed as (0).

Upon switching from keyword search to XQuery search, a default XQuery statement is displayed (that is, the displayed XQuery does not correspond to the keyword search.)

➤ **To search the catalog by keyword**

1. In CentraSite Control, go to **Asset Catalog > Search**.

This opens the Asset Search page.

2. In the **Keyword** tab, type in a keyword in the text box. You can use a wildcard character at any point in the keyword text, and multiple times throughout the keyword text.

To view the complete the list of assets and supporting documents, leave the Search text box blank or type a wildcard character.

3. Click **Search**.

CentraSite displays the list of assets and supporting documents that match the specified keyword.

Advanced Search

CentraSite's advanced search capabilities allow you to build sophisticated search clauses to search for assets on the basis of asset types and attribute values. The search criteria can be combined by a logical conjunction (AND) or disjunction (OR) operation.

The number of search results is displayed in brackets in the title line of the results area, for example, Assets (43). If no results are found, this is displayed as (0).

On switching from advanced search to XQuery search the XQuery statement represented by the advanced search is displayed.

Searching the Catalog by Type

➤ **To search the catalog by asset type**

1. In CentraSite Control, go to **Asset Catalog > Search**.

This opens the Asset Search page.

2. Click the **Advanced** tab.

3. In the **Types** panel, choose an asset type from the drop-down list. This list contains the predefined and user-defined asset types currently available in CentraSite.

To specify multiple asset types, use the **Add** button to include additional rows. When you specify several asset types in this way, the search result will include assets from all of the chosen asset types.

To search across all asset types, select **All Asset Types** in the list.

You may use the chevron to expand or collapse the **Types** panel.

4. Click **Search**.

CentraSite displays assets that belong to the selected type(s) in the area labeled **Assets**.

Searching the Catalog by Attribute Value

You can further refine your search for assets by choosing one or more searchable attributes. The list of searchable attributes varies depending on asset type selection.

> To search the catalog by attribute value

1. In CentraSite Control, go to **Asset Catalog > Search**.

This opens the Asset Search page.

2. Click the **Advanced** tab.

3. In the **Types** panel, choose an asset type from the drop-down list. This list contains the predefined and user-defined asset types currently available in CentraSite.

To specify multiple asset types, use the **Add** button to include additional rows. When you specify several asset types in this way, the search result will include assets from all of the chosen asset types.

To search across all asset types, select **All Asset Types** in the drop-down list.

You may use the chevron to expand or collapse the **Types** panel.

4. In the **Criteria** section, choose a searchable attribute from the first drop-down list. Depending on the data type of a chosen attribute, subsequent fields, for example, operators, and input fields varies widely.

To specify multiple attributes, use the **Add** button to include additional rows. When you specify several attributes in this way, the search result will include assets that match all of the chosen attributes.

You may use the chevron to expand or collapse the **Criteria** panel.

5. Click **Search**.

CentraSite displays assets that belong to the selected type(s) and that match the specified attribute values in the area labeled **Assets**.

XQuery Search

You can search for assets and supporting documents by specifying an XQuery search. There are two general approaches you can use to specify an XQuery search:

- Write an XQuery from scratch. This requires a good knowledge of the XQuery language.
- Define an advanced search, then switch to the **XQuery** tab. In this case, the equivalent XQuery code is displayed and you can adapt it to your requirements.

Note:

The results are displayed in an XML format, rather than as a list of matching assets and supporting documents.

➤ **To search the catalog by XQuery**

1. In CentraSite Control, go to **Asset Catalog > Search**.

This opens the Asset Search page.

2. To base your XQuery search on an advanced search, follow these steps:

- a. Click the **Advanced** tab.
- b. In the **Types** panel, choose an asset type from the drop-down list. This list contains the predefined and user-defined asset types currently available in CentraSite.

To specify multiple asset types, use the **Add** button to include additional rows. When you specify several asset types in this way, the search result will include assets from all of the chosen asset types.

To search across all asset types, select **All Asset Types** in the drop-down list.

You may use the chevron to expand or collapse the **Types** panel.

- c. In the **Criteria** section, choose a searchable attribute from the first drop-down list. Depending on the data type of a chosen attribute, subsequent fields, for example, operators, and text box fields varies widely.

To specify multiple attributes, use the **Add** button to include additional rows. When you specify several attributes in this way, the search result will include assets that match all of the chosen attributes.

You may use the chevron to expand or collapse the **Criteria** panel.

- d. Click **Save**.
3. Click the **XQuery** tab.
4. Type an XQuery code or modify an XQuery code derived from existing search criteria in the accompanying text box.

5. Click **Search**.

CentraSite displays assets that match the specified XQuery code.

The search results are shown as a single XML document containing assets that match the XQuery definition.

Combining Search Criteria

You can specify in which way the search criteria should be combined:

> To specify how the search criteria should be combined

1. In CentraSite Control, go to **Asset Catalog > Search**.

This opens the Asset Search page.

2. Click the **Advanced** tab.
3. After specifying your required search criteria, from the **Search Uses** box, choose one of the following:
 - To specify that an asset must meet all criteria to be considered a match, select **AND**.
 - To specify that an asset must meet at least one of the criteria to be considered a match, select **OR**.

Saving and Re-Executing Searches

After you have defined a search in the CentraSite Control, you might want to save the search definition, so that you can execute the same search again at a later stage.

Saving a Search Definition

> To save a search definition

1. In CentraSite Control, go to **Asset Catalog > Search**.

This opens the Asset Search page.

2. Specify a keyword search (in the **Keyword** tab), an advanced search (in the **Advanced** tab), or an XQuery search (in the **XQuery** tab), as described earlier in this section.
3. Click **Save**.

This opens the **Save Search** dialog.

4. In the **Saved Search Name** text box, type a name for the new search, and then click **OK**.

The saved search is displayed with this name in the **My Favorites** list.

Re-executing a Saved Search

Note:

You are allowed to modify or execute only those saved searches which you defined in the CentraSite Control.

> To re-execute a saved search

1. In CentraSite Control, go to **Home > Welcome**.

This opens the Welcome to CentraSite page.

2. In the **Application Shortcuts** menu, click **My Favorites**.
3. From the **My Favorites** menu, click **Assets I Consume** or **Assets I Provide**. This starts the search directly.

CentraSite displays assets in the **Assets I Consume** or **Assets I Provide** section.

Redefining an Existing Saved Search

You can redefine an existing saved search to suit your new requirements. However, you cannot overwrite the existing saved search with the updated search criteria.

> To redefine an existing saved search

1. In CentraSite Control, go to **Home > Welcome**.

This opens the Welcome to CentraSite page.

2. Click **My Favorites** on the **Application Shortcuts** menu.
3. From the **My Favorites** menu, click **Assets I Consume** or **Assets I Provide**. This starts the search directly.

CentraSite displays assets in the **Assets I Consume** or **Assets I Provide** section.

4. In the **My Favorites** list, click the saved search you want to recreate, and then click **Refine**.

The Search panel displays the criteria and operators defined in the saved search.

5. Modify the fields and operators of the search criteria as required.
6. To save the updated search criteria as a saved search, click **Save**, and then type a name for the new saved search.

Browsing the Asset Catalog

With the Browse feature in CentraSite Business UI, you have the following options:

- View the complete list of assets.
- View a list of assets whose name attribute contains a certain keyword (character string).
- View a list of assets that belong to certain asset types.
- View a list of assets that have been classified according to a specific taxonomy or lifecycle model.

Important:

In CentraSite Business UI, when the **Exclude sub types from Business UI search** check box is selected in a base asset type, for example, in a Service type definition, you cannot browse for asset instances of its sub type, Virtual Service, in the Search Results page. You can only browse for asset instances of Service type. In this scenario, to browse for asset instances of the Service type, you must manually add the Service type to Search Recipe.

You can select or clear the **Exclude sub types from Business UI search** check box in the **Edit Asset Type - Advanced Settings** dialog of the base type definition. However, to select or clear the selection, you must have the Manage Asset Types system-level permission.

➤ To browse the catalog

- In CentraSite Business UI, click the **Browse** link in the upper-left corner of the menu bar.
CentraSite displays a list of assets for which you have View permission in the Search Results page.

Browsing the Catalog by Asset Name

The name search is an easy to use keyword-based search facility.

➤ To browse the catalog by asset's Name attribute

1. In CentraSite Control, go to **Asset Catalog > Browse**.

A list of currently defined assets for which you have View permission is displayed in the area labeled **Assets**.

You may use the chevron to expand or collapse the **Types** panel.

2. To view a list of recently updated asset types in the CentraSite registry, click **Update**.

CentraSite displays a list of predefined and user-defined custom asset types.

3. To display the list of assets based on the selected type(s), leave the Search text box blank.

4. To further refine your list of assets, type in a keyword in the Search text box. You can use a wildcard character at any point in the keyword text, and multiple times throughout the keyword text.

Assets whose Name attribute contain the specified keyword are displayed in the **Assets** panel.

Browsing the Catalog by Asset Type

You may use the **Types** panel to restrict the types on which the search is conducted. In the **Types** panel, CentraSite displays a list of the predefined asset types and user-defined custom asset types.

> To browse the catalog by asset type

1. In CentraSite Control, go to **Asset Catalog > Browse**.

A list of currently defined assets for which you have View permission is displayed in the area labeled **Assets**.

You may use the chevron to expand or collapse the **Types** panel.

2. To view a list of recently updated asset types in the CentraSite registry, click **Update**.

CentraSite displays a list of predefined and user-defined custom asset types.

3. In the **Types** panel, select one or more asset types whose asset instances you want to view.
4. To display the list of assets based on the selected type(s), leave the Search text box blank.
5. To further refine your list of assets, type in a keyword in the Search text box. You can use a wildcard character at any point in the keyword text, and multiple times throughout the keyword text.

Assets belonging to the selected type(s) and that contain the specified keyword are displayed in the **Assets** panel.

Browsing the Catalog by Taxonomy

The taxonomy search capability allows you to search assets that have been classified according to a specific taxonomy or a category within a taxonomy.

There are several generic entries in the **Browse by** dialog. These are:

- **[None]**: This lists the currently registered asset instances, according to the asset types that you have selected.
- **[By Type]**: This lists all of the currently registered object and asset instances, sorted according to object/asset type.

[By Organization]: This lists all currently registered object and asset instances, sorted according to the owning organization.

➤ **To browse the catalog by taxonomy**

1. In CentraSite Control, go to **Asset Catalog > Browse**.

A list of currently defined assets for which you have View permission is displayed in the area labeled **Assets**.

2. Click the **Browse By and Column Selection** icon in the upper-right corner of the **Assets** panel.
3. In the **Browse By and Column Selection** dialog, select one or more taxonomies that were used to classify the asset instances you want to view, and then click **OK**.
 - If you have selected a taxonomy or a taxonomy category in this dialog, assets classified by that taxonomy or taxonomy category are displayed in a simple tree structure. To expand or collapse a taxonomy category, click the chevron next to the taxonomy name.
 - If you have not selected at least one taxonomy or taxonomy category in this dialog, assets belonging to the selected asset type(s) are displayed in a list view.

Note:

CentraSite Control only displays the taxonomies that are currently browsable.

4. To display the list of assets based on the selected taxonomy categories, leave the Search text box blank.
5. To further refine your list of assets, type in a keyword in the Search text box. You can use a wildcard character at any point in the keyword text, and multiple times throughout the keyword text.

Assets classified using the selected taxonomy categories and that contain the specified keyword are displayed in the **Assets** panel.

Web Service Management

This section describes operations you can perform to manage web services through CentraSite Control.

Adding Web Service Asset to Catalog

To add assets to an organization's asset catalog, you must belong to a role that has the **Create Assets** or **Manage Assets** permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

In CentraSite Control, you can add a **Service** asset to the catalog in the following ways:

- *You can create a Web Service using an importer*, which is a utility that generates a Web Service asset from a Web Service Definition Language (WSDL) file.
- *You can create a Web Service asset from scratch*, meaning that you create the Web Service asset (and set its attributes) manually.

Importing Web Service (including Abstract Service) Asset

Before you import a Web Service asset to the catalog, you must have the WSDL file that you want to import. This file can reside on the file system of the computer where your browser is running or it can reside anywhere on the network, as long as its location is addressable through a URL.

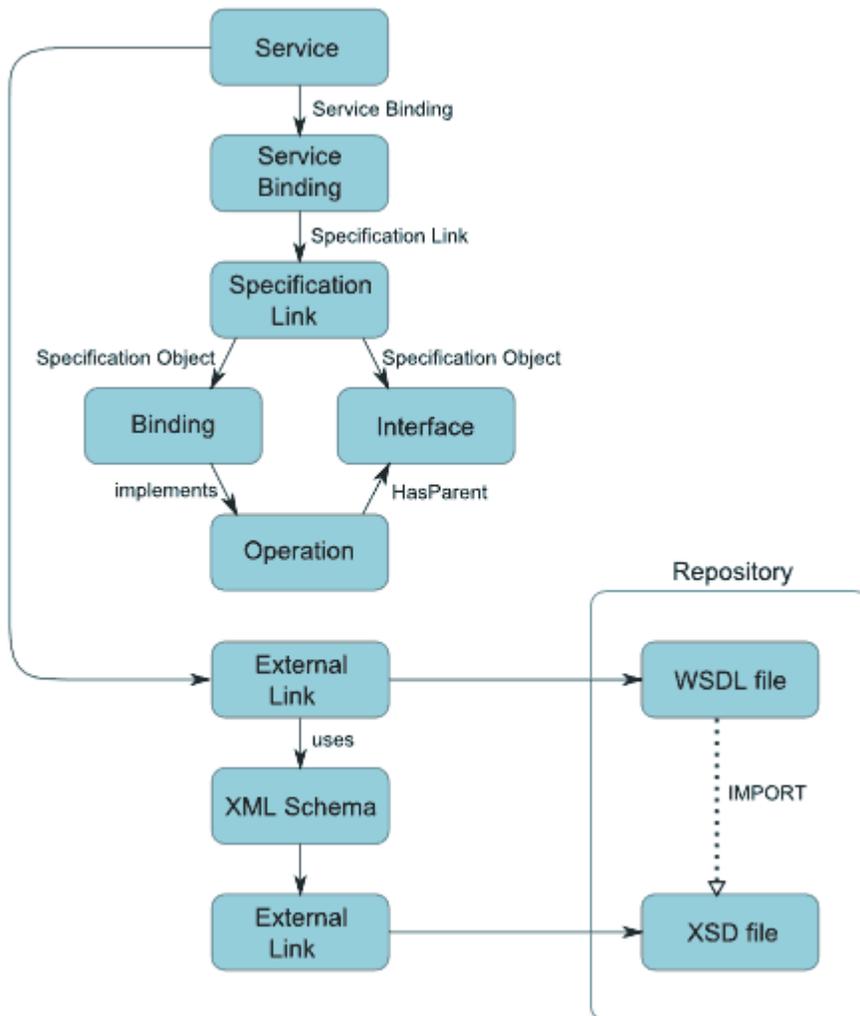
The *Web Service* importer copies a WSDL file to the repository, and creates a Web Service asset in the registry. The asset contains an External Link to the WSDL file in the repository.

The Web Service asset is represented in CentraSite by a set of related entries in the registry and in the repository. Some of the entries are visible in the detail pages of the Web Service asset, others are not displayed in the detail pages but are displayed when you use the impact analysis feature.

If you import a Web Service asset for which registry entries already exist (for example, from a previous import), the existing set of entries is extended as appropriate. Thus, for example, if you had previously imported a Web Service asset and now you import the Web Service asset again with an additional operation in the WSDL file, a new *Operation* object will be added to the existing set of entries for the Web Service asset in the registry.

Registry and Repository Entries for a Web Service Asset

The registry and repository entries created are summarized in the following diagram:



The registry entries are as follows:

- An asset of type *Service*.
- An object of type *Service Binding*, representing the binding defined in the WSDL file. This object contains the access URI of the launchable Web Service. The Web Service asset contains an internal reference to the service binding, that is, there is no explicit association.
- An object of type *Specification Link*. The service binding object contains an internal reference to the specification link, that is, there is no explicit association.
- An object of type *Binding*. The *Service* object contains an internal reference to the binding, that is, there is no explicit association object.
- One or more objects of the type *Operation*. The operation objects represent the operations of the Web Service, as defined in its WSDL. For each operation, there is an association link *Implements* from the binding object and an association link *HasParent* to the interface object.
- An object of type *Interface*. This object contains the port type elements of the WSDL definition.

- An object of the type `External Link`. This object contains a link to the Service's WSDL file stored in the CentraSite repository.

The repository entries are as follows:

- The WSDL file of the Web Service.

A WSDL definition can be spread across several physical files that contain `IMPORT` statements to refer to each other. Also, a WSDL definition can contain an XML schema definition, which can also be spread across several physical files. In such cases, each of the WSDL and schema files is stored in the repository, and appropriate references are created in the registry as follows:

- An `External Link` object in the registry refers to the top-level WSDL file of the Web Service in the repository, as described above.
- Each WSDL file in the repository that is referred to by a higher-level WSDL file for the same Web Service has an `External Link` object in the registry that is referred to by the `External Link` object of the higher-level WSDL file. The reference is implemented as an association of the type `uses`.
- Each schema file in the repository that is referred to by a WSDL file in the repository has an `XML Schema` object in the registry that is referred to by the `External Link` object of the higher-level WSDL file. The reference is implemented as an association of the type `uses`. The `XML Schema` object in the registry contains an `External Link` to the schema file in the repository.
- Each schema file in the repository that is referred to by a higher-level schema file in the repository has an `XML Schema` object in the registry that is referred to by the `XML Schema` object of the higher-level schema file. The reference is implemented as an association of the type `uses`. The `XML Schema` object in the registry contains an `External Link` to the schema file in the repository.

If the importer detects an equivalent Web Service within CentraSite during the import of a Web Service, the import dialog will prompt you to specify either **Overwrite latest version** or **Create new version**. If you decide to create a new version, the importing WSDL will be applied to the new version.

Importing Abstract Services

You can use abstract services to support a top-down service development. An abstract service asset just contains abstract definitions like interface, operation, or message definitions. But, in contrast to a normal service asset, it does not contain the complete information that is necessary to call the Web Service that it represents. This means the required definitions are missing or not complete. You can supply the concrete definitions of an abstract service at a later time in order to turn the abstract service into a normal service. An abstract service is represented in the CentraSite registry by a normal service asset.

You can define abstract services by creating them from scratch or by importing WSDLs that comply with the WSDL 1.1 specification.

According to the WSDL 1.1 specification, a WSDL file does not need to contain any service or binding element. This kind of WSDL is called an abstract WSDL. Importing an abstract WSDL to CentraSite results in an abstract service. The name and the targetNamespace of the abstract service

are taken from the `name` and the `targetNamespace` attributes of the WSDL's `definition` element. If the attributes are missing, the import is rejected.

To enable the import of name-less abstract WSDLs, the import dialog in CentraSite Control allows you to specify a name.

The minimal abstract WSDL that is supported just contains a `definitions` element with a `name` and a `targetNamespace` attribute.

An abstract service does not contain concrete definitions that refer to the registry objects for the abstract definitions. Instead the abstract service is linked to the abstract definitions through a `HasParent` relationship attribute. The relationship attribute points from the Interface objects to the `Service` object. Operation objects are not referred to directly since they are always part of an Interface. Types or messages do not need to be considered, since they are not represented in the registry. The `HasParent` relationship is classified as an aggregation relationship to ensure that the abstract definitions are considered properly when deleting, moving, or exporting the abstract service. The abstract WSDL is linked to the abstract service through an `externalLink`.

You can update an abstract service's definitions by attaching a WSDL to it. Attaching a WSDL overwrites all the registry objects that can be defined through a WSDL. This ensures that the attached WSDL reflects correctly the registry objects representing the service and its components. This also affects the service object itself. Objects that cannot be added by attaching a WSDL are not overwritten. The service object's name is not overwritten automatically when you attach a WSDL. This makes the name provided by the WSDL a technical name of a service. By default, the technical name is reflected by the local-name classification of the service object. You can align the service name with the technical name by marking the appropriate check box in the **Attach WSDL** dialog.

To show the technical name in the CentraSite Control, a computed read-only attribute is needed.

Restrictions for Abstract Services

The following restrictions apply for abstract services:

- WSDL 2.0 is not supported for importing abstract services.
- An abstract service cannot be virtualized.

Additionally, when importing a WSDL file that refers to XML schema, keep the following points in mind:

- When you have at least a `Modify` permission on the referenced XML schema, both the Web Service and the XML schema are imported automatically.
- When you have only `View` permission on the referenced XML schema, the Web Service is imported and the XML schema implicitly reused.
- When you do not have any permission on the referenced XML schema, the Web Service is imported and a warning message logged.

Tip:

If you have previously imported a WSDL that has an associated schema file and you now re-import the same WSDL with a modified schema file, your browser might not display the

updated contents of the schema file when you click on the external link for the schema file. This can happen if the browser cache is not being updated automatically. To rectify the problem, you can change your browser settings so that pages are always updated on every visit.

➤ To import a Web Service asset to the catalog

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. Click the **Import** icon that is located in the upper-right corner of the **Assets** pane.
3. In the **Import** dialog, type the appropriate information for each of the displayed data fields.

Field	Description
Organization	The organization to which you want to add the new Web Service asset. (The Organization list only displays organizations for which you have the Manage Assets permission or at least the Create Assets permission.) If you select an organization other than your own organization, you will nevertheless be the owner of the asset.

Important:

Select the organization carefully. You cannot change the organization assignment later. You can, however, export a Web Service asset from one organization and import it to another.

Import as The asset type, **Service**.

Initial Version *Optional.* An identifier for the initial version of the Web Service. This is the user-defined version, as opposed to the automatically assigned system version. You can enter any string in this field, that is, the version identifier does not need to be numeric. You can also leave the field blank. You can later create new versions of the asset.

If the versioning feature is disabled for the **Service** asset type, the field is nevertheless displayed, thus allowing you to assign an identifier for this first version.

If the import of the Web Service also causes other related objects to be imported (for example, if the WSDL definition for a Web Service includes references to other WSDL or schema definition files), the initial version is only assigned to the main asset identified in this dialog, and the initial version of the other imported objects is not assigned.

Name *Optional.* Name of the Web Service.

Note:

This is the name that users will see when they view this Web Service asset in the CentraSite User Interfaces. Therefore, we recommend that you specify a meaningful name for each Web Service.

Field	Description						
	<p>A Web Service asset name can contain any characters (including spaces), and must be unique within an organization.</p> <p>If you do not specify a name, the name is set automatically with the value that is specified for the name attribute in the <code>definitions</code> element of the WSDL. If the name attribute is not specified in the WSDL, a dialog opens in which you can specify the name.</p>						
Import from	<p>The input WSDL file for the Web Service. You may want to read the input WSDL file from a URL-addressable location on the network (the URL option) or from your local file system (the File option).</p> <table><thead><tr><th>Option</th><th>Description</th></tr></thead><tbody><tr><td>URL</td><td>If the WSDL file you are importing resides on the network, specify its URL.</td></tr><tr><td>File</td><td>If the WSDL file resides in your local file system, specify the file name. You can use the Browse button to navigate to the required folder.</td></tr></tbody></table>	Option	Description	URL	If the WSDL file you are importing resides on the network, specify its URL.	File	If the WSDL file resides in your local file system, specify the file name. You can use the Browse button to navigate to the required folder.
Option	Description						
URL	If the WSDL file you are importing resides on the network, specify its URL.						
File	If the WSDL file resides in your local file system, specify the file name. You can use the Browse button to navigate to the required folder.						
URL Authentication	<p>If you have specified a URL, and the site you want to access through the URL requires user authentication, select the URL Authentication check box. This opens the Authentication sub-dialog box. Type a user name and password for authentication at the URL site.</p>						
Interactive resolution of Import/Includes	<p>This option determines how referenced WSDLs/schemas are handled when the WSDL/schema that is referred to already exists in the registry. When this option is enabled, you will be prompted during import to specify whether you want to reuse any WSDL/schema files referred to in the main file or upload new files.</p>						
Overwrite all Imports/Includes	<p>This option determines whether to overwrite the importing WSDL file with new content.</p>						

4. Click **Finish**.

CentraSite retrieves the specified file and generates the catalog entry. The details page of the Web Service asset is displayed.

What Happens When You Import a New WSDL File?

CentraSite retrieves the specified file and generates the catalog entry. If you have specified **Interactive resolution of Import/Includes**, you will be prompted to specify whether you wish to reuse any of the WSDL/schema files referred to in the main WSDL file or upload new files.

What Happens When You Import an Existing WSDL File?

If the importer detects that the Web Service you are trying to import already exists within CentraSite, the import dialog will prompt you to specify whether you want to **Overwrite latest version** or **Create new version**. If you have specified **Interactive resolution of**

Import/Includes, you will be prompted to specify whether you wish to reuse any of the WSDL/schema files referred to in the main WSDL file using **IMPORT** or **INCLUDE** statements or upload new files.

5. During import of a Web Service asset, CentraSite does not allow you to add the Web Service asset to the catalog unless you have specified all required attributes in the WSDL definition and all referenced objects to which the WSDL file has an association. The value for the required attribute must be specified in the Web Service asset's profile. Additionally, if the Web Service asset has internally referenced objects, the value for the required attributes of all such referenced objects should be specified by choosing the **Next** button that is located in the upper-right corner of the details page in order to add the Web Service asset to the asset catalog. Thus, for example, if the WSDL/XML schema files that are referenced in the main WSDL file have required attributes, then you will be prompted to specify a value for the required attributes in order to save the Web Service asset.
6. Click **Save**.

The newly created Web Service asset is added to the CentraSite Registry Repository.

7. Review the import log that CentraSite displays after the import process. If errors occur while reading and processing the file, they will be reported in this log.
8. Configure the extended attributes of the Web Service asset as described later in this topic.

Tip:

If you had previously imported a WSDL file that has an associated schema file and you now re-import just the schema file with modifications, your browser may not display the updated contents of the schema file. This can happen if the browser cache is not being updated automatically. To rectify the problem, you can change your browser settings so that pages are always updated on every visit.

Adding Web Service Asset from Scratch

You might not be able to manually set all of the attributes for a Web Service asset. Certain attributes can only be set by an importer. For example, CentraSite allows you to add a Web Service asset to the catalog from scratch, but attributes such as the list of operations and the service endpoints cannot be specified manually. To set these attributes, you must attach the WSDL file to the Web Service asset using the **Attach WSDL** command on the asset's **Actions** menu.

> To add a Web Service asset to the catalog

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. Click **Add Asset**.
3. In the **Add Asset** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Type	The asset type, Service .
Name	<i>Optional.</i> Name of the Web Service asset. Note: This is the name that users will see when they view this Web Service in the CentraSite User Interfaces. Therefore, we recommend that you specify a meaningful name for each Web Service. A Web Service asset name can contain any characters (including spaces), and must be unique within an organization.
Description	<i>Optional.</i> The description for the Web Service. Note: This is the description information that users will see when they view this Web Service in the CentraSite User Interfaces. Therefore, we recommend that you specify a meaningful description for each Web Service.
Organization	The organization to which you want to add the Web Service. (The Organization list only displays organizations for which you have the Manage Assets permission or at least the Create Assets permission.) If you select an organization other than your own organization, you will nevertheless be the owner of the asset. Important: Select the organization carefully. You cannot change the organization assignment later. You can, however, export a Web Service asset from one organization and import it to another.
Initial Version	<i>Optional.</i> An identifier for the initial version of the Web Service. This is the user-defined version, as opposed to the automatically assigned system version. You can enter any string in this field, that is, the version identifier does not need to be numeric. You can also leave the field blank. You can later create new versions of the asset. The default is 1.0. If the versioning feature is disabled for the Service asset type, the field is nevertheless displayed, thus allowing you to assign an identifier for this first version.0

4. Click **OK**.

The newly created Web Service asset is added to the CentraSite Registry Repository.

5. Configure the extended attributes of the Web Service asset.

Viewing Web Service List

You use the **Asset Catalog > Browse** to display the list of Web Service assets in CentraSite.

➤ **To view the list of Web Service assets**

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Service**.

The Web Service assets (for which you have the View permission) are displayed in the **Assets** pane.

The **Assets** pane provides the following information about each Web Service asset:

Column	Description
Name	Name of the Web Service asset.
Type	The asset's type definition, Service.
Version	The user-defined version identifier of the Web Service.

You can adjust the view to show or hide the individual column by using the **Browse By and Column Selection** icon that is located in the upper-right corner of the **Assets** pane.

The shortcut menu of a particular Web Service asset displays one or more actions that you can perform on that Web Service.

Action	Description
Copy	Copies an existing Web Service that is similar to the one you need, and allows you to examine and modify the copy. CentraSite treats the copy just as if it were a new Web Service that you created from scratch.
Details	Displays the details page of the Web Service.
Change Lifecycle State	Changes the lifecycle state of the Web Service.
Revert Pending State	Reverts a request that has been submitted for approval, and that is struck in Pending mode.
Change Owner	Changes the user ownership of the Web Service.
Run Policy	Executes a set of policies that are applicable for the asset's type definition, Service.
Change Organization	Changes the organizational ownership of the Web Service.
Export	Exports the Web Service from the registry to an archive file on the file system.

Action	Description
Impact Analysis	Helps to easily visualize the associations that exist between the Web Service asset and registry objects.
Attach WSDL	Appends a WSDL document to the Web Service.
Add New Version	Generates a new version of the Web Service.
Notify Me	Sends notifications to all the registered users when changes are made to the Web Service.
Add to List	Adds the Web Service to a list in My Favorites .
Add to Favorites	Adds a shortcut to the Web Service you want to use routinely or otherwise keep close at hand.
Virtualize	Creates a proxy for the Web Service for consumption.
Download Documents	Downloads files that are attached to the Web Service from Supporting Document Library (SDL).

Modifying Web Service Details

Each tab on a Web Service asset's details page represents a collection of attributes called a Profile. The Service asset type has a unique set of profiles. However, an administrator can configure the Service asset type to display a customized set of profiles and attributes.

You can modify the basic and type-specific attributes for a Web Service asset. You can view the bindings, operations, WSDL file, associated schema files, and the external links to the WSDL and schema files.

The following general guidelines apply when modifying the details of a Web Service asset in CentraSite Control:

- If you are not the owner of the Web Service, you cannot examine or modify the details of the Web Service, unless you have the View or Modify permission on the Web Service (granted though either a role-based permission or an instance-level permission).
- When you view the details page of a Web Service, you will only see profiles for which you have the profile-level View permission. You will only be able to modify the attributes of the profiles on which you have the Modify permission.
- Some attributes accept only specific types of information. For example, if the Service asset type includes a URL type attribute, you must supply a URL when you modify that attribute. Other attribute types that require a specific type of value include the Date and Email attributes.
- Some attributes are designed to be read-only. You cannot modify the read-only attributes even if you have the CentraSite Administrator role.

➤ [To modify the details of a Web Service asset](#)

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Service**.
3. In the **Assets** pane, right-click the Web Service for which you want to modify the attributes, and click **Details**.

The Web Service Details page is displayed.

4. To modify the generic attributes of the Web Service that are displayed in the area labeled **Basic Information**, change values of the attributes in the respective data fields as required.
5. To modify the extended attributes of the Web Service that are displayed in the individual profiles, perform the following:
 - a. Select the profile for which you want to modify the attributes.
 - b. Change values of the attributes in the respective data fields as necessary.
 - c. Repeat steps 5.a and 5.b for each profile for which you want to modify the attributes.
6. Click **Save** to save the updated changes.

Modifying an Input WSDL File

The Web Service asset includes a WSDL file and one or more associated files. You can upload a new file or update an existing file for the Web Service accordingly.

On the **Actions** menu, click **Attach WSDL**.

If you are attaching a WSDL file to the Web Service which already has a WSDL, the Web Service name in the new WSDL must be identical to the Web Service name in the existing one or the process will fail.

If you are attaching an abstract WSDL file to an abstract Service asset which already has a WSDL, the Web Service name in the new WSDL does not need to be identical to the Web Service name in the existing one. Select the **Overwrite Service name** check box to replace the name of the existing attached WSDL with the name of the new attached WSDL.

If you select the option **Interactive resolution of Imports/Includes**, and the attached WSDL contains an Import or Include reference to a WSDL that already exists in the registry, the **Attach WSDL to ...** dialog box allows you to choose whether to retain the existing WSDL or to replace the existing WSDL by a uploading a new one. If you choose to use upload a new WSDL, you can specify whether the new WSDL should overwrite the existing one, or whether a new version of the WSDL should be created. Regardless of whether you overwrite the existing version or create a new version, you can specify a user-defined version number of the uploaded WSDL in the **User Version** field.

If you select the option **Interactive resolution of Imports/Includes**, and the attached WSDL contains an Import or Include reference to a WSDL that does not already exist in the registry, the **Attach WSDL to ...** dialog box allows you to upload the WSDL. You can also specify a user-defined version number of the uploaded WSDL in the **User Version** field.

Deleting Web Services

If you are not the owner of a Web Service asset, you cannot delete the Web Service unless you have Full permission on the Web Service (granted through either a role-based permission or instance-level permission).

The following general guidelines apply when deleting a (Web) Service asset in CentraSite Control:

- Deleting a Web Service asset permanently removes the Web Service from the catalog.
- A Web Service asset can only be deleted if it is not the target of an association from another registry object.
- When you delete a Web Service asset, CentraSite removes the catalog entry for the Web Service (that is, it removes the instance of the Web Service asset from CentraSite's object database). Also note that:

- The performance metrics and event information of the Web Service are also deleted.

Note:

When you delete the Web Service, this information is deleted by the built-in action **Delete RuntimeEvents and RuntimeMetrics of Service**, which is installed with CentraSite.

- When you delete a composite Web Service asset, all of its nonshared components are also deleted.
- Deleting a Web Service asset will *not* remove:
 - Other assets to which the Web Service refers (unless the reference is to an asset that is a nonshared component of the Web Service you are deleting). For example, if you are deleting a Web Service which has a Consumes or Consumed By relationship with other services in the registry, the related services will not be deleted.
 - Supporting documents that are attached to the Web Service.
 - Earlier versions of the Web Service. Only the latest version of an asset can be deleted; to remove earlier versions, they must be purged.
- You cannot delete a Web Service asset if:
 - The Web Service is in a pending state (for example, awaiting approval).
 - Any user in your CentraSite registry is currently modifying the Web Service.

> To delete Service assets

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Service**.

The Web Service assets (for which you have the View permission) are displayed in the **Assets** pane.

3. Right-click a Web Service you want to delete, and click **Delete**.

You can also select multiple Web Services, click the **Actions** menu, and click **Delete**.

4. Click **OK** in the confirmation dialog box.

Note:

If you have selected a set of Web Services, where one or multiple Web Services are in a pending state (for example, awaiting approval), CentraSite ignores the pending list of Web Services, and deletes any remaining Web Services for which you have the required permission.

REST Service Management

This section describes operations you can perform to manage REST services through CentraSite Control.

Adding REST Service Asset to Catalog

Beginning with version 9.7, CentraSite Control does not support the enhanced interface for a REST Service asset (in contrast, earlier versions of CentraSite Control supported the standardized interface for a REST Service asset). As a result, you cannot use CentraSite Control to add a REST Service or a Virtual REST Service asset to the catalog. Instead, use the CentraSite Business UI to create a REST Service asset in the CentraSite Registry Repository.

Documentation of the prior REST Service interface is available to Software AG customers who have a current maintenance contract in Empower Product Support Website.

Viewing REST Service List

You use the **Asset Catalog > Browse** to display the list of REST Service assets in CentraSite Control.

➤ **To view the list of REST Service assets**

1. In CentraSite Control, go to **Asset Catalog > Browse**.

A list of defined asset types is displayed in the **Types** pane.

2. In the displayed list of asset types, select **REST Service**.

The REST Service assets (for which you have the View permission) are displayed in the **Assets** pane.

The **Assets** pane provides the following information about each REST Service asset:

Column	Description
Name	Name of the REST Service asset.
Type	The asset's type definition, REST Service.
Version	The user-defined version identifier of the REST Service.

You can adjust the view to show or hide the individual column by using the **Browse By and Column Selection** icon that is located in the upper-right corner of the **Assets** pane.

The shortcut menu of a particular REST Service asset displays one or more actions that you can perform on that REST Service.

Action	Description
Copy	Copies an existing REST Service that is similar to the one you need, and allows you to examine and modify the copy. CentraSite treats the copy just as if it were a new REST Service that you created from scratch.
Details	Displays the details page of the REST Service.
Change Lifecycle State	Changes the lifecycle state of the REST Service.
Revert Pending State	Reverts a request that has been submitted for approval, and that is struck in Pending mode.
Change Owner	Changes the user ownership of the REST Service.
Run Policy	Executes a set of policies that are applicable for the asset's type definition, Service.
Change Organization	Changes the organizational ownership of the REST Service.
Export	Exports the REST Service from the registry to an archive file on the file system.
Impact Analysis	Helps to easily visualize the associations that exist between the REST Service asset and registry objects.
Attach WSDL	Appends a WSDL document to the REST Service.
Add New Version	Generates a new version of the REST Service.
Notify Me	Sends notifications to all the registered users when changes are made to the REST Service.

Action	Description
Add to List	Adds the REST Service to a list in My Favorites .
Add to Favorites	Adds a shortcut to the REST Service you want to use routinely or otherwise keep close at hand.
Virtualize	Creates a proxy for the REST Service for consumption.
Download Documents	Downloads files that are attached to the REST Service from Supporting Document Library (SDL).

Modifying REST Service Details

Beginning with version 9.7, CentraSite Control does not support the enhanced interface for a REST Service asset (in contrast, earlier versions of CentraSite Control supported the standardized interface for a REST Service asset). As a result, CentraSite Control only provides read-only access to the attributes of a REST Service asset.

You cannot use CentraSite Control to modify the attributes of a REST Service. Instead, use the CentraSite Business UI to modify the attributes of a REST Service in the CentraSite Registry Repository.

Documentation of the prior REST Service interface is available to Software AG customers who have a current maintenance contract in Empower Product Support Website.

Deleting REST Services

Beginning with version 9.7, CentraSite Control does not support the enhanced interface for a REST Service asset (in contrast, earlier versions of CentraSite Control supported the standardized interface for a REST Service asset). As a result, CentraSite Control only provides a read-only access to the REST Service asset.

You cannot use CentraSite Control to delete a REST Service. Instead, use the CentraSite Business UI to delete a REST Service from the CentraSite Registry Repository.

Documentation of the prior REST Service interface is available to Software AG customers who have a current maintenance contract in Empower Product Support Website.

OData Service Management

This section describes operations you can perform to manage OData services through CentraSite Control.

Adding an OData Service Asset to the Catalog

CentraSite Control does not provide a standardized interface for OData Service assets.

As a result, you cannot use CentraSite Control to add an OData Service asset to the catalog. Instead, use the CentraSite Business UI to create an OData Service in the CentraSite Registry Repository.

Viewing OData Service List

You use the **Asset Catalog > Browse** to display the list of OData Service assets in CentraSite Control.

> To view the list of OData Service assets

1. In CentraSite Control, go to **Asset Catalog > Browse**.

A list of defined asset types is displayed in the **Types** pane.

2. In the displayed list of asset types, select **OData Service**.

The OData Service assets (for which you have the View permission) are displayed in the **Assets** pane.

The **Assets** pane provides the following information about each OData Service asset:

Column	Description
Name	Name of the OData Service asset.
Type	The asset's type definition, OData Service.
Version	The user-defined version identifier of the OData Service.

You can adjust the view to show or hide the individual column by using the **Browse By and Column Selection** icon that is located in the upper-right corner of the **Assets** pane.

The shortcut menu of a particular OData Service asset displays one or more actions that you can perform on that OData Service.

Action	Description
Copy	Copies an existing OData Service that is similar to the one you need, and allows you to examine and modify the copy. CentraSite treats the copy just as if it were a new OData Service that you created from scratch.
Details	Displays the details page of the OData Service.
Change Lifecycle State	Changes the lifecycle state of the OData Service.
Revert Pending State	Reverts a request that has been submitted for approval, and that is struck in Pending mode.
Change Owner	Changes the user ownership of the OData Service.
Run Policy	Executes a set of policies that are applicable for the asset's type definition, Service.

Action	Description
Change Organization	Changes the organizational ownership of the OData Service.
Export	Exports the OData Service from the registry to an archive file on the file system.
Impact Analysis	Helps to easily visualize the associations that exist between the OData Service asset and registry objects.
Attach WSDL	Appends a WSDL document to the OData Service.
Add New Version	Generates a new version of the OData Service.
Notify Me	Sends notifications to all the registered users when changes are made to the OData Service.
Add to List	Adds the OData Service to a list in My Favorites .
Add to Favorites	Adds a shortcut to the OData Service you want to use routinely or otherwise keep close at hand.
Virtualize	Creates a proxy for the OData Service for consumption.
Download Documents	Downloads files that are attached to the OData Service from Supporting Document Library (SDL).

Modifying OData Service Details

CentraSite Control does not provide a standardized interface for OData Service assets. As a result, CentraSite Control only provides read-only access to the attributes of an OData Service asset.

You cannot use CentraSite Control to modify the attributes of an OData Service. Instead, use the CentraSite Business UI to modify the attributes of an OData Service in the CentraSite Registry Repository.

Deleting OData Services

CentraSite Control does not provide a standardized interface for OData Service assets. As a result, CentraSite Control only provides a read-only access to the OData Service asset.

You cannot use CentraSite Control to delete an OData Service. Instead, use the CentraSite Business UI to delete an OData Service from the CentraSite Registry Repository.

Application Management

This section describes operations you can perform to manage Application assets through CentraSite Control.

Adding an Application to Your Asset Catalog

Pre-requisites:

To add assets to an organization's asset catalog, you must belong to a role that has the `Create Assets` or `Manage Assets` permission for that organization.

Note:

By default, users with the `CentraSite Administrator` or `Asset Administrator` role have this permission.

You create an **Application** asset to specify the consumer applications that are authorized to consume a particular Service, BPEL Process, or XML Schema.

The Application asset type is one of the predefined asset types installed with CentraSite. Application assets are used by the policy-enforcement point (PEP) to determine from which consumer application a request for an asset originated. An Application asset defines the precise characteristics by which the PEP can identify or authenticate messages from a specific consumer application at run-time.

In CentraSite Control, you add an Application asset to the catalog *from scratch*, meaning that you create the Application asset (and set its attributes) manually.

When defining Application assets, keep the following points in mind:

- Any user who has permission to publish an asset to CentraSite can define an Application asset. However, not all users are generally qualified to create an asset of this type. Defining applications is a critical task that should be performed only by an administrator who is familiar with the Mediator(s), virtual services, and run-time policies in your environment.
- An Application asset becomes available to Mediator only when you synchronize the consumer application in CentraSite with the Mediator.
- Treat Application assets as global objects and make them available to all organizations. Be sure that your registry contains only one Application asset per consumer application (that is, a consumer application should be represented by *one and only one Application asset* in the registry).
- Be sure that the identifiers that you assign to an Application asset are unique to that Application asset. If multiple Application assets have the same identifier, Mediator associates the identifier with the first matching application it finds in its local list of Application assets at run time.
- If you control access to virtual services based on consumer applications (that is, you use run-time policies that include the `Authorize User` action), consider:
 - Including an approval step in your consumer-registration policy that requires a security administrator to review and approve the registration event.
 - Giving only a small group of knowledgeable administrators permission to modify an Application asset after it is registered to a virtual service. This prevents users from adding unauthorized identifiers to an existing Application asset and thus, allowing unauthorized consumer applications to access the virtual service.

➤ To add an Application asset to the catalog

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. Click **Add Asset**.
3. In the **Add Asset** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Type	Select the asset type, Application .
Name	<p><i>Optional.</i> Name of the Application asset.</p> <p>Note: This is the name that users will see when they view this Application in the CentraSite User Interfaces. Therefore, we recommend that you specify a meaningful name for each Application.</p> <p>An Application asset name can contain any characters (including spaces), and must be unique within an organization.</p>
Description	<p><i>Optional.</i> The description for the Application.</p> <p>Note: This is the description information that users will see when they view this Application in the CentraSite User Interfaces. Therefore, we recommend that you specify a meaningful description for each Application.</p>
Organization	<p>The organization to which you want to add the Application. (The Organization list only displays organizations for which you have the Manage Assets permission or at least the Create Assets permission.) If you select an organization other than your own organization, you will nevertheless be the owner of the asset.</p> <p>Important: Select the organization carefully. You cannot change the organization assignment later. You can, however, export a Application asset from one organization and import it to another.</p>
Initial Version	<p><i>Optional.</i> An identifier for the initial version of the Application. This is the user-defined version, as opposed to the automatically assigned system version. You can enter any string in this field, that is, the version identifier does not need to be numeric. You can also leave the field blank. You can later create new versions of the asset. The default is 1.0.</p>

Field	Description
	If the versioning feature is disabled for the Service asset type, the field is nevertheless displayed, thus allowing you to assign an identifier for this first version.0

4. Click **OK**.

The newly created Application asset is added to the CentraSite Registry Repository.

5. Configure the extended attributes of the Application asset.

Viewing Application Asset List

You use the **Asset Catalog > Browse** to display the list of Application assets in CentraSite.

> To view the list of Applications

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Application**.

The Application assets (for which you have the View permission) are displayed in the **Assets** pane.

The **Assets** pane provides the following information about each Application asset:

Column	Description
Name	Name of the Application asset.
Type	The asset's type definition, Application.
Version	The user-defined version identifier of the Application.

You can adjust the view to show or hide the individual column by using the **Browse By and Column Selection** icon that is located in the upper-right corner of the **Assets** pane.

The shortcut menu of a particular Application asset displays one or more actions that you can perform on that Application.

Action	Description
Copy	Copies an existing Application that is similar to the one you need, and allows you to examine and modify the copy. CentraSite treats the copy just as if it were a new Application that you created from scratch.
Details	Displays the details page of the Application.

Action	Description
Change Lifecycle State	Changes the lifecycle state of the Application.
Revert Pending State	Reverts a request that has been submitted for approval, and that is struck in Pending mode.
Change Owner	Changes the user ownership of the Application.
Run Policy	Executes a set of policies that are applicable for the asset's type definition, Service.
Change Organization	Changes the organizational ownership of the Application.
Export	Exports the Application from the registry to an archive file on the file system.
Impact Analysis	Helps to easily visualize the associations that exist between the Application asset and registry objects.
Attach WSDL	Appends a WSDL document to the Application.
Add New Version	Generates a new version of the Application.
Notify Me	Sends notifications to all the registered users when changes are made to the Application.
Add to List	Adds the Application to a list in My Favorites .
Add to Favorites	Adds a shortcut to the Application you want to use routinely or otherwise keep close at hand.
Virtualize	Creates a proxy for the Application for consumption.
Download Documents	Downloads files that are attached to the Application from Supporting Document Library (SDL).

Modifying Application Asset Details

Each tab on an Application asset's details page represents a collection of attributes called a *Profile*. The Application asset type has a unique set of profiles. However, an administrator can configure the Application asset type to display a customized set of profiles and attributes. You use the asset details page to examine or modify the attributes displayed in the individual profiles.

The following general guidelines apply when modifying the details of an Application asset in CentraSite Control:

- If you are not the owner of the Application asset, you cannot examine or modify the details of the asset, unless you have the View or Modify permission on the asset (granted through either a role-based permission or an instance-level permission).

- When you view the details page of an Application asset, you will only see profiles for which you have the profile-level View permission. You will only be able to modify the attributes of the profiles on which you have Modify permission.
- Some attributes accept only specific types of information. For example, if the Application asset type includes a URL type attribute, you must supply a URL when you modify that attribute. Other attribute types that require a specific type of value include the Date and Email attributes.
- Some attributes are designed to be read-only. You cannot modify the read-only attributes even if you have the CentraSite Administrator role.

➤ **To modify Application asset's details**

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Application**.
3. In the **Assets** pane, right-click the Application for which you want to modify the attributes, and click **Details**.

The Application Details page is displayed.

4. To modify the generic attributes of the Application that are displayed in the area labeled **Basic Information**, change values of the attributes in the respective data fields as required.
5. To modify the extended attributes of the Application that are displayed in the individual profiles, perform the following:
 - a. Select the profile for which you want to modify the attributes.
 - b. Change values of the attributes in the respective data fields as necessary.
 - c. Repeat steps 5.a and 5.b for each profile for which you want to modify the attributes.
6. Click **Save** to save the updated changes.

Deleting Application Assets

If you are not the owner of an Application asset, you cannot delete the Application unless you have Full permission on the Application (granted though either a role-based permission or instance-level permission).

The following general guidelines apply when deleting an Application asset in CentraSite Control:

- Deleting an Application asset permanently removes the Application from the catalog.
- An Application asset can only be deleted if it is not the target of an association from another registry object.

- When you delete an Application asset, CentraSite removes the catalog entry for the Application (that is, it removes the instance of the Application asset from CentraSite's object database). Also note that:

- The performance metrics and event information of the Application are also deleted.

Note:

When you delete the Application asset, this information is deleted by the built-in action **Delete RuntimeEvents and RuntimeMetrics of Service**, which is installed with CentraSite.

- Deleting an Application asset will *not* remove:
 - Other assets to which the Application asset refers (unless the reference is to an asset that is a nonshared component of the asset you are deleting). For example, if you are deleting is an Application asset with a Consumes or Consumed By relationship with other services in the registry, the related services will not be deleted.
 - Supporting documents that are attached to the Application asset.
 - Earlier versions of the Application asset. Only the latest version of an asset can be deleted; to remove earlier versions, they must be purged.
- You cannot delete an Application asset if:
 - The Application is in a pending state (for example, awaiting approval).
 - Any user in your CentraSite registry is currently modifying the Application.

➤ **To delete Application assets**

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Application**.

The Application assets (for which you have the View permission) are displayed in the **Assets** pane.

3. Right-click a Application you want to delete, and then click **Delete**.

You can also select multiple Applications, click the **Actions** menu, and click **Delete**.

4. Click **OK** in the confirmation dialog box.

Note:

If you have selected a set of Applications, where one or multiple Applications are in a pending state (for example, awaiting approval), CentraSite ignores the pending list of Applications, and deletes any remaining Applications for which you have the required permission.

General Procedures across Assets

This section outlines the general procedures across assets performed through CentraSite Control.

Attaching Supporting Documents to Asset

To attach documents to an asset, you must have the Manage Assets permission for the organization to which the asset belongs.

Some assets include attributes that allow you to associate supporting documents such as programming guides, sample code, script files, and project plan with the asset.

For example, the Service asset includes the **Specification** profile. This profile contains several file-related attributes representing external documents such as Functional Requirements, Error Messages, Release Notes and so forth.

When attaching documents to an asset, keep the following points in mind:

- A document in the supporting document library can be shared by multiple assets in the catalog. For example, if you have two Service assets that refer to the same programming guide, you can upload one copy of the programming guide to the supporting document library and attach it to both assets.
- If the document that you want to attach to an asset is not already in the supporting document library, you can upload it to the library when you set the file-related attribute. (To use this feature, you must belong to a role that has the Create Assets permission for the organization into whose library you want place the document.)
- If you attach a document to an asset that will be viewed by users in other organizations, those users will only be able to view the attached document if they have the View Supporting Documents permission.
- CentraSite relies on file extensions to determine a file's type. When you upload a file from your local machine to the supporting document library, be sure the name of the file on your local machine includes a file extension so that CentraSite can determine the file's type and mark it correctly in the supporting document library.
- When you attach a document to an attribute using a URL, be aware that the attribute simply creates a reference to the document at that URL. CentraSite does not retrieve the document from the URL and place it in its document repository. If the referenced document is subsequently modified, renamed or deleted, the reference may become invalid.

Attaching Documents Using URL

You can attach a document from your organization's *supporting document library*. The supporting document library is a collection of shareable documents that members of your organization have uploaded to CentraSite's document repository.

> To attach a document using URL

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the **Assets** pane, right-click an asset to which you want to attach a document, and then click **Details**.

This opens the details page of the asset.

3. Click the profile that contains the attribute to which you want to attach the document.
4. Locate the attribute and click the corresponding **Attach** button. (If the attribute has existing attachments, be sure to click the *lower-most* **Attach** button. If you click an **Attach** button that belongs to an existing attachment, you will *replace* that attachment. If you do not see an available **Attach** button, use the plus button to display one.)

The **Add External Link** dialog box is displayed.

5. Select the **Point to URL** option button, and type the document's URL in the text box.

Supported protocols are http, https, file, and ftp.

6. Click **OK**.
7. Repeat steps 3 to 5 for each URL that you want to attach to the attribute.
8. Click **Save** to save your changes.

Attaching Documents from Supporting Document Library

You can attach any document on the network that is accessible through a URL (permitted protocols are http, https, file, and ftp).

➤ To attach a document from supporting document library

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the **Assets** pane, right-click an asset to which you want to attach a document, and then click **Details**.

This opens the details page of the asset.

3. Click the profile that contains the attribute to which you want to attach the document.
4. Locate the attribute and click the corresponding **Attach** button. (If the attribute has existing attachments, be sure to click the *lower-most* **Attach** button. If you click an **Attach** button that belongs to an existing attachment, you will *replace* that attachment. If you do not see an available **Attach** button, use the plus button to display one.)

The **Add External Link** dialog box is displayed.

5. Select the **Add a document from the Supporting Document Library** option button.
6. If the document that you want to attach to the asset is not already in the supporting document library, use the following steps to upload it to the library:
 - a. Click **Upload New Document to Selected Folder**.
 - b. In the **Add Document** dialog box, type the appropriate information for each of the displayed data fields.

In this field...	Specify...
Folder	The name of the target folder in the supporting document library. Click Browse to select the required folder.
File	The full pathname within your operating system environment of the file that you want to upload to the supporting document library. Click Browse to select the required file. To ensure that CentraSite sets the file type correctly in the supporting document library, the name of the file should include an extension that indicates the type of data it contains.
Name	The name by which the document will be identified in the library. This is also the name that users will when the document is attached to a File attribute.
Description	<i>Optional.</i> A descriptive comment that provides users with more information about the document. Description cannot exceed 4000 characters.

7. In the **Supporting Documents** list, select the document that you want to attach to the asset.
8. Click **OK**.
9. Repeat steps 3 to 7 for each document that you want to attach to the attribute.
10. Click **Save** to save your changes.

Removing Supporting Documents from Asset

You can remove a document that is attached to an asset.

> To remove supporting document from asset

1. In CentraSite Control, go to **Asset Catalog > Browse**.

2. In the **Assets** pane, right-click an asset whose supporting document you want to remove, and then click **Details**.

This opens the details page of the asset.

3. Click the asset's profile that contains the required File attribute.
4. Locate the supporting document you want to remove, and then click the minus button. Repeat for each document that you want to remove.
5. Click **Save**.

Changing Lifecycle State of Asset

If an asset has an associated lifecycle model, you might need to switch the lifecycle state of an asset.

When changing the lifecycle state of an asset, keep the following points in mind:

- For any given lifecycle model, a list of names of users and/or groups who are allowed to move assets to new states is maintained within the definition of the lifecycle model. For each user or group, the permission to move assets to new states can be restricted to a subset of the available states in the model. When the lifecycle model is assigned to an asset and a state has users or groups defined for it, only a user who is one of the defined users or groups can make the transition of the asset into that state. If no users or groups are defined for a particular state, any user who has Modify permission on the asset can change the lifecycle state for that asset.
- Users with the Manage Lifecycle Models permission can define the list of users and groups who are allowed to enter new states in a lifecycle model.

Note that this function can be performed on a single asset or a set of assets.

➤ To change the lifecycle state of assets

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the **Assets** pane, right-click an asset whose lifecycle state you want to change, and click **Change Lifecycle State**.

You can also select multiple assets, click the **Actions** menu, and click **Change Lifecycle State**. If there are pending changes for the asset, you will receive an intermediate prompt asking you if you wish to save or reject the changes before the lifecycle state is changed.

Important:

If you have selected several assets where one or more of them are predefined assets such as UDDI Services. You can use the **Change Lifecycle State** command to switch the lifecycle state of the assets. However, as you are not allowed to change lifecycle state of predefined assets, only assets you have permission for will be changed. The same applies to any other assets for which you do not have the required permission.

3. In the **Change Lifecycle State** dialog box, select the state to which you want to switch the asset.

The list contains only the states that you are permitted to assign to the asset.

4. Click **OK**.

If the state change requires approval, CentraSite Control will initiate an approval workflow and your request for a state change will be submitted to the appropriate approvers. While the request is awaiting approval, the asset appears in the pending mode.

Setting Permissions on Asset

To set permissions on an asset, you must have the Manage Assets permission or have the Full instance-level permission on the asset itself.

By default, everyone in your organization is permitted to view the assets that you publish. However, only you (as the owner of the asset) and users who belong to a role with the Manage Assets permission for your organization are allowed to view, edit, and delete these assets. To enable other users to view, edit, and delete an asset that you have published, you must modify the asset's permission settings.

Important: CentraSite Control does not allow you to set permissions on assets of the following types:

- XML Service
- REST Service
- OData Service
- Virtual XML Service
- Virtual REST Service
- Virtual OData Service

When setting permissions on assets, keep the following points in mind:

- To set permissions on an asset, you must belong to a role that has the Manage Assets permission or have the Full instance-level permission on the asset itself.
- You can assign permissions to any individual user or group defined in CentraSite.
- The groups to which you can assign permissions include the following system-defined groups:

Group Name	Description
Users	All users within a specified organization.
Members	All users within a specified organization and its child organizations.
Everyone	All users of CentraSite <i>including guest users</i> .

- If a user is affected by multiple permission assignments, the user receives the union of all the assignments. For example, if group ABC has Modify permission on an asset and group XYZ

has Full permission on the same asset, users that belong to both groups will, in effect, receive Full permission on the asset.

The same principle applies to users who have both role-based permissions and instance-level permissions on the same asset. In this case, users receive the union of the role-based permission and the instance-level permission on the asset.

- If you intend to give users in other organizations access to the asset and the asset includes supporting documents that you want those users to be able to view, make sure you give those users permission to view the supporting documents as well as the asset itself.

You can set the permissions on an asset in two ways:

- **Using the Permissions profile in the user interface**

You can use the **Permissions** profile in CentraSite Control.

- **Using the Set Instance and Profile Permissions policy action**

You can use the **Set Instance and Profile Permissions** policy action in a design/change-time policy to automatically assign permissions to an asset during any of the following events:

- PostCreate
- PreStateChange
- PostStateChange
- OnTrigger

For more information about creating policies and about using the Set Instance and Profile Permissions action, see the *CentraSite Developer's Guide*.

Restricting Access to Asset Profiles

CentraSite allows you to set permissions on individual profiles within an asset. This feature enables you to specify which of the available profiles can be viewed or edited by users when they display the asset in CentraSite Control. For any given asset, you can define different profile permissions for different users. For example, if an asset includes a profile called Source Control that displays links to your source control systems, you might want to restrict the visibility of that profile to authorized developers.

You define the user-specific or group-specific profile permissions of an asset through the asset's the **Permissions** profile.

The profile permissions that can be set on a given asset for any user or group are:

Permission	Description
View	Enables the specified user or group to the profile when they view the asset.
Modify	Enables the specified user or group to modify the attribute settings in the profile when they view the asset.

The individual profiles do not include the Full permission because users cannot delete a profile from an individual asset.

Important:

Be aware that profile permissions can be used to prevent users from viewing and/or accessing a particular set of attributes through CentraSite's graphical user interfaces. However, they do not restrict access to the attributes themselves at the API level.

By default, if a user with Guest role has permission to view the details of an asset, CentraSite Control includes the asset's **Summary** profile in the set of profiles displayed to this user. If you wish to suppress the display of this profile for users with Guest role, you can do this as follows:

> To suppress display of Summary tab for Guest users

1. Open the configuration file `plugin.xml` in a rich text editor.

You can find the file in the `<RuntimeDir>\workspace\webapps\PluggableUI\CentraSiteControl` folder.

2. Locate the `<extension>` property:

```
<extension point="com.softwareag.cis.plugin.parameter"
id="guestAllowSummaryProfileVisible" value="true" />
```

3. Set the value of the extension `guestAllowSummaryProfileVisible` to `false`. The default is `true`.
4. After you save the file, restart Software AG Runtime.

To reactivate the **Summary** profile for Guest users, replace the extension's value `false` with `true`, and then restart Software AG Runtime.

Setting Instance Level Permissions on Asset**> To assign instance-level permissions on an asset**

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the **Assets** pane, right-click an asset whose permissions you want to assign, and click **Details**.

You can also select multiple assets, click the **Actions** menu, and click **Details**.

3. Click the **Permissions** tab.
4. To add users or groups to the **Users and Groups** list, do the following:
 - a. Click **Add Users and Groups**.

- b. Select the users and groups to which you want to assign permissions.

To filter the list, type a partial string in the **Search** field. CentraSite applies the filter to the **Name** column.

Examples

String	Description
b	Displays names that contain b
bar	Displays names that contain bar
%	Displays all users and groups

- c. Click **OK**.

5. To assign specific permissions to each user or group do the following:

Permission	Allows the selected user or group to...
View	View the asset.
Modify	View and edit the asset.
Full	View, edit, and delete the asset. This permission also allows the selected user or group to assign instance-level permissions to the asset.

When you assign instance-level permissions on an asset, the related objects (for example, bindings, operations, interfaces, and so on.) receive the same permissions that are assigned on the asset.

6. To ensure that the asset's dependent objects (for example, a WSDL or schema) receive the same permissions, select the **Propagate permissions to dependent objects** check box. If you do not select this check box, the permissions of the dependent objects will not be modified.
7. To ensure that the dependent objects of the same object type receive the same profile permissions, select **Propagate profile permissions**.
8. Click **Save**.

Setting Instance Level Profile Permissions on Asset

➤ To assign instance-level permissions on an asset's profiles

1. In CentraSite Control, go to **Asset Catalog > Browse**.

2. In the **Assets** pane, right-click an asset whose permissions you want to assign, and click **Details**.

You can also select multiple assets, click the **Actions** menu, and click **Details**

3. Click the **Permissions** tab.
4. Locate the user or group for which you wish to set profile permissions.
5. Click the arrow next to the user or group name to open the profile permission list.
6. To indicate which profiles the user or group is permitted to view or modify do the following:

Permission	Allows the selected user or group to...
View	View the asset.
Modify	View and edit the asset.
Full	View, edit, and delete the asset. This permission also allows the selected user or group to assign instance-level permissions to the asset.

When you assign instance-level permissions on an asset, the related objects (for example, bindings, operations, interfaces, and so on.) receive the same permissions that are assigned on the asset.

7. Click **Save** to save the new permission settings.

Propagation of Permissions

An asset can have one or more dependent objects. For example, a Service asset can refer to a WSDL which in turn can refer to one or more XML Schema assets. You can optionally choose whether the permissions assigned to an asset instance should be automatically propagated to the asset instance's dependent objects.

Propagation of Instance Level Permissions

The access level permissions that are assigned on an asset are implicitly propagated to these dependent objects. This behavior is activated when you select the check box **Propagate permissions to dependent objects** in the asset's **Permissions** tab. For example, assigning Modify permission on a Service asset propagates the Modify permission to the asset's WSDL, schemas, and so on.

If you do not have permission to assign instance-level permissions to a dependent object, the dependent object will not be modified and a warning message will be issued.

You can propagate permissions only for the following asset types:

- Service

- XML Schema
- BPEL

Propagation of Profile Permissions

In addition to propagating permissions that control the access to an asset instance, it is also possible to propagate permissions that control the access to the asset instance's profiles. This means that the profile permissions that you define for an asset instance can be propagated to the asset's dependent objects. However, this is only possible if the dependent object is of the same asset type as the first object; this restriction arises because different asset types can have different sets of profiles.

This behavior is activated when you mark the check box **Propagate profile permissions** in the asset's **Permissions** tab. This check box is only available for the following asset types:

- Service
- XML Schema

Versioning an Asset

To version an asset, you must have the Manage Assets permission for the organization to which the asset belongs.

You can use the versioning feature in CentraSite to add an updated version of an asset to the catalog. For example, if you make significant changes to a Service asset (such as adding operations to the service or modifying the data types that it uses), you can use the versioning feature to add the new version of the service to the catalog.

Versioning can be active or inactive for any given asset type. The method for activating versioning for an asset type is included in the CentraSite. Note also the restrictions for activating versioning, described in [“Considerations for Asset Types of the Suite” on page 601](#).

When you generate a new version of an asset, CentraSite adds a new asset of the same type to the catalog. The new asset will have the same name and description as the one from which it was versioned. It will have an updated version number. The new version is related to the old version by a Supersedes association from the new version to the old version. In cases where the detail page of an asset has a **Summary** profile, the association is displayed under the **Summary** profile.

Note:

Depending on the type of asset you version, some of the attributes are cloned from the original asset and others are not. For example, when you version a Web service, the settings on the **Classifications** profile are cloned, however, the attribute settings on many of the other profiles, including the **Permissions** profile, are not. After you version an asset, you should always examine the attribute settings for the new version and set them appropriately.

The metrics and event information that was collected for the old version of the asset will remain unchanged in the registry/repository. The old version's metrics and event information will not be copied to the new version. CentraSite will begin collecting metrics and event information for the new version of the asset.

CentraSite maintains two sets of version numbers for an asset. One set is maintained for CentraSite's own internal use. CentraSite automatically assigns this version number when you create a new version of an asset. You cannot modify it. The version numbers assigned by CentraSite have the format *<MajorVersion>.<Revision>* and are always sequentially numbered starting from 1.0 (for example, 1.0, 2.0, 3.0). If the revision feature is enabled, the revision number is incremented automatically each time you modify the current version of the asset. For detailed information, see [“Managing Asset Revisions” on page 601](#).

Each version of an asset also has a separate user-defined version identifier. This is the public version number that CentraSite shows to users when it displays the catalog. The user-defined version identifier does not need to be numeric. For example, you might use a value such as V2.a (beta) to identify a version.

Important:

CentraSite does not support versioning of assets of the following types in CentraSite Control:

- XML Service
- REST Service
- OData Service
- Virtual Service
- Virtual XML Service
- Virtual REST Service
- Virtual OData Service

Instead, you can use the CentraSite Business UI for versioning assets of any of the custom and predefined assets types in CentraSite.

Creating New Version of Asset

When you version an asset, you become the owner of the new version of the asset. Ownership is not carried forward from the previous version.

The new version of the asset will belong to the same organization as its previous version.

> To create new version of an asset

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the **Assets** pane, right-click an asset for which you want to create a new version, and click **Add Version**.

You can also select multiple assets, click the **Actions** menu, and click **Add Version**

Note:

You can also initiate this step using the **Versions** profile in the details page of an asset.

3. In the **Add Version** dialog box, type the appropriate information for each of the displayed fields:

In this field...	Do the following...
Namespace	<p>The namespace associated with this new version. This is of specific relevance for Service assets. The namespace given here reflects the target namespace defined in the associated WSDL file. A change of the namespace can be a differentiating factor between versions. Note that if you supply a new namespace here, you should ensure that the WSDL associated with this asset also reflects the new namespace.</p>
System Version	<p>In addition to the user-defined version, CentraSite automatically creates a System Version number. The system version number is independent from the version number you specify. The system version numbers are maintained for CentraSite's own internal use. CentraSite automatically assigns this version number when you create a new version of an asset. You cannot modify it. The version numbers assigned by CentraSite have the format <MajorVersion>. <Revision> and are always sequentially numbered starting from 1.0 (for example, 1.0, 2.0, 3.0). The revision number is incremented automatically each time you modify the current version of the asset.</p>
User Version - New	<p>An identifier for the new version. You can use any versioning scheme you choose. The version identifier does not need to be numeric. Examples:</p> <pre data-bbox="639 1052 1463 1178"> 0.0a 1.0.0 (beta) Pre-release 001 V1-2007.04.30 </pre>
Organization	<p>Choose the organization to which the versioned asset will be added. (The drop-down list will contain the list of organizations to which you are permitted to add assets.) If you select an organization other than your own organization, you will nevertheless be the owner of the asset.</p> <p>Important: Choose the organization carefully. You cannot change the organization assignment later. You can, however, export a versioned asset from one organization and import it to another.</p>
Change Log	<p><i>Optional.</i> Type a comment or other descriptive information about the new version.</p>
Propagate versions to dependent objects	<p><i>(CentraSite only processes this check box for assets of type Service.)</i></p> <p>Select this check box to automatically create new versions of all of the service's dependent schemas. The schemas will only be updated if you have permissions to modify them.</p>

4. Click **OK**.
 - If one or more of the selected assets is not the most recent version of the asset, CentraSite displays an error message, and no new version is created for any of the assets.
 - If one or more of the selected assets is a Virtual Service asset, CentraSite displays an error message, and no new version is created for any of the assets.

Locating Other Versions of Asset

The **Versions** profile for an asset displays the list of all the asset's versions. To locate other versions of an asset, display the details page of an asset and examine its **Versions** profile.

> To locate other versions of an asset

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the **Assets** pane, right-click an asset to view a list of its versions, and click **Details**.

You can also select multiple assets, click the **Actions** menu, and click **Details**

3. Click the **Versions** tab.

A list of all versions of the asset is displayed in the **Versions** profile.

4. To display the details for one of the listed versions, click the name of that version.

Purging Older Versions

If you have created several versions of an asset, you might want to delete some or all the older versions. You can do this by purging old versions.

> To purge old versions of an asset

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the **Assets** pane, right-click an asset to view a list of its versions, and then click **Details**.

You can also select multiple assets, click the **Actions** menu, and click **Details**.

3. Click the **Versions** tab.

A list of the available versions for the asset is displayed in the **Versions** profile.

4. Right-click a version you want to purge, and then click **Purge**.

All versions older than the selected version are deleted.

If, for example, you have versions 1, 2, 3, 4, and 5 of an asset, and you want to retain the versions 4 and 5, you can select the version 4, and click **Purge**. This deletes all versions older than version 4.

Considerations for Asset Types of the Suite

If you are using CentraSite in conjunction with other components of the webMethods Suite, the versioning capability for the asset types defined by these components is by default not activated. Unless the documentation for the webMethods product suite components states otherwise, do not activate the versioning for these asset types.

Managing Asset Revisions

Within each version of an asset you can have several revisions. When revision processing is enabled, CentraSite stores a new revision of the current asset version each time you update the asset. In CentraSite Control, you can view the stored asset properties for each stored revision.

For example, if you have an asset whose current version is 2.1, you might want to modify the contents of the `Description` property of the asset in the asset's detail page, but without creating a new version. In this case, when you save the new description of the asset, the version number is updated automatically by CentraSite to 2.2.

As described in [“Versioning an Asset” on page 597](#), CentraSite automatically maintains an internal version number for each asset. The version number has the format `<MajorVersion>.<Revision>`, for example, `2.0`.

When revision processing is enabled, the revision number of an asset is initially 1 and is automatically incremented each time you save changes to the asset. When revision processing is disabled, all revisions of an asset except the most recent are discarded and the revision number is automatically reset to 0.

When you create a new version of an asset, CentraSite internally treats the new asset version as a new registry object and assigns a new internal object ID to it. When you create a new revision of an asset, CentraSite internally treats the new asset revision as the same registry object and does not assign a new internal object ID to it.

Currently, when you switch the revision feature on or off, you can only do this for all assets in all organizations; there is no possibility of limiting the effects of revision processing to a subset of the assets or organizations.

By default, that is, immediately after the installation of CentraSite, revision processing is switched off.

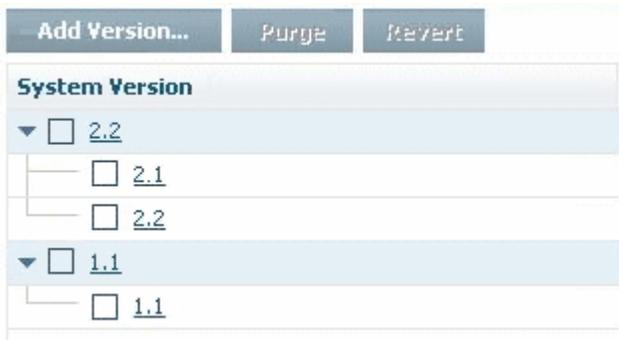
Deleting an object also deletes all of its revisions. The constraints for deleting objects however apply only to the current revision. This means that all incoming associations that exist on the current state of the object have to be released before deletion.

If an object with existing revisions is exported, then only the currently selected revision is exported and the revision history is not exported.

Searching (including Advanced Search) always defaults to the current revision of an object. It is possible to express a revision label in an advanced search. If an advanced search expects revisions to be found, it is not possible to define additional search criteria.

When revision processing is switched on, you can view the revisions of any given asset by choosing the **Versions** tab in the asset's detail view. If an asset has several revisions, these will be shown. If an asset has not been modified since it was created, the asset's version will be shown with a single revision with the number .1

The following example shows an asset with two versions. Version 2 of the asset has two revisions, namely 2.1 and 2.2. Version 1 is unchanged since it was created and has therefore a revision 1.1:



If you want to view the asset properties stored for any particular revision, choose the link for the required revision. Note that you cannot change the properties of a revision of an asset version if there is a newer revision of the same asset version.

Purging Old Revisions

If you have several revisions of a version of an asset, you can delete one or more of the older revisions by using the **Purge** button as follows:

> To purge revisions of an asset

1. Ensure that the **Versions** tab of the asset is selected in the asset's detail view.
2. In the asset version where you want to purge the revisions, select the oldest revision that you want to retain.
3. Click **Purge**.

This deletes all revisions older than the selected revision.

Reverting to an Older Revision

If you have several revisions of an asset version and you want to revert to an older revision than the current revision, you can do this using the **Revert** button. When you revert to an older revision, all revisions newer than the selected revision are deleted.

Follow these general guidelines when reverting asset to an older revision:

- The revert feature is only available if the outgoing associations of the older revision are still valid, that is, the objects referred to by the older revision must still exist.
- If an older revision is reactivated in this way, the ownership and organization of the asset are set to those of the user who activated the revert feature.
- If you want to revert to a revision that has a different lifecycle state than the current revision, you will be asked to confirm that you want to revert to the older revision. In this case, you must also have permission to change the lifecycle state.
- When you revert to an older revision, instance level permissions are not restored.
- To be able to use the revert feature, you need View and Modify permissions on the current state of the object as well as at least View permissions on the revision that you want to revert to.

➤ To revert to an older revision of an asset version

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the **Assets** pane, right-click an asset to view a list of its revisions, and click **Details**.

You can also select multiple assets, click the **Actions** menu, and click **Details**.
3. In the asset details page, click the **Versions** tab.

A list of the available revisions for the asset is displayed in the **Versions** profile.
4. In the asset version where you want to revert to an older revision, right-click the older revision that you want to revert.
5. Click **Revert**.

This removes all revisions newer than the selected revision, and makes the selected revision the new current revision.

Switching Revision Processing On

To switch the revision processing on, ensure that:

- All of the JAR files of the `redist` folder are included in the CLASSPATH variable.
- The location of the `redist` folder is included in the PATH variable. This is because the `redist` folder contains libraries required by the Java classes.

The `redist` folder is located directly under the CentraSite installation directory.

Currently, revision processing is switched on using a Java command at the command line.

➤ To switch the revision processing on

- Run the command `java com.centrasite.registry.revision.admin.RevisionAdministrator -enable -user user -password password`.

The user you specify must belong to the CentraSite Administrator role.

The Java program now runs to completion and the revision processing is switched on.

If you want to run the Java program to access a CentraSite registry running on a remote host, you can add the option `-h <host>` to the Java command, where `<host>` is the URL of the remote CentraSite host.

Switching Revision Processing Off

To switch the revision processing off, ensure that:

- All of the JAR files of the `redist` folder are included in the CLASSPATH variable.
- The location of the `redist` folder is included in the PATH variable. This is because the `redist` folder contains libraries required by the Java classes.

The `redist` folder is located directly under the CentraSite installation directory.

Currently, revision processing is switched off using a Java command at the command line.

➤ To switch the revision processing off

- Run the command `java com.centrasite.registry.revision.admin.RevisionAdministrator -disable -user user -password password`.

The user you specify must belong to the CentraSite Administrator role.

The Java program now runs to completion and the revision processing is switched off.

If you want to run the Java program to access a CentraSite registry running on a remote host, you can add the option `-h <host>` to the Java command, where `<host>` is the URL of the remote CentraSite host.

Checking Current State of Revision Processing

To check the current state of revision processing, ensure that:

- All of the JAR files of the `redist` folder are included in the CLASSPATH variable.
- The location of the `redist` folder is included in the PATH variable. This is because the `redist` folder contains libraries required by the Java classes.

The `redist` folder is located directly under the CentraSite installation directory.

You can check whether revision checking is currently switched on or off using a Java command at the command line.

➤ To check the current state of revision processing

- Run the command `java com.centrasite.registry.revision.admin.RevisionAdministrator -check -user user -password password`.

The user you specify must belong to the CentraSite Administrator role.

The Java program now runs to completion and indicates the current status of revision processing: the value `true` indicates that revision processing is switched on; the value `false` indicates that revision processing is switched off.

If you want to run the Java program to access a CentraSite registry running on a remote host, you can add the option `-h <host>` to the Java command, where `<host>` is the URL of the remote CentraSite host.

Changing Ownership of Asset

To change the ownership of an asset, you must belong to the CentraSite Administrator role.

In CentraSite, there are two concepts of ownership. An asset belongs to a particular *user* (known as the asset's *owner*) and it also belongs to a particular *organization*. The owner of an asset has special access rights to the asset and serves as the asset's main point of contact. The asset's organization determines whose rules of governance apply to the asset.

After an asset is created, it is sometimes necessary to change its ownership. For example:

- You may want to transfer an asset to another user if the original owner leaves the company, transfers to another position, or is otherwise unable to continue serving as the owner of an asset.
- You may want to transfer ownership of an asset to another organization when the asset reaches a point in its lifecycle where it is managed by a different group of users. When a service moves into production, for example, you might want to transfer it to your operations organization.

The following general guidelines apply when changing the ownership of an asset:

- The asset must not belong to the default user (nor can you move an asset to the default user).
- The asset must not be in a pending state (for example, awaiting approval) or have a consumer registration request pending for it.
- The asset must not be a component of a composite asset. If an asset is a component of another asset, you can move it only by moving the root asset to which it belongs.
- The asset must not be an instance of the asset type: REST Service, XML Service, Virtual REST Service, Virtual XML Service.
- The asset cannot be moved to an inactive user.

This section provides procedures for transferring assets to a different user or a different organization.

Note:

If you want to transfer an asset to a different user and a different organization at the same time, use the procedures for changing user ownership. These procedures allow you to optionally change the asset's organization in addition to its owner.

Changing User Ownership of Asset

To transfer asset ownership of a user, you must have the Manage Organizations permission or at least the Manage Users permission in CentraSite.

Note:

Users who have the Manage Organizations permission have the Manage Users permission by implication.

➤ **To change the user ownership of assets**

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the **Assets** pane, right-click an asset whose ownership you want to change, and then click **Change Owner**.

You can also select multiple assets, click the **Actions** menu, and click **Change Owner**

Note:

You can also initiate this step using the **Actions** menu in the details page of an asset.

3. In the **Change Owner** dialog box, select the user to whom you want to transfer ownership of the asset.

To filter the list, type a partial string in the **Search** field. CentraSite applies the filter to the **Name** column.

Examples

String	Description
b	Displays names that contain b
bar	Displays names that contain bar
%	Displays all users and groups

4. To transfer the assets to a different organization, select the organization from the **Organization** list box.

Only organizations for which the user has Create Assets permission are available for selection.

Note:

If you do not select an organization from the **Organization** list box, the selected assets remain in their current organizations. CentraSite does not automatically transfer the assets to the selected user's organization.

Important:

If you have selected several assets where one or more of them are predefined assets, such as UDDI Services, you can use the **Change Owner** button to change the ownership of all of the selected assets. However, as you are not allowed to change lifecycle state of predefined assets, only assets you have permission for are changed.

5. Click **OK**.

Changing Organization Ownership of Asset

To transfer asset ownership of an organization, you must have the Manage Organizations permission.

➤ **To change the organization ownership of assets**

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the **Assets** pane, right-click an asset whose ownership you want to change, and click **Change Organization**.

You can also select multiple assets, click the **Actions** menu, and click **Change Organization**.

Note:

You can also initiate this step using the **Actions** menu in the details page of an asset.

3. In the **Change Organization** dialog box, select the organization to which you want to transfer ownership of the asset.

To filter the list, type a partial string in the **Search** field. CentraSite applies the filter to the **Name** column.

Examples

String	Description
b	Displays names that contain b
ban	Displays names that contain ban
%	Displays all organizations

4. To transfer the assets to a different organization, select the organization from the **Organization** list box.

Only organizations for which the user has `Create Assets` permission are available for selection.

Note:

If you do not select an organization from the **Organization** list box, the selected assets remain in their current organizations. CentraSite does not automatically transfer the assets to the selected user's organization.

Important:

If you have selected several assets where one or more of them are predefined assets, such as UDDI Services, you can use the **Change Organization** button to change the ownership of all of the selected assets. However, as you are not allowed to change lifecycle state of predefined assets, only assets you have permission for are changed.

5. Click **OK**.

Downloading Asset

CentraSite Control offers two methods of retrieving the source files of CentraSite assets, namely *exporting* and *downloading*. The source file is the file that was imported into CentraSite in order to create the registry entry for the asset. For example, the source file for a web service asset is the service's WSDL file. The source file for an XML schema asset is its schema file. The difference between exporting and downloading is as follows:

- The *export* feature creates a zip file containing one or more assets from the repository, as well as all associated registry objects.
- The *download* feature creates a zip file containing just the source file of a single asset from the repository, without any of the associated registry objects. If the source file refers to other source files in the repository (for example, a WSDL file can reference XML schema files), the referenced files will also be included in the zip file. If the asset refers to files in the Supporting Document Library, these can optionally be included in the zip file.

If an asset was not created by an importer, but was instead created from scratch without using a source file, the download feature can still be activated. In this case, however, the downloaded zip file does not contain an asset source file but instead only contains files from the Supporting Document Library that are attached to the asset.

When downloading an asset, keep the following points in mind:

- To download any given asset, you must belong to a role that has the "Manage Assets" permission for the organization in which the asset resides.
- If you use the download feature to create a zip file, it contains only the files that you have permission to view. The default location to which the zip file is downloaded is `My Documents\Downloads`.
- The asset that you want to download must belong to an asset type for which there is an importer. The importer can be either one of the predefined importers or a user-defined importer.
- *This is of specific relevance to REST and XML based service assets.* Beginning with version 9.7, CentraSite supports the enhanced interface for REST services (in contrast, earlier versions of

CentraSite supported a standardized interface for REST services). Note that the standardized REST service interface that was implemented by versions of CentraSite prior to version 9.7 is not compatible with the enhanced interface that is implemented by current version of CentraSite. Documentation of the prior REST service interface is available to CentraSite customers who have a current maintenance contract in CentraSite.

- If you have REST services that were created from a previous version of CentraSite or if you are using the current version of CentraSite Business UI, you can only view details of these services in CentraSite. Keep in mind that you cannot download schemas from REST services using the CentraSite user interface (not even if you belong to the CentraSite Administrator role). You will only be able to download schemas from REST services using the CentraSite Business UI.

Performing Zip Download

The asset that you want to download must belong to an asset type for which there is an importer. The importer can be either one of the predefined importers or a user-defined importer.

When attaching documents to an asset, keep the following points in mind:

- If you use the download feature to create a zip file, it contains only the files that you have permission to view. The default location to which the zip file is downloaded is `My Documents\Downloads`.
- The asset that you want to download must belong to an asset type for which there is an importer. The importer can be either one of the predefined importers or a user-defined importer.

> To download an asset and its associated files

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the **Assets** pane, right-click an asset for which you want to generate a new version, and click **Download Documents**.

You can also select multiple assets, click the **Actions** menu, and click **Download Documents**.

Note:

You can also initiate this step from the **Actions** menu that is located in the upper-right corner of the asset details page.

3. In the **Download Documents** dialog box, select the **Include Supporting Documents** check box to include attached documents from the Supporting Document Library.
4. Click **OK**.

This starts the creation of asset zip file.

Downloading Documents from Supporting Document Library

➤ To download documents directly from the SDL

1. In CentraSite Control, go to **Asset Catalog > Supporting Documents**.
2. In the **Folders** pane, select the organization whose documents you want to download.

A list of documents belonging to the selected organization is displayed in the **Supporting Documents** pane.

3. Select one or more documents you want to download.
4. Click **Download**.

Note:

If you have selected more than one document, the output is directed to a zip file.

Downloading a Single Document from Supporting Document Library

➤ To download an attached document from the SDL

1. In CentraSite Control, go to **Asset Catalog > Browse**.

A list of currently defined assets is displayed in the **Assets** pane.

2. In the **Assets** pane, right-click an asset to which the supporting document is attached, and click **Details**.

You can also select multiple assets, click the **Actions** menu, and click **Details**.

3. In the asset details page, click the **Specification/Details/ Technical Details** tab.
4. Locate the document, and click the corresponding **Download** button.
5. In the **Download** dialog box, specify a location in the file system to store the supporting document, and click **OK**.

Downloading WSDL or XSD Document from Service or XML Schema Asset

➤ To download a WSDL or XSD document from Service or XML Schema asset

1. In CentraSite Control, go to **Asset Catalog > Browse**.

2. In the **Assets** pane, right-click an asset to which the supporting document is attached, and click **Details**.

You can also select multiple assets, click the **Actions** menu, and click **Details**.

3. In the asset details page, click the **Summary** tab.

4. Click the **WSDL/URL** hyperlink.

This opens the details of the **WSDL/URL** file.

5. Click the **Download** button.

6. In the **Download** dialog box, specify a location in the file system to store the supporting document, and click **OK**.

If the WSDL or XSD file includes a reference to another file (usually a relative address) in the repository, then this reference is changed to an absolute repository address.

Managing Supporting Documents

Any user with Create Assets permission for an organization can upload documents to that organization's supporting document library. By default, everyone in your organization is permitted to view the documents that you upload to your organization's supporting document library. To enable users in other organizations to view documents, you must grant them View permission on the document.

The Supporting Document Library (SDL) contains the collection of documents that you can associate with an asset. The documents that make up your organization's supporting document library reside in the document repository, which is the physical data store in which CentraSite maintains file-like objects.

Besides the documents in your organization's supporting document library, the document repository stores other large, file-like objects such as XML schemas, WSDL files, and report templates. However, when you interact with the supporting document library using CentraSite Control, you will only the portion of the repository that comprises the supporting document library for an organization. You will not other files that reside in the repository.

You use folders to group the documents within your organization's supporting document library. A folder can contain sub-folders or documents, and sub-folders can contain further sub-folders or documents, and so on.

Adding Folder to Supporting Document Library

> To add a new folder to the library

1. In CentraSite Control, go to **Asset Catalog > Supporting Documents**.

A list of folders and documents that are currently stored for your organization is displayed in the **Supporting Document Library** page.

2. In the **Folders** pane, right-click an existing folder to which you want to add a sub-folder, and click **Add Folder**.
3. In the **Add Folder** dialog box, type a name for the new folder.

The name can contain any combination of characters (including spaces), and must be unique within an organization. The name must not exceed 256 characters.

4. Click **OK**.

The newly created folder is listed in the **Folders** pane.

Deleting Folder in Supporting Document Library

Important:

Consider the following guidelines when deleting a folder in the Supporting Document Library:

- CentraSite does not allow you to delete a folder if it contains documents that are currently attached to an asset.
- If none of the documents in the folder are attached to an asset, CentraSite deletes the folder and all documents contained.

> To delete a folder in the library

1. In CentraSite Control, go to **Asset Catalog > Supporting Documents**.

A list of folders and documents that are currently stored for your organization is displayed in the **Supporting Document Library** page.

2. In the **Folders** pane, right-click a folder you want to delete, and click **Delete**.

When you are prompted to confirm the delete operation, click **OK**.

Adding Documents to Supporting Document Library

The document to upload must reside on the file system of the computer where your browser is running. You cannot upload a document from a URL.

> To add a supporting document to the library

1. In CentraSite Control, go to **Asset Catalog > Supporting Documents**.

A list of folders and documents that are currently stored for your organization is displayed in the **Supporting Document Library** page.

2. In the **Supporting Documents** pane, click **Add Document**.

3. In the **Add Document** dialog box, type the appropriate information for each of the displayed data fields.

In this field...	Do the following...
Folder	Specify the folder where the new document will be stored. This is by default the name of the folder you selected. You can use the Browse button to navigate to the required folder.
File	Specify the file that you want to upload to the supporting document library from the computer's file system. You can use the Browse button to navigate to the required folder.
Name	Type a name for the new document. The name can contain any combination of characters (including spaces), and must be unique within a folder. This name appears when a user displays the list of documents in the Supporting Document Library page. The name should be meaningful. Name must not exceed 256 characters.
Description	<i>Optional.</i> Type a short description for the new document. This description appears when a user displays the list of documents in the Supporting Document Library page. Description cannot exceed 4000 characters.

4. Click **OK**.

This adds the document to the Supporting Document Library.

Viewing Details of the Supporting Document Library

After you have added a document to the supporting library, you can view the details of the stored document.

» To view the details of a supporting document in the library

1. In CentraSite Control, go to **Asset Catalog > Supporting Documents**.

A list of folders and documents that are currently stored for your organization is displayed in the **Supporting Document Library** page.

2. In the **Folders** pane, click the folder whose document details you want to view.

A list of documents that are stored in the selected folder is displayed in the **Supporting Documents** pane.

3. Right-click a document whose details you want to view, and click **Details**.

CentraSite displays the details page of the selected document with the following information:

Tab	Description
Permissions	Displays the current permission settings for the document.
Attached To	Displays the list of assets, if any, to which the document is attached.
Versions	Displays the existing versions of the document.

Note:

Due to the current implementation, which creates supporting documents in a two-phase process, two version numbers are assigned when you add a supporting document. Thus, the document versions 1.0 and 2.0 are displayed for a newly added document.

Replacing Documents in Supporting Document Library

➤ To replace a supporting document in the library

1. In CentraSite Control, go to **Asset Catalog > Supporting Documents**.

A list of folders and documents that are currently stored for your organization is displayed in the **Supporting Document Library** page.

2. In the **Supporting Documents** pane, right-click a document you want to replace, and click **Details**.
3. In the **Edit Document Details** dialog box, click **Upload File**.
4. In the **Create New Version** dialog box, specify the file you want to upload to the library.
5. Click **OK** to upload the document.

Note that a version history of the document is visible in the **Versions** profile of the document's detail page. You cannot, however, revert to an older version of the document, since the supporting document library only stores the most recent version of the document.

Moving Document to Another Folder

➤ To replace a supporting document in the library

1. In CentraSite Control, go to **Asset Catalog > Supporting Documents**.

A list of folders and documents that are currently stored for your organization is displayed in the **Supporting Document Library** page.

2. In the **Supporting Documents** pane, right-click a document you want to replace, and then click **Details**.

This opens the **Edit Document Details** dialog box.

3. In the **Edit Document Details** dialog box, click **Upload File**.

This opens the **Create New Version** dialog box.

4. In the **Create New Version** dialog box, specify the file you want to upload to the library.
5. Click **OK** to upload the document.

Note that a version history of the document is visible in the **Versions** profile of the document's detail page. You cannot, however, revert to an older version of the document, since the supporting document library only stores the most recent version of the document.

Finding Assets Referring to Document in Supporting Document Library

➤ To display the list of assets that refer to a supporting document in the library

1. In CentraSite Control, go to **Asset Catalog > Supporting Documents**.

A list of folders and documents that are currently stored for your organization is displayed in the **Supporting Document Library** page.

2. In the **Folders** pane, click the folder whose document details you want to view.

A list of documents that are stored in the selected folder is displayed in the **Supporting Documents** pane.

3. Right-click a document whose list of assets you want to view, and then click **Details**.

This opens the details page of the selected document.

4. Click the **Attached To** tab.

A list of assets to which the supporting document is attached is displayed.

As just described, there is a tab labeled **Attached To** for each supporting document. Note that when you display the set of supporting documents in a folder, there is also a display column labeled **Attached To** that you can optionally select or cancel the selection with the column chooser. This display column shows the assets to which each supporting document is attached. If the list of assets is too long to be fully displayed on the screen, the ellipsis ("...") indicates that some asset names are not displayed. You can view the whole list of assets by moving the cursor over the visible assets. This causes the whole list to be displayed as rollover text.

CAUTION:

If a folder contains many supporting documents and if the supporting documents are attached to many assets, CentraSite might take some time to retrieve all of the information required for the **Attached To** display column. Therefore, you might prefer to keep the

Attached To column deselected and instead view the contents of the **Attached To** tab for an individual supported document, as described above.

Deleting Documents from Supporting Document Library

CentraSite does not allow you to delete a document that is attached to an asset. You can only delete unattached documents.

➤ To delete a document from the library

1. In CentraSite Control, go to **Asset Catalog > Supporting Documents**.

A list of folders and documents that are currently stored for your organization is displayed in the **Supporting Document Library** page.

2. In the **Supporting Documents** pane, select one or more documents to delete, and then click **Delete**.

When you are prompted to confirm the delete operation, click **OK**.

Managing Assets through Command Line Interface

This section describes operations you can perform to manage assets, such as, web services, REST services, OData services, and Applications through Command Line Interface.

Importing an Asset

An importer generates a catalog entry for an asset from a particular type of input file. For example, the Web service importer installed with CentraSite reads a WSDL file and from it, generates an asset instance for the service that the WSDL describes. In most cases, the importer also uploads the input file to the CentraSite repository and links the file to the asset. When you import a Web service from a WSDL file, for example, the importer copies the WSDL file into the repository and then links the file to the asset instance for the Web service.

The following table lists the importers installed with CentraSite and identifies the types of files they require as input:

To import this asset...	You must supply this type of file...
Web Service (including Abstract Service)	Web Service Definition Language (WSDL) file.
REST Service	RESTful API Modeling Language (RAML) file. Swagger file.
OData Service	Entity Data Model XML (EDMX) file.

To import this asset...	You must supply this type of file...
XML Schema	XML Schema Definition (XSD) file.
BPEL	Business Process Execution Language for Web Services (BPEL) file.
Process	XML Process Definition Language (XPDL) file.
Archive	A file that was previously exported from CentraSite Control.

Each importer includes a command line tool, which allows it to be executed from the command line `CentraSiteCommand.cmd` (Windows) or a terminal `CentraSiteCommand.sh` (UNIX) of CentraSite. The command is followed by a list of appropriate input parameters and options. The parameters are key-value pairs, where the key starts with a hyphen. An option is a switch that modifies the behavior of the command and is preceded by a hyphen. Using the `-help` option lists all available input parameters and options with a short description.

Importing Web Service

Pre-requisites:

To import a Web Service through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `ImportWSDL` for this purpose.

> To import a Web Service

- Run the command `ImportWSDL`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>ImportWSDL [-h <host>] [-p <port>] [-dburl <dburl>] -user <user> -password <pwd> -w <wsdlFile> [-accessuser <u>] [-accesspassword <p>] [-o <organization>] [-project <name>] [-name <name>] [-attach <id>] [-v <user-version>] [-reuse] [-help] [-forcenew] [-createversion]`

The input parameters and options are:

Parameter	Description
<code>-h <host></code>	The host name or IP address of the computer where the CentraSite registry or repository component is running. If you omit this parameter, the importer assumes that the registry or repository is running on <code>localhost</code> .
<code>-p <port></code>	The port number on which the CentraSite registry or repository is configured to listen for incoming requests. If you omit this parameter, the importer assumes that the registry or repository is listening on the default port, <code>53307</code> .

Parameter	Description
<code>-dburl <dburl></code>	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
	Note: If the registry or repository is running on a different machine and port number, you can use this parameter to specify its location instead of using the individual <code>-h</code> and <code>-p</code> parameters. (If you specify the <code>-dburl</code> parameter with the <code>-h</code> and <code>-p</code> parameters, the <code>-h</code> and <code>-p</code> parameters is ignored.)
<code>-user <user></code>	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
<code>-password <pwd></code>	The password for the registered CentraSite user identified by the parameter <code>-user</code> .
<code>-w <wsdlFile></code>	The absolute or relative path to the WSDL file that you want to import. If relative, the path should be relative to the location from where the command is executed.
<code>-accessuser <u></code>	(Optional). The username for accessing the WSDL file.
<code>-accesspassword <p></code>	(Optional). The password of the user identified by the parameter <code>-accessuser</code> for accessing the WSDL file.
<code>-o <organization></code>	(Optional). The name of the organization into which you want to import the web service. If the organization name contains a space, enclose the name in double-quotes.
<code>-project <name></code>	(Optional). The project name of an existing project for Web service.
<code>-name <name></code>	(Optional). The display name of the abstract service.
<code>-attach <id></code>	(Optional). Attach the WSDL to the existing service with an unique identifier.
<code>-v <user-version></code>	(Optional). The user-defined version number for the service.
<code>-reuse</code>	(Optional). Reuse the imported files instead of overwriting any data.
<code>-help</code>	(Optional). Display the full description of <code>ImportWSDL</code> command (with detailed parameters and options description).
<code>-forcenew</code>	(Optional). Force creating new asset instance of the Web service instead of updating an existing service.
<code>-createversion</code>	(Optional). Create a new version of existing service before the import.

Example (all in one line):

The command for importing a WSDL into CentraSite running on *localhost* with an administrator account *INTERNAL\Admin* and password *AdminPW* would be:

```
C:\SoftwareAG\CentraSite\utilities>ImportWSDL -dburl
http://localhost:53307/CentraSite/CentraSite -user INTERNAL\Admin -password AdminPW
-w c:\MyWSDL.wSDL
```

Importing REST Service

Pre-requisites:

To import a REST Service through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `import Service` to import a REST service using RAML or Swagger file.

> To import a REST Service

- Run the command `import Service`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand import Service [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> [-organization <ORGANIZATION>] -type <TYPE> -serviceUri <SERVICE-URI> [-serviceUser <SERVICE-USER>] [-servicePassword <SERVICE-PASSWORD>] [-update <ASSET-ID>] [-imports <IMPORTS-DESCRIPTOR>] [-batch <BATCH-MODE>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
ORGANIZATION	(Optional). The organization to which the newly imported REST service is long.
TYPE	The type of REST service you want to import. Possible values are: <ul style="list-style-type: none"> ■ <code>RAML</code> - Reads a RAML specification file and from it, creates a REST Service with RAML metadata. ■ <code>Swagger</code> - Reads a Swagger specification file and from it, creates a REST Service with Swagger metadata.

Parameter	Description
SERVICE-URI	<p>The location of a specification file that you want to use for importing the REST service in CentraSite.</p> <ul style="list-style-type: none">■ If the import file resides on a remote network, specify its URL.■ If the import file resides on your local system, specify the absolute or relative path to the file. If relative, the path should be relative to the location from where the command is executed.
SERVICE-USER	<p>(Optional). If you have specified a URL in the above parameter SERVICE-URI, and the import file you want to access through the URL requires user authentication, enter a user ID that CentraSite is to use for authentication at the URL site.</p>
SERVICE-PASSWORD	<p>(Optional). The password that CentraSite is to use for authentication at the URL site.</p>
ASSET-ID	<p>(Optional). The ID of the REST service you want to update. You can specify the UDDI key of the service using an optional prefix "uddi:". For example:</p> <pre>uddi:207ff1cc-25c5-544c-415c-d98ea91060c</pre>
IMPORTS-DESCRIPTOR	<p>(Optional). The absolute or relative path to the imports descriptor file that contains all referenced objects to which the RAML/Swagger import file has an association.</p> <p>The imports descriptor file contains a map of key-value pairs and defines the referenced objects as <import-alias>=<import-uri>.</p>
BATCH-MODE	<p>(Optional). Determines how referenced schemas are handled when the schema that is referred to in the imports descriptor file does not exist in the registry. Possible values are true and false.</p> <ul style="list-style-type: none">■ If the parameter value is set to true, then CentraSite ignores any missing referenced schemas.■ If the parameter value is set to false, then you is prompted at runtime to specify the referenced schema file's location. <p>The default value is false.</p>

Example (all in one line):

The command for importing a RAML specification into CentraSite with an administrator account *INTERNAL\Admin* and password *AdminPW* would be:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd import Service -url
http://localhost:53307/CentraSite/CentraSite -user INTERNAL\Admin -password AdminPW
-organization "Default Organization" -type RAML -serviceUri
```

```
C:/include-resource-types.yaml -serviceUser INTERNAL\Claire -servicePassword Clairepwd
-imports C:/imports.properties -batch true
```

The response to this command could be:

```
Executing the command : import Service
=====
The service has been imported successfully with:
Name: RAML RESTful API
Id: b7377c94-2993-11e5-bad6-e9f07cabfe13
Organization: Default Organization
=====
...
Successfully executed the command : import Service
```

Note:

To update an existing REST Service in CentraSite, you can use the `import Service` command and provide an edited specification file as input to the command along with the key that identifies the REST service that you want to update.

Importing OData Service

Pre-requisites:

To import an OData Service asset to the catalog through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `import Service` for this purpose.

Note:

To update an existing OData service in CentraSite, you can use the `import Service` command and provide an edited EDMX file as input to the command along with the key that identifies the OData service that you want to update.

> To add an OData Service asset

- Run the command `import Service`.

The syntax is of the format `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand import Service [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -organization <ORGANIZATION> -type <TYPE> -serviceUri <SERVICE-URI> [-serviceUser <SERVICE-USER>] [-servicePassword <SERVICE-PASSWORD>] [-update <ASSET-ID>] [-imports <IMPORTS-DESCRIPTOR>] [-batch <BATCH-MODE>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .

Parameter	Description
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
ORGANIZATION	(Optional). The organization to which the newly imported OData service will belong.
TYPE	The type of service you want to import. Possible values are: <code>RAML</code> , <code>Swagger</code> , and <code>OData</code> . Select OData . This reads the Service URL or the EDMX file and from it, creates an OData service with the corresponding metadata.
SERVICE-URI	The location of an EDMX file that you want to use for importing the OData service in CentraSite. <ul style="list-style-type: none">■ If the import file resides on a remote network, specify its URL.■ If the import file resides on your local system, specify the absolute or relative path to the file. If relative, the path should be relative to the location from where the command is executed.
SERVICE-USER	(Optional). If you have specified a URL in the above parameter <code>-serviceUri</code> , and the import file you want to access through the URL requires user authentication, type a user ID that CentraSite is to use for authentication at the URL site.
SERVICE-PASSWORD	(Optional). The password that CentraSite is to use for authentication at the URL site.
ASSET-ID	(Optional). The ID of the OData service you want to update. You can specify the UDDI key of the service using an optional prefix <code>uddi:</code> . For example: <pre>uddi:207ff1cc-25c5-544c-415c-d98ea91060c</pre>
IMPORTS-DESCRIPTOR	(Optional). The absolute or relative path to the imports descriptor file that contains all referenced objects to which the EDMX import file has an association. The imports descriptor file contains a map of key-value pairs and defines the referenced objects as <code><import-alias>=<import-uri></code> .
BATCH-MODE	Note: CentraSite does not support this parameter for OData services.

Example (all in one line):

The command for importing an EDMX specification into CentraSite with an administrator account *INTERNAL\Admin* and password *AdminPW* would be:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd import Service -url
http://localhost:53307/CentraSite/CentraSite -user INTERNAL\Admin -password AdminPW
-organization "Default Organization" -type OData -serviceUri C:/TripPinService.edmx
-serviceUser INTERNAL\Claire -servicePassword Clairepwd -imports C:/imports.properties
```

The response to this command could be:

```
Executing the command : import Service
=====
The service has been imported successfully with:
Name: TripPin OData Service
Id: b7377c94-2993-11e5-bad6-e9f07cabfe13
Organization: Default Organization
=====
...
Successfully executed the command : import Service
```

Importing XML Schema

Pre-requisites:

To import an XML Schema through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `ImportSchema` for this purpose.

➤ To import an XML Schema

- Run the command `ImportSchema`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>ImportSchema -s <schema-file> [-help] [-accessuser <u>] [-accesspassword <p>] [-o <organization>] [-v <user-version>] [-name <name>] [-attach <id>] [-reuse] [-newversion <current|all>] [-h <host>] [-p <port>] [-dburl <dburl>] -user <user> -password <pwd>`

The input parameters and options are:

Parameter	Description
-h <host>	The host name or IP address of the computer where the CentraSite registry or repository component is running. If you omit this parameter, the importer assumes that the registry or repository is running on <code>localhost</code> .
-p <port>	The port number on which the CentraSite registry or repository is configured to listen for incoming requests. If you omit this parameter, the importer assumes that the registry or repository is listening on the default port, 53307.

Parameter	Description
<code>-dburl <dburl></code>	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> . Note: If the registry or repository is running on a different machine and port number, you can use this parameter to specify its location instead of using the individual <code>-h</code> and <code>-p</code> parameters. (If you specify the <code>-dburl</code> parameter with the <code>-h</code> and <code>-p</code> parameters, the <code>-h</code> and <code>-p</code> parameters is ignored.)
<code>-user <user></code>	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
<code>-password <pwd></code>	The password for the registered CentraSite user identified by the parameter <code>-user</code> .
<code>-s <schema-file></code>	The absolute or relative path to the schema file that you want to import. If relative, the path should be relative to the location from where the command is executed.
<code>-accessuser <u></code>	(Optional). The username for accessing the XML schema file.
<code>-accesspassword <p></code>	(Optional). The password of the user identified by the parameter <code>-accessuser</code> for accessing the XML schema file.
<code>-o <organization></code>	(Optional). The name of the organization into which you want to import the XML schema. If the organization name contains a space, enclose the name in double-quotes.
<code>-name <name></code>	(Optional). The display name of the schema object.
<code>-attach <id></code>	(Optional). Attach the XML schema to the existing service with an unique identifier.
<code>-v <user-version></code>	(Optional). The user-defined version number for the XML schema.
<code>-reuse</code>	(Optional). Reuse the imported files instead of overwriting any data.
<code>-help</code>	(Optional). Display the full description of <code>ImportSchema</code> command (with detailed parameters and options description).
<code>-createversion</code>	(Optional). Create a new version of existing XML schema before the import. The possible values are: <ul style="list-style-type: none">■ <code>current</code>■ <code>all</code>

Example (all in one line):

The command for importing a XML schema into CentraSite running on host *myhost* registered under the organization *MyOrg* with an administrator account *INTERNAL\Admin* and password *AdminPW* would be:

```
C:\SoftwareAG\CentraSite\utilities>ImportSchema -dburl
http://localhost:53307/CentraSite/CentraSite -user INTERNAL\Admin -password AdminPW
-h myhost -p 53307 -s c:\MySchema.xsd -o MyOrg
```

Importing BPEL Process

Pre-requisites:

To import a BPEL process through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `ImportBPEL` for this purpose.

» To import a BPEL process

- Run the command `ImportBPEL`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>ImportBPEL -file bpeFile [-help] [-project <project>] [-nowarning] [-h <host>] [-p <port>] [-dburl <dburl>] -user <user> -password <pwd>`

The input parameters and options are:

Parameter	Description
<code>-h <host></code>	The host name or IP address of the computer where the CentraSite registry or repository component is running. If you omit this parameter, the importer assumes that the registry or repository is running on <code>localhost</code> .
<code>-p <port></code>	The port number on which the CentraSite registry or repository is configured to listen for incoming requests. If you omit this parameter, the importer assumes that the registry or repository is listening on the default port, <code>53307</code> .
<code>-dburl <dburl></code>	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .

Note:

If the registry or repository is running on a different machine and port number, you can use this parameter to specify its location instead of using the individual `-h` and `-p` parameters. (If

Parameter	Description
	you specify the <code>-dburl</code> parameter with the <code>-h</code> and <code>-p</code> parameters, the <code>-h</code> and <code>-p</code> parameters is ignored.)
<code>-user <user></code>	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
<code>-password <pwd></code>	The password for the registered CentraSite user identified by the parameter <code>-user</code> .
<code>-file bpelFile</code>	The absolute or relative path to the BPEL file that you want to import. If relative, the path should be relative to the location from where the command is executed.
<code>-project <project></code>	(Optional). The project name of an existing project for the BPEL process.
<code>-nowarning</code>	(Optional). Suppresses the warning message that the importer issues if the BPEL process references a BPEL Partner Link Type that refers to a web service that is not present in the CentraSite registry.
<code>-help</code>	(Optional). Display the full description of <code>ImportBPEL</code> command (with detailed parameters and options description).

Example (all in one line):

The command for importing a BPEL process file into CentraSite running on *localhost* with an administrator account *INTERNAL\Admin* and password *AdminPW* would be:

```
C:\SoftwareAG\CentraSite\utilities>ImportBPEL -dburl
http://localhost:53307/CentraSite/CentraSite -user INTERNAL\Admin -password AdminPW
-file c:/tmp/MyBPEL.bpel
```

Importing XPD L File

Pre-requisites:

To import an XPD L File through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `ImportXPDL` for this purpose.

➤ To import an XPD L File

- Run the command `ImportXPDL`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>ImportXPDL -file <filename> [-h <host>] [-p <port>] -user <user> -password <pwd>`

The input parameters are:

The input parameters and options are:

Parameter	Description
-h <host>	The host name or IP address of the computer where the CentraSite registry or repository component is running. If you omit this parameter, the importer assumes that the registry or repository is running on localhost.
-p <port>	The port number on which the CentraSite registry or repository is configured to listen for incoming requests. If you omit this parameter, the importer assumes that the registry or repository is listening on the default port, 53307.
-dburl <dburl>	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
	Note: If the registry or repository is running on a different machine and port number, you can use this parameter to specify its location instead of using the individual -h and -p parameters. (If you specify the -dburl parameter with the -h and -p parameters, the -h and -p parameters is ignored.)
-user <user>	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
-password <pwd>	The password for the registered CentraSite user identified by the parameter -user.
-file <filename>	The absolute or relative path to the XPDL file that you want to import. If relative, the path should be relative to the location from where the command is executed.

Example (all in one line):

The command for importing a XPDL process file into CentraSite running on host *myhost* and default port 53307 with an administrator account *INTERNAL\Admin* and password *AdminPW* would be:

```
C:\SoftwareAG\CentraSite\utilities>ImportXPDL -dburl
http://localhost:53307/CentraSite/CentraSite -user INTERNAL\Admin -password AdminPW
-h myhost -file c:/tmp/MyProc.xpdl
```

Importing Archive

Pre-requisites:

To import an archive through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `ImportArchive` for this purpose.

➤ To import an archive from the command line

- Run the command tool `ImportArchive`.

```
The syntax is of the format: C:\SoftwareAG\CentraSite\utilities>ImportArchive.cmd
<CentraSite URL> <archive filename> <username> <password> [-help] [-setreplace]
[-keepowner] [-setowner name] [-keeporganization] [-keepplcmstate]
[-removemissingreferences] [-importgroup] [-executewsdlpolicy] [-importorg id]
[-importorgname name] [-ignoreauthtokens] [-simulate] [-importkeys id[,id...]]
[-sequential [-minimizeaudits] [-listonly] [-skip cnt]]
```

The input parameters and options are:

Parameter	Description
CentraSite-URL	(Optional). The URL of the CentraSite registry/repository. For example, <code>http://localhost:53307/CentraSite/CentraSite</code>
archive-filename	The name of the exported archive (Zip) file. The archive file can contain an organization with its assets or can contain a set of objects that were exported from one or more organizations.
username	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
password	The password for the CentraSite user identified by the parameter <code>username</code> .
-help	(Optional). Display the full description of <code>ImportArchive</code> command (with detailed parameters and options description).
-executewsdlpolicy	(Optional). Execute the WSDL Regeneration policy for service import. By default, the command execution ignores the policy.
-ignoretypeversion	(Optional). Ignore the version check on asset types.
-importgroup	(Optional). Import the groups that include a single user.
-importkeys id[,id...]	(Optional). Import only those objects which have keys specified.
-importorg id	(Optional). Import the objects into the organization specified by the UUID.
-importorgname name	(Optional). Import the objects into the organization specified by the name.

Parameter	Description
-keeplcmstate	(Optional). Keep the LCM state of the object which is set at export.
-keepowner	(Optional). Keep the object owner instead of assigning the importing user.
-keeporganization	(Optional). Keep the organization of the imported objects instead of assigning the active one performing the message logging.
-listonly	(Optional). Print only the list of objects to be imported. The <code>-listonly</code> option can be used to adjust the <code>-skip</code> option. No updates will be performed with the <code>-sequential</code> option.
-minimizeaudits	(Optional). Import only first and last audit record for each object, along with the <code>-sequential</code> option.
-removemissingreferences	(Optional). Remove all missing associations that could cause dangling references.
-setowner name	(Optional). All the imported objects will be set to a specified user.
-setreplace	(Optional). Replace objects if already present in the target registry.
-simulate	(Optional). Simulate the import, however, do not updates the objects in the target registry.
-sequential	(Optional). Imports the objects sequentially in a reasonable order.
-skip cnt	(Optional). Skip records before importing, along with the <code>-sequential</code> option.
-ignoreauthtokens	(Optional). Ignore API keys and OAuth2 tokens of the imported assets.

Example (all in one line):

The command for importing an archive into CentraSite running on host *myhost*, and force a replacement of all assets from archive in the CentraSite registry, with an administrator account *INTERNAL\Admin* and password *AdminPW* would be:

```
C:\SoftwareAG\CentraSite\utilities>ImportArchive.cmd -setreplace
http://myhost:53307/CentraSite/CentraSite c:/tmp/export.zip INTERNAL\Admin AdminPW
```

Bulk Importing Assets

Pre-requisites:

To bulk import assets through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

Note:

Ensure that all of the necessary JAR files are located on `<CentraSiteInstall_Directory>/redist`.

Required JAR files for a CSV (.csv) file:

Download OpenCSV 2.4.0 jar from the URL <https://code.google.com/p/opencsv/downloads/detail?name=opencsv-2.4.jar&can=2&q=> and place it in the `redist` folder.

Required JAR files for an Excel (.xlsx) file:

Download Apache POI 3.13 from the URL <https://poi.apache.org/download.html> and place the following Jar files on `<CentraSiteInstall_Directory>/redist`.

- `poi*.jar`
- `poi-ooxml*.jar`
- `poi-ooxml-schemas*.jar`

Supported data types and their format:

- String
- Boolean
- Number
- URI/URL
- Multiline String
- Classification
- IP Address
- Email
- Date/Time
- Duration

Format for data type Duration:

Example: P1Y2M3DT4H5M

Format for data type Date:

- `yyyy-MM-dd`

Example: 2015-12-10

Format for data type Date and Time:

- `yyyy-mm-dd hh:mm:ss`

Example: 2015-12-10 12:04:10

- yyyy-mm-dd hh:mm

Example: 2015-12-10 12:04

Format for data type Time:

- hh:mm:ss

Example: 12:04:10

- hh:mm

Example: 12.10

CentraSite provides a Java tool named `CentraSiteAssetImporter.jar` for this purpose. The tool imports multiple assets from a CSV file or an Excel file (in the .xlsx format).

> To import multiple assets

- Run the Java tool `CentraSiteAssetImporter.jar`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd CentraSiteAssetImporter.jar [-url <CENTRASITE-URL>] -username <USERNAME> -password <PASSWORD> [-ownername <OWNER-NAME>] -qname <FULLY-QUALIFIED-NAME> -inputfile <FULL-PATH-TO-FILE> [-excelworksheet <EXCEL-WORKSHEET-NAME>] -namecolumn <ASSET-COLUMN-NAME> [-desccolumn <ASSET-DESC-COLUMN-NAME>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USERNAME	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USERNAME</code> .
OWNER-NAME	(Optional) . The domain in which the user wants to include the imported asset. If this parameter is not specified, the user identified by the parameter <code>USERNAME</code> will be the owner.
FULLY-QUALIFIED-NAME	The namespace of an asset type along with its schema name. For example: <code>{http://namespaces.DefaultOrganization.com/Schema}</code>

Parameter	Description
FULL-PATH-TO-FILE	The absolute or relative path to the CSV or Excel file. If relative, the path should be relative to the location from where the command is executed.
EXCEL-WORKSHEET-NAME	(Applicable only to Excel). The name of the Excel worksheet which contains the details of assets to be imported into CentraSite.
ASSET-COLUMN-NAME	The name of the column or header in Excel sheet or CSV file from which a display name would be assigned to the imported asset.
ASSET-DESC-COLUMN-NAME	The name of the column or header in Excel sheet or CSV file from which a brief description would be assigned to the imported asset.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd CentraSiteAssetImporter.jar
-url http://localhost:53307/CentraSite/CentraSite -username Administrator -password
manage -ownername sag\inosec1 -qname {http://namespaces.DefaultOrganization.com/Schema}
MyAssetType -filelocation c:\repository\assetdata.xlsx -namecolumn "Interface Name"
-excelworksheet Lowes.com -desccolumn "desc"
```

Deleting an Asset

Pre-requisites:

To delete an asset through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

In some circumstances, you may not be able to delete an asset because internal objects that reference it cannot be deleted. This can happen when there are internal references to an asset even though the asset is no longer in association with any other objects.

CentraSite provides a Java tool named `CentraSiteDeleteAsset.jar` for this purpose.

> To delete an asset

- Run the Java tool `CentraSiteDeleteAsset.jar`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd CentraSiteDeleteAsset.jar <arg1_CentraSite_DB_URL> <arg2_Administrator_User> <arg3_Administrator_User_Password> <arg4_UUID_of_asset_to_be_deleted>`

The input parameters are:

Parameter	Description
<arg1_CentraSite_DB_URL>	The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
<arg2_Administrator_User>	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
<arg3_Administrator_User_Password>	The password for the CentraSite user identified by the parameter <arg2_Administrator_User>.
<arg4_UUID_of_asset_to_be_deleted>	The ID of the asset you want to delete. You can specify the UDDI key of the asset using an optional prefix "uddi:".

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd CentraSiteDeleteAsset.jar
http://localhost:53307/CentraSite/CentraSite DOMAIN\\admin pAsSw0rD
"uddi:3fb8dbd0-6122-11e5-933f-e8830d260cc1"
```

Deleting Assets

Pre-requisites:

To delete all the assets in CentraSite through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

If you are migrating a set of assets from CentraSite version 8.2 to version 9.7 or above, the promoted assets in the target organization cannot be deleted.

CentraSite provides a Java tool named `CentraSiteDeleteAllAsset.jar` for this purpose. This tool will delete all the assets from an organization in CentraSite.

Note:

Before you run the tool, if there are assets of the Virtual Service type for deletion, make sure these assets are unpublished from the appropriate gateways.

> To delete assets

- Run the Java tool `CentraSiteDeleteAllAsset.jar`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd CentraSiteDeleteAllAsset.jar <arg1_CentraSite_DB_URL> <arg2_Administrator_User> <arg3_Administrator_User_Password> <arg4_AssetType> <arg5_OrganizationName> <arg6_OwnerName>`

The input parameters are:

Parameter	Description
<arg1_CentraSite_DB_URL>	The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
<arg2_Administrator_User>	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
<arg3_Administrator_User_Password>	The password for the CentraSite user identified by the parameter <arg2_Administrator_User>.
<arg4_AssetType>	Type of asset you want to delete. For assets other than the Service type and its variants, you must specify the schema name and its namespace. Example: <code>{http://namespaces.CentraSite.com/Schema}XMLSchema</code>
<arg5_OrganizationName>	Name of the (target) organization whose assets you want to delete.
<arg6_OwnerName>	Name of the user who owns the assets you want to delete.

Examples (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd CentraSiteDeleteAllAsset.jar
http://localhost:53307/CentraSite/CentraSite DOMAIN\admin pAsSw0rD
"{http://namespaces.CentraSite.com/Schema}XMLSchema" "TARGET_SERVICE" "*"

C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd CentraSiteDeleteAllAsset.jar
http://localhost:53307/CentraSite/CentraSite DOMAIN\admin pAsSw0rD "Service"
"TARGET_SERVICE" "*"
```

RAML to CentraSite REST API Mappings

This section describes how the RAML objects and properties (as described in the RAML specification) are mapped in the CentraSite REST data model. It shows the relationships between the RAML root section, parameters, body fields, and their REST data model. It also lists the relationships between RAML resources, methods, reference documentation, and security schemes, and how these values are mapped in REST API.

Note:

CentraSite supports RAML version 0.8.

The following tables list the correspondences (mappings) between the fields of a RESTful API and RAML document:

RAML Root Section Fields

The root section of the RAML definition describes the basic information of a RESTful API, such as its title, version, base URI, default media types, and common schema references.

RAML Root Section Field	REST API Field	Notes
RAML Document	REST Service	<p>The <code>RAML Document</code> property is represented by an asset of type <code>REST Service</code> in the CentraSite registry.</p> <p>The <code>RAML Document</code> is stored as a file attribute in the Documents field of the Specification profile in the REST API's details page.</p> <p>One or more RAML documents can be attached to a REST API. The attached documents are stored in the asset-specific WebDAV folder like the schema and WSDL documents.</p>
API Title	Name	The <code>API Title</code> property is represented by the Name field in the REST API's details page.
API Version	Version	<code>API Version</code> is represented by the Version field in the REST API's details page.
Base URI	Base URL	The <code>Base URI</code> property is represented by the Base URL field in the Technical Details profile.
baseUri Parameters		Currently, CentraSite does not support mapping <code>URI Parameter</code> property at the Base URL level.
Protocols	Base URLs	<p>In CentraSite, the <code>Protocols</code> property is represented by the multiple Base URL fields of the Technical Details profile.</p> <p>Example:</p> <p>A RAML Specification:</p> <pre>baseUri: http://products.api.apievangelist.com protocols:[HTTP, HTTPS]</pre> <p>Reads in CentraSite REST Model:</p> <pre>http://products.api.apievangelist.com https://products.api.apievangelist.com</pre>

RAML Root Section Field	REST API Field	Notes
Default Media Type	Request Content Type Response Content Type	In CentraSite, the Default Media Type property is represented by the Request Content-Type and Response Content-Type fields of the Technical Details in the REST API's details page.
Schemas	Schema	The Schemas property is represented by the Request Schema or Response Schema field of a REST Method.
URI Parameters (for baseUri)		Currently, CentraSite does not support the URI Parameter property at the Base URL level.
User Documentation	Description	The User Documentation property is represented by the Description field in the REST API's details page.

RAML Parameter Fields

The following tables show the mappings between various RAML parameter types, data types, and attributes and how they are used by REST API. The mappings are grouped into tables for RAML Parameter Types, RAML Parameter Data Types, and RAML Parameter Attributes:

RAML Parameter Types

RAML Parameter Field	REST API Field	Notes
URI Parameter	Parameter of type, Path	The URI Parameter property is represented by the Path option in the Parameter Type field of a REST Parameter, and is specifically defined at the Resource level. Any URI Parameter defined at the API level or Method level is not mapped in CentraSite.
Query Parameter	Parameter of type, Query-String	The QueryParameter property is represented by the Query-String option in the Parameter Type field of a REST Parameter.
Header	Parameter of type, Header	The Header property is represented by the Header option in the Parameter Type field of a REST Parameter.
Form Parameter	Parameter of type, Form	The Form Parameter property is represented by the Form option in the Parameter Type field of a REST Parameter.

RAML Parameter Data Types

RAML Data Type Field	REST API Field	Notes
string	String	The string property is represented by the String option in the Data Type field of a REST Parameter.
number	Number	The number property is represented by the Number option in the Data Type field of a REST Parameter.
integer	Number	The integer property is represented by the Number option in the Data Type field of a REST Parameter.
date	Date/Time	The date property is represented by the Date/Time option in the Data Type field of a REST Parameter.
boolean	Boolean	The boolean property is represented by the Boolean option in the Data Type field of a REST Parameter.
file	URL/URI	Currently, CentraSite does not support parameter of the data type file.

RAML Parameter Attributes

RAML Attribute Field	REST API Field	Notes
enum	Possible Values	The enum property is represented by the Possible Values field of a REST Parameter.
required	Required	The required property is represented by the Required field of a REST Parameter.
default	Default Value	The default property is represented by the Default Value field of a REST Parameter.
pattern		Currently, CentraSite does not support the pattern attribute.
minLength		Currently, CentraSite does not support the minLength attribute.
maxLength		Currently, CentraSite does not support the maxLength attribute.

RAML Attribute Field	REST API Field	Notes
minimum		Currently, CentraSite does not support the minimum attribute.
maximum		Currently, CentraSite does not support the maximum attribute.
example		Currently, CentraSite does not support the example attribute.
repeat		The repeat property is represented by the Array field of a REST Parameter.

RAML Resource Fields

RAML resource properties (data) are mapped to the REST API resource objects. The following table shows the resource mappings between a RAML definition and a REST API:

RAML Resource Field	REST API Field	Notes
Resource	REST Resource	The Resource property is represented by an asset of type REST Resource in the CentraSite registry.
Resource Relative Path	Resource Path	The Resource Relative Path property is represented by the Resource Path field of a REST Resource.
Display Name	Name	The Display Name property is represented by the Name field of a REST Resource.
Description	Description	The Description property is represented by the Description field of a REST Resource.
Template URIs	Resource Path with Path parameter	The Template URI property is represented by the Resource Path field which has the Path parameter appended to it, and is defined as a REST Resource object.
URI Parameters	Parameter of type, Path	The URI Parameter property is defined as a REST Parameter object of the type Path at the REST Resource level.
Base URI Parameters		Currently, CentraSite does not support mapping URI Parameter property at the Base URL level.
Resource Types and Traits		Limitation: Resource Types and Traits are not explicitly mapped to the REST API; instead,

RAML Resource Field	REST API Field	Notes
		propagated to the referencing REST Resources and REST Methods.

RAML Method Fields

The following tables show the mappings between various RAML method objects, bodies and responses, and how they are used by REST API. The mappings are grouped into tables for RAML Method Headers, RAML Method Body, and RAML Method Responses:

RAML Method Headers

RAML Method Header Field	REST API Field	Notes
Method	REST Method	The Method property is represented by an asset of type REST Method in the CentraSite registry.
Description	Description	Description property is represented by the Description field of a REST Method.
Header	Parameter of type "Header"	The Header property is defined as a REST Parameter object of the type Header at the REST Resource level.
Protocols		Currently, CentraSite does not support mapping Protocol property at the REST Method level.
Query Parameters	Parameter of type "Query-String"	The QueryParameters property is defined as a REST Parameter object of the type Query-String at the REST Resource level.

RAML Method Body

RAML Method Body Field	REST API Field	Notes
Body	Request	The Body property is defined as the Request Payload object. CentraSite automatically populates the Media type as Name of a REST Payload object.
Media type	Request Content-Type Response Content-Type	The Media type property is represented by the Request Content-Type or Response Content-Type field of a REST Payload object.

RAML Method Body Field	REST API Field	Notes
Schema	Schema	The Schema property is represented by the Request or Response Schema field of a REST Payload object.
Example	Example	The Example property is represented by the Request or Response Example field of a REST Payload object.

RAML Method Responses

RAML Method Response Field	REST API Field	Notes
HTTP status code	REST Payload	The HTTP status code property is defined as the REST Payload object of a REST Method. CentraSite automatically populates the HTTP status code as the Name of a REST Payload object.
Example	Response Example	The Example property is represented by the Response Example field of a REST Payload object.
Description	Description	The Description property is represented by the Response Description field of a REST Payload object.
Body	Response Content-Type	The Body property is represented by the Response Content-Type field of a REST Payload object.
Body media type	Response Content-Type	The Body Media type property is represented by the Response Content-Type field of the REST Payload.

RAML Resource Type and Trait Fields

All RAML resource types and traits properties are not explicitly mapped to the REST API. Instead, the RAML resource types are propagated to the referencing document structure named REST Resource and RAML traits are propagated to the document structure named REST Method.

RAML Security Scheme Fields

RAML security schemes are not explicitly mapped to the REST API. Instead, the RAML security schemes are propagated to the taxonomy structure named **Security Types**, and represented by the **Supported Access Token Types** property defined in the **API Portal Information** profile.

Security Type	Description
API Key	The API's authentication requires using an API key.
Basic Authentication	The API's authentication requires using Basic Access Authentication as described in RFC2617.
Digest Authentication	The API's authentication requires using Digest Access Authentication as described in RFC2617.
OAuth 1.0	The API's authentication requires using OAuth 1.0 as described in RFC5849.
OAuth 2.0	The API's authentication requires using OAuth 2.0 as described in RFC6749.
x-{other}	The API's authentication requires using another authentication method.

The supported security schemes along with their scheme reference name are mapped in the CentraSite registry. For more information about the security scheme, see the RAML specification.

Swagger to CentraSite REST API Mappings

This section describes how the Swagger objects and properties (as described in the Swagger specification) are mapped in the CentraSite REST data model. It shows the relationships between the Swagger root section, parameters, body fields, and their REST data model. It also lists the relationships between Swagger resources, methods, reference documentation, and security schemes, and how these values are mapped in REST API.

Note:

CentraSite supports Swagger version 2.0.

The following tables list the correspondences (mappings) between the fields of a RESTful API and Swagger document.

Swagger Root Section Fields

The Root section of the Swagger definition describes the basic information of a RESTful API, such as its title, external documents, security schemes and references.

Swagger Root Section Field	REST API Field	Notes
swagger		Currently, CentraSite does not support mapping the <code>swagger</code> version in the REST API's details page.
title	Name	The <code>title</code> property is represented by the Name field in the REST API's details page.
description	Description	The <code>description</code> property is represented by the Description field in the REST API's details page.
termsOfServiceUrl		Currently, CentraSite does not support mapping the <code>termsOfServiceUrl</code> at the API level.
contact		Currently, CentraSite does not support mapping the <code>contact</code> property at the API level.
license		Currently, CentraSite does not support mapping the <code>license</code> property at the API level.
licenseUrl		Currently, CentraSite does not support mapping the <code>licenseUrl</code> at the API level.
externalDocs		The <code>externalDocs</code> are not explicitly mapped to the REST API; instead, the external documentation is attached as a file attribute to the Specification profile in the REST API details page.
host		Currently, CentraSite does not support mapping the <code>host</code> property at the API level.
basePath		Currently, CentraSite does not support mapping the <code>basePath</code> property at the API level.
schemes		<p>The <code>schemes</code> are not explicitly mapped to the REST API; instead, the schemes (HTTP, HTTPS) are represented by the multiple base URLs in the Technical Details profile.</p> <p>Note that CentraSite does not support mapping the WS and WSS schemes.</p>
produces		<p>Currently, CentraSite does not support mapping the <code>produces</code> property at the API level.</p> <p>However, it is represented by the Response Content-Type field of a REST Method.</p>
consumes		Currently, CentraSite does not support mapping the <code>consumes</code> property at the consumes level.

Swagger Root Section Field	REST API Field	Notes
		However, it is represented by the Request Content-Type field of a REST Method.
paths	REST Resource	The paths property is represented by a collection of resources, wherein each of the resources have its path value mapped by the Name field of a REST Resource.
definitions		The definitions property is represented by the Request Schema or Response Schema field of a REST Method.
security		Currently, CentraSite does not support mapping the security at the API level. However, it is represented by the Supported Access Token Types field in the API Portal Information profile.

Swagger Parameter Fields

The following tables show the mappings between various Swagger parameter types and attributes and how they are used by REST API. The mappings are grouped into tables for Swagger Parameter Types, and Swagger Parameter Attributes.

Swagger Parameter Types

Swagger Parameter Field	REST API Field	Notes
URI Parameter	Parameter of type, Path	The URI Parameter property is represented by the "Path" option in the Parameter Type field of a REST Parameter.
Query Parameter	Parameter of type, Query-String	The QueryParameter property is represented by the "Query-String" option in the Parameter Type field of a REST Parameter.
Header	Parameter of type, Header	The Header property is represented by the "Header" option in the Parameter Type field of a REST Parameter.
Form Parameter	Parameter of type, Form	The Form Parameter property is represented by the "Form" option in the Parameter Type field of a REST Parameter.

Swagger Parameter REST API Field	Notes
Body Parameter	<p>Currently, CentraSite does not support mapping parameters of the type "Body".</p> <p>However, a body parameter is represented by the Sample Request field of a REST Method.</p>

Swagger Parameter Attributes

Swagger Attribute REST API Field	Notes	
name	Name	The <code>Display Name</code> property is represented by the Name field of a REST Parameter.
description	Description	The <code>Description</code> property is represented by the Description field of a REST Parameter.
in		<p>The <code>in</code> property is represented by the Parameter Type field of a REST Parameter.</p> <p>Currently, CentraSite does not support mapping parameters of the type "formData" and "body".</p>
required	Required	The <code>required</code> property is represented by the Required field of a REST Parameter.
type	Data Type	<p>The <code>type</code> property is represented by the Data Type field of a REST Parameter.</p> <p>Currently, CentraSite does not support mapping parameters of the data type "file".</p>
format	Data Type	<p>The <code>format</code> property is represented by the Data Type field of a REST Parameter.</p> <p>Currently, CentraSite does not support mapping parameters of the data type "file".</p>
items	Possible Values	<p>The <code>enum</code> property is represented by the Possible Values field of a REST Parameter.</p> <p>Limitation: Currently, CentraSite supports the Possible Values field only for a "string" data type.</p>
collectionFormat		Currently, CentraSite does not support the <code>collectionFormat</code> attribute.

Swagger Attribute	REST API Field	Notes
default	Default Value	The default property is represented by the Default Value field of a REST Parameter.
maximum		Currently, CentraSite does not support the maximum attribute.
exclusivemaximum		Currently, CentraSite does not support the maximum attribute.
minimum		Currently, CentraSite does not support the minimum attribute.
exclusiveminimum		Currently, CentraSite does not support the minimum attribute.
minLength		Currently, CentraSite does not support the minLength attribute.
maxLength		Currently, CentraSite does not support the maxLength attribute.
enum	Possible Values	The enum property is represented by the Possible Values field of a REST Parameter.
pattern		Currently, CentraSite does not support the pattern attribute.
maxItems		Currently, CentraSite does not support the maxItems attribute.
minItems		Currently, CentraSite does not support the minItems attribute.
uniqueItems		Currently, CentraSite does not support the uniqueItems attribute.
multipleOf		Currently, CentraSite does not support the multipleOf attribute.

Swagger Resource Fields

Swagger resource properties (data) are mapped to the REST API resource fields. The following table shows the resource mappings between a Swagger definition and a REST API:

Swagger Resource Field	REST API Field	Notes
\$ref		Currently, CentraSite does not support the \$ref property. However, the \$ref property is represented by mapping the contained REST Resources and attaching the referenced files to the Specification profile in the API details page.
get, put, post, delete, options, head, patch	REST Method	The get property is represented as the REST Method object.
parameters	REST Parameter	The parameter property is represented as a REST Parameter object.

Swagger Method Fields

Swagger method properties (data) are mapped to the REST API method fields. The following table shows the resource mappings between a Swagger definition and a REST API:

Swagger Method Field	REST API Field	Notes
tags		Currently, CentraSite does not support the tags property.
summary	Name	The summary property is represented by the Name field of a REST Method.
description	Description	The description property is represented by the Description field of a REST Method.
externalDocs		The externalDocs are not explicitly mapped to the REST API; instead, the external documentation is attached as a file attribute to the Specification profile.
operationId		Currently, CentraSite does not support the operationId property.
security		Currently, CentraSite does not support mapping the security at the API level. However, it is represented by the Supported Access Token Types field in the API Portal Information profile.

Swagger Method Field	REST API Field	Notes
parameters	Parameter	The parameter property is represented by the Parameter field of a REST Method.
responses	Status Code	The responses property is represented by the Status Code field of a REST Method.
produces	Response Content-Type	The produces property is represented by the Response Content-Type field of a REST Method.
consumes	Request Content-Type	The consumes property is represented by the Request Content-Type field of a REST Method.
deprecated	Deprecated	The deprecated property is propagated to the taxonomy structure named Deprecated and represented by the API Maturity Status property defined in the API Portal Information profile.

Swagger Status Code Fields

Swagger status code properties (data) are mapped to the REST API status code fields. The following table shows the resource mappings between a Swagger definition and a REST API:

Swagger Method Field	REST API Field	Notes
code	Status Code	The HTTP status code property is represented by the Status Code field of a REST Method.
description	Description	The description property is represented by the Description field of a REST Method level.
schema		Currently, CentraSite does not support the schema property.
headers		Currently, CentraSite does not support the headers property.
examples		Currently, CentraSite does not support the examples property.

Swagger Security Scheme Fields

Swagger security schemes are not explicitly mapped to the REST API. Instead, the Swagger security schemes are propagated to the taxonomy structure named **Security Types**, and represented by the **Supported Access Token Types** property at the **API Portal Information** profile.

Security Type	Description
API Key	The API's authentication requires using an API key.
Basic Authentication	The API's authentication requires using Basic Access Authentication as described in RFC2617.
Digest Authentication	The API's authentication requires using Digest Access Authentication as described in RFC2617.
OAuth 1.0	The API's authentication requires using OAuth 1.0 as described in RFC5849.
OAuth 2.0	The API's authentication requires using OAuth 2.0 as described in RFC6749.
x-{other}	The API's authentication requires using another authentication method.

The supported security schemes along with their scheme reference name are mapped in the CentraSite registry. For more information about the security scheme, see the Swagger specification.

Asset Navigator

CentraSite offers the possibility of viewing associations between the registry objects and hence identifying the impact when updating or deleting an asset in the registry.

You can visualize the currently defined associations for an asset with other registry objects in CentraSite using the:

- *Asset Navigator* feature in CentraSite Business UI
- *Impact Analysis* feature in CentraSite Control

Asset Navigator in CentraSite Business UI

This section provides information on the Asset Navigator feature in CentraSite Business UI.

Introduction to Asset Navigator

The Asset Navigator provides a graphical representation of relationships between assets based on most common use cases. The graphical asset navigator in CentraSite Business UI helps you to easily navigate and visualize the dependencies between assets for various use cases. Several use cases for navigation are provided out of the box to allow for analysis of the runtime deployment landscape, organization structure or dependencies to or from particular assets. The asset navigator can be extended through saved searches. This feature helps you to:

- Understand dependencies between assets
- Visualize your runtime landscape and deployment information
- Understand your versioning and consumption history of services

- Get an organization chart of your organization structure in CentraSite

You can view the information based on the use case requirements. For example, you can visualize the information based on the Global Usecases or the Asset Specific Usecases.

- Global Usecases include use cases such as Organization Structure and Runtime Landscape view.
- Asset Specific Usecases include use cases such as Asset Dependency, Asset Usage, and so on.

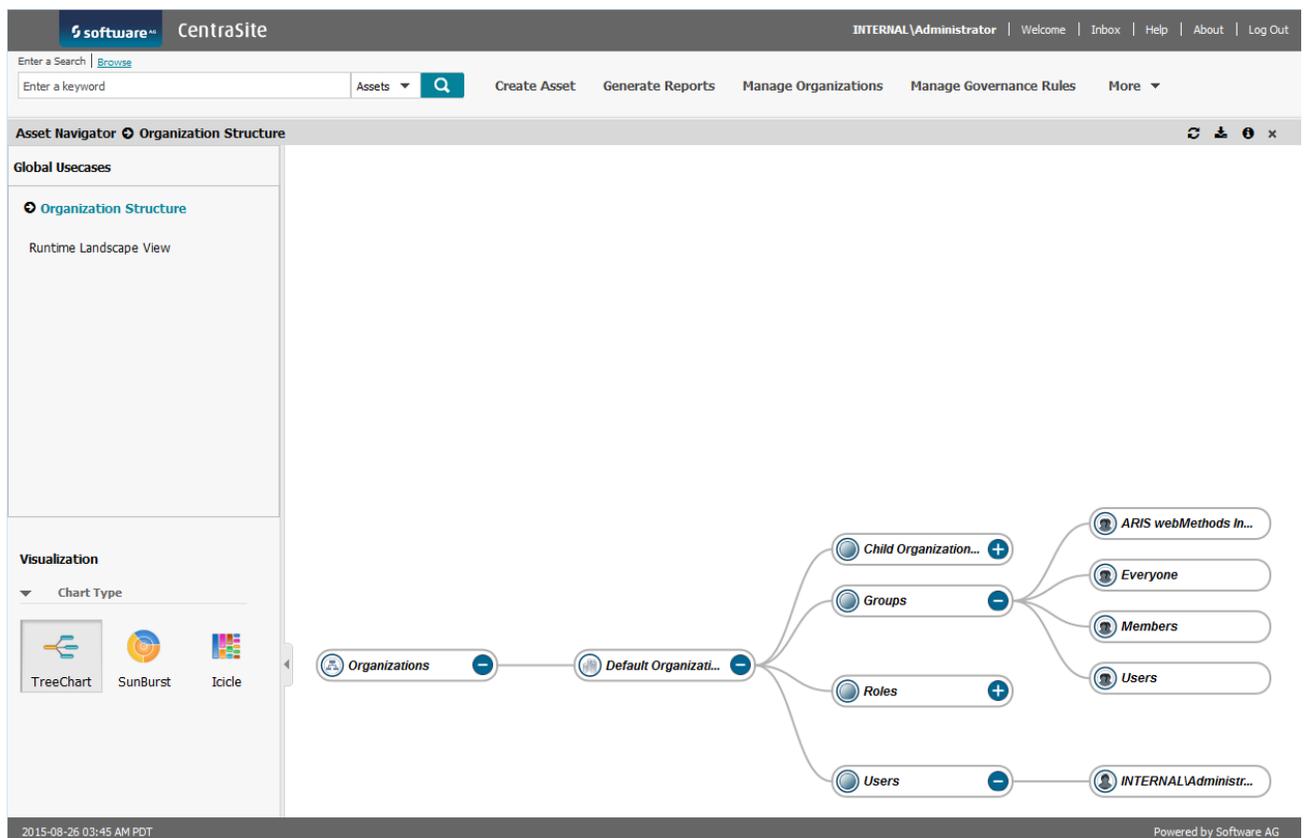
Graphical Visualization

You can visualize the asset in graphical form in one of the following ways:

- On the activity bar, point to **More**, and then click **Asset Navigator**.

You can view the following sections in the left pane.

- Global Usecases
- Visualization

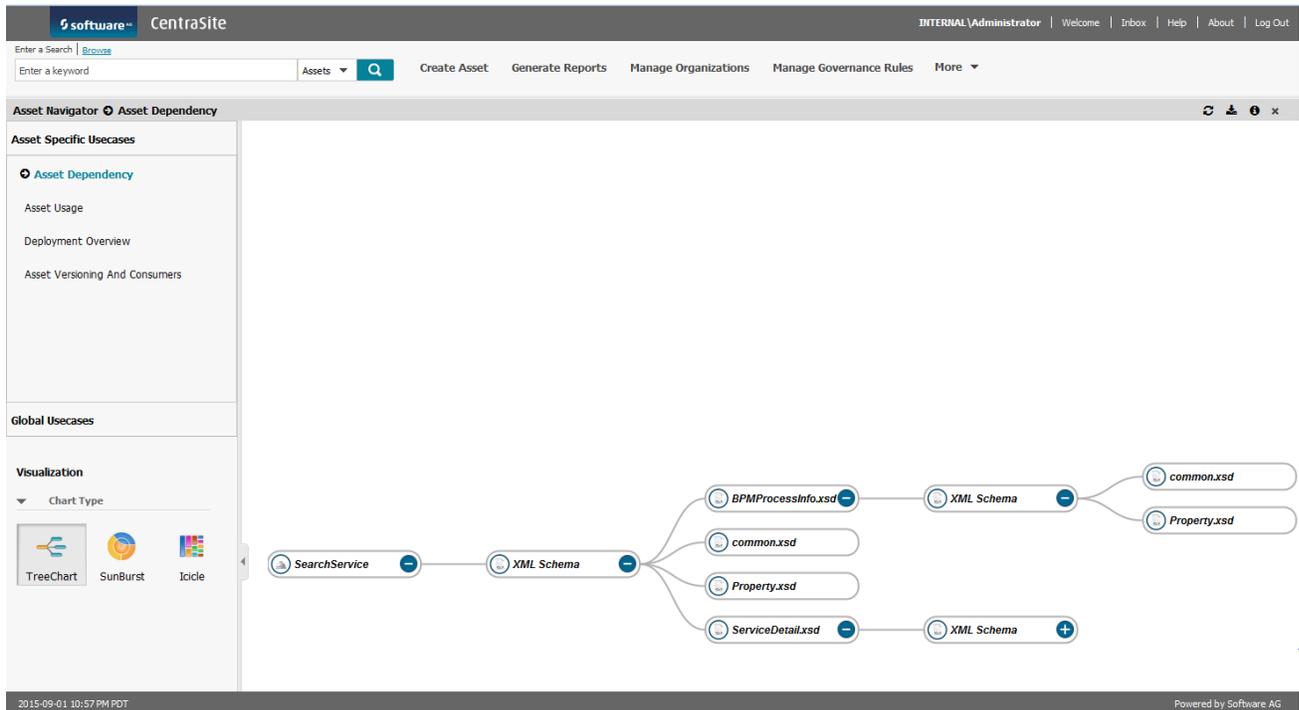


- On the actions bar of the Asset Details page, click **Asset Navigator**.

You can view the following sections in the left pane.

- Asset Specific Usecases

- Global Usecases
- Visualization



Navigating to Asset Navigator Through the Activity Bar

On the activity bar, point to **More**, and then click **Asset Navigator**. This displays the Global Usecases section.

The global use cases that CentraSite provides out-of-the-box are:

- **Organization Structure:** This provides a graphical view of the available organizations, and the Roles, Groups, Users, and Child Organizations within each organization.
- **Runtime Landscape View:** This provides a graphical view of all the gateways that are currently integrated in CentraSite. It also provides a view of the Services published in each gateway.

Navigating to Asset Navigator Through the Actions Bar

On the actions bar of the Asset Details page, click **Asset Navigator**. This displays the Asset Specific Usecases and Global Usecases sections.

The asset specific use cases that CentraSite provides out-of-the-box are:

- **Asset Dependency:** This provides a graphical view of all the outbound references from an asset, for example, assets that are dependent on the current asset such as, schema objects, other services, custom asset types, and so on. Asset Dependency also provides a graphical view of all the use cases of a given relationship type for the given asset type.

- **Asset Usage:** This provides a graphical view of all the inbound references to an asset such as, other services, processes, and so on.
- **Deployment Overview:** This provides a graphical view of the list of API Portals and Mediators which the given version of the asset is published to.
- **Asset Versioning And Consumers:** This provides a graphical view the different versions of the service. It also provides a graphical view of the consumers for the given version of the service.

You can click on each node to go to the asset details page of the displayed asset. The last viewed chart type is retained for the last viewed use case even if you switch between the pages.

Types of Charts

Asset Navigator supports the following visualization formats:

-  - TreeChart
-  - SunBurst
-  - Icicle

The following information is displayed when you hover over a node in any of the above mentioned graphs. These attributes can also be configured in the **centrasite.xml** file:

- **Name:** The name of the asset.
- **Description:** The description of the asset.
- **Type:** The type of the asset, for example, Service, Virtual Service, and so on.
- **Created Date:** The date when the asset was created.
- **Last updated:** The date when the asset was last updated.
- **Owner:** The owner of the asset.
- **Organization:** The name of the organization to which the asset belongs.
- **Version:** The version of the asset.

You can click on each node to go to the asset details page of the selected asset.

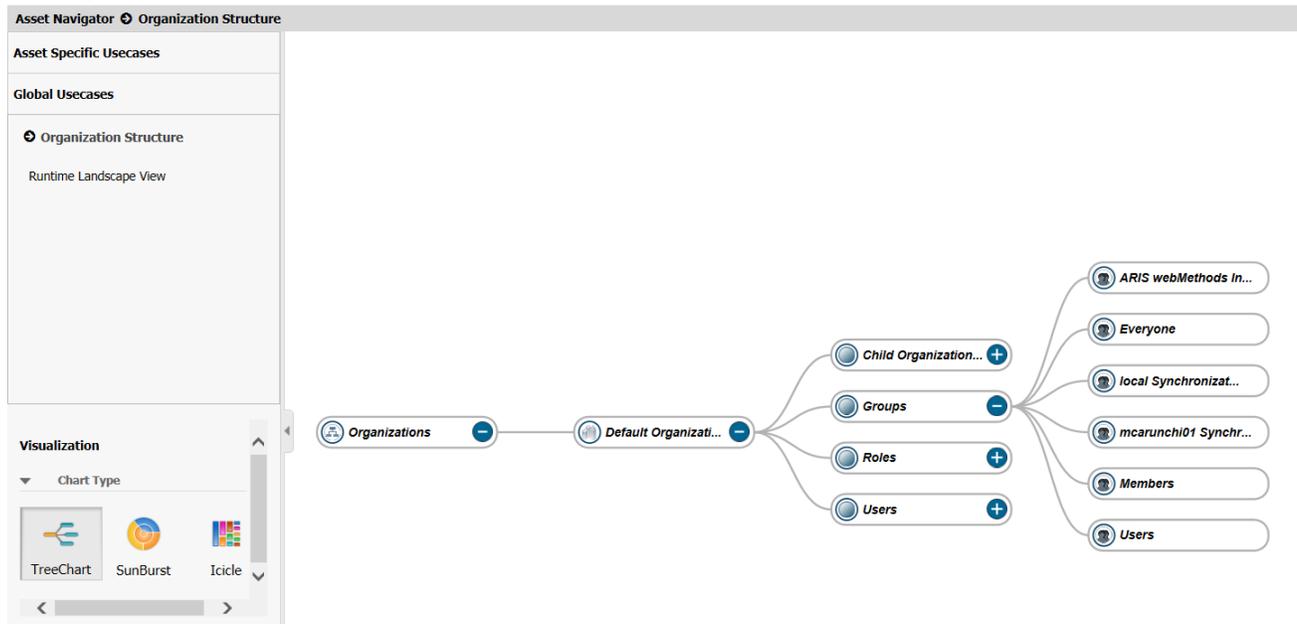
By default, all use cases are configured with the TreeChart type. However, you can configure the use cases and default chart type for use cases in the **centrasite.xml** configuration file.

For more information on configuring use cases and charts, see [“Configuration Settings” on page 654](#).

TreeChart Chart

TreeChart is used to represent the hierarchical data in a graphical form. The tree elements are called nodes. The root is the starting node. A TreeChart view consists of a root node and one or more child nodes. Each child node consists of a parent node, except for the root node. Click  to

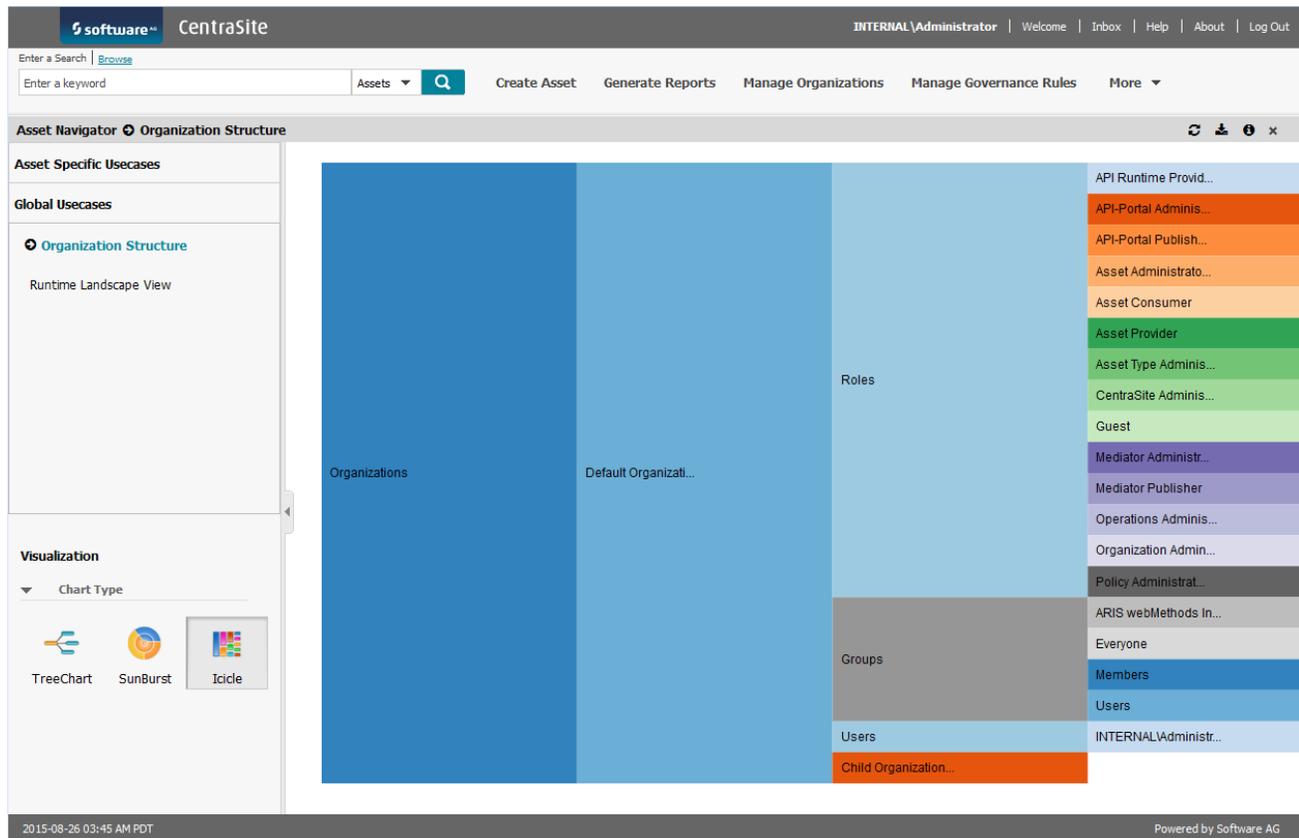
expand the tree and click  to collapse the tree. Here is an example of the TreeChart diagram for the organizations.



SunBurst Chart

Sunburst visualization is a radial space-filling visualization technique for displaying tree like structures. The Sunburst chart displays a hierarchy of series of rings. Each ring represents the child of the ring it encloses and the area of each slice corresponds to its value. The outer ring can have a colour gradient with respect to the parent ring.

You can visualize an asset in detail by clicking on the corresponding asset. To visualize the complete relationship between the assets, click on the centre of the chart. Here is an example of the SunBurst diagram for the organizations.



You can click anywhere to zoom in, or click on the top bar to zoom out.

Configuration Settings

The visualization types, use cases, and information for the visualization types can be configured in the CentraSite's customization file, **centrasite.xml**.

You can find this file on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\custom\conf`. However, CentraSite provides the following built-in configurations available in the `<AssetNavigatorConfigurations>` tag in the **centrasite.xml** file. You can modify the configurations to suit your requirements.

1. **Use Cases:** You can configure the different use cases to view the assets and also specify the default chart for that object. For example,

```
<UseCases>
  <UseCase id="OrganizationStructure" rootQueryId="getRootForOrgStructure"
    fillChild="false" default = "true" defaultChart="SunBurst"
    applicableChartTypes="SunBurst,TreeChart,Icicle">
    INMCL_USECASE_ORG</UseCase>
  <UseCase id="RuntimeLandscapeView"
    rootQueryId="getRootForRuntimeLandscapeview"
    fillChild="false" defaultChart="TreeChart"
    applicableChartTypes="SunBurst,TreeChart,Icicle">
    INMCL_USECASE_RUNTIMELANDSCAPEVIEW</UseCase>
</UseCases>
```

The following attributes can be configured for each use case:

Parameter	Description
id	Unique identifier for the use case.
rootQueryId	ID for the first saved search to be executed.
fillChild	Boolean to determine whether to display the complete information at the beginning of the tree chart.
default	Use cases to be displayed by default for the first time.
defaultChart	Chart to be displayed by default for the first time.
applicableChartTypes	The chart types that you want to be displayed for a specific use case.

2. **Asset Specific Use cases:** You can configure the list of use cases displayed under the Asset Specific Use cases. For example,

```
<AssetSpecificUseCases>
  <UseCase id="AssetDependency"
    rootQueryId="getRootForAssetDependencies"
    default="true" defaultChart="TreeChart"
    applicableChartTypes="SunBurst,TreeChart,Icicle">
    INMCL_USECASE_ASSETDEPENDENCY</UseCase>
  <UseCase id="AssetUsage"
    rootQueryId="getRootForAssetUsage"
    defaultChart="TreeChart"
    applicableChartTypes="SunBurst,TreeChart,Icicle">
    INMCL_USECASE_ASSETUSAGE</UseCase>
  <UseCase id="DeploymentOverview"
    rootQueryId="getRootForDeploymentOverview"
    defaultChart="TreeChart"
    applicableChartTypes="SunBurst,TreeChart">
    INMCL_USECASE_DEPLOYMENT</UseCase>
</AssetSpecificUseCases>
```

The following attributes can be configured for each use case:

Parameter	Description
id	Unique identifier for the use case.
rootQueryId	ID for the first saved search to be executed.
fillChild	Boolean to determine whether to display the complete information at the beginning of the tree chart.
default	Use cases to be displayed by default for the first time.

Parameter	Description
defaultChart	Chart to be displayed by default for the first time.
applicableChartTypes	The chart types that you want to be displayed for a specific use case.

3. **Chart Types:** You can configure the type of charts and their corresponding images. For example,

```
<ChartTypes>
  <Chart id="SunBurst" nameTruncationLimit="12"
    imageUrl="images/system/charticons/sunburst.png" />
  <Chart id="TreeChart" nameTruncationLimit="18"
    imageUrl="images/system/charticons/treechart.png" />
  <Chart id="Icicle" nameTruncationLimit="18"
    imageUrl="images/system/charticons/icicle.gif" />
</ChartTypes>
```

The following attributes can be configured for each use case:

Parameter	Description
id	Unique identifier for the chart.
nameTruncationLimit	Truncation limit for the displaying text.
imageUrl	Display icons for the charts are configured here.

4. **Images:** You can configure the icons used for the nodes in the tree chart. For example,

```
<Images>
  <Image ref="getRootForOrgStructure">d3/images/org.png</Image>
  <Image ref="getOrgNames">d3/images/org-child-icon.png</Image>
  <Image ref="rolesDummy">d3/images/medium dummy.png</Image>
  <Image ref="groupsDummy">d3/images/medium dummy.png</Image>
</Images>
```

Note:

The path mentioned in this configuration is related to the `images` folder. You can find this folder on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\d3`.

The following attributes can be configured for each use case:

Parameter	Description
ref	Id of the savedQuery.

5. **Common Attributes:** You can configure the list of common attributes (displayed when you hover over an asset or node) for each asset or node. For example, name, description, type, and so on.

```
<CommonAttributes>
  <Attribute qname=
    "{http://namespaces.CentraSite.com/Schema/jaxr}name"
```

```

description=
  "INMCL_COMMON_ATTR_NAME_DESC">INMCL_COMMON_ATTR_NAME</Attribute>
<Attribute qname=
  "{http://namespaces.CentraSite.com/Schema/jaxr}description"
description=
  "INMCL_COMMON_ATTR_DESC_DESC">INMCL_COMMON_ATTR_DESC</Attribute>
<Attribute qname=
  "{http://namespaces.CentraSite.com/Schema/jaxr}type"
description=
  "INMCL_COMMON_ATTR_TYPE_DESC">INMCL_COMMON_ATTR_TYPE</Attribute>
</CommonAttributes>

```

The following attributes can be configured for each use case:

Parameter	Description
qname	The qualified name.
description	The description of the common attribute.

6. **Saved query:** To retrieve data with respect to the asset and node from the database. For example,

```

<SavedQueries>
  <SavedQuery id="getRootForAssetDependencies"
    childQueries="getTypeNamesForAssetDependencies"
    ref="AssetDependency" >${AssetName}</SavedQuery>
  <SavedQuery id="getTypeNamesForAssetDependencies"
    resource="GetOutgoingTypesAssociatedWithAsset"
    parameters="${assetKey}" ref="AssetDependency"
    childQueries="getAssetsForOutgoingAssociation"></SavedQuery>
  <SavedQuery id="getAssetsForOutgoingAssociation"
    resource="GetAssetsForOutgoingAssociation"
    parameters="${typeId},${assetKey}"
    childQueries="getTypeNamesForAssetDependencies" ref="AssetDependency"/>
</SavedQueries>

```

The following attributes can be configured for each use case:

Parameter	Description
id	Unique identifier for the saved query.
resource	Name of the saved search file.
childQueries	Set of queries to be executed next to the current saved query.
ref	Use case ID for which the data is provided.
parameters	Input parameters required by the saved search.

Adding Custom Use Cases

You can add additional use cases other than the use cases preloaded with CentraSite. To add a custom use case in the asset navigator, see the **read-me** file in the Asset Navigator folder created

for demonstration. You can find the **read-me** file on <CentraSiteInstall_Directory>\demos\AssetNavigator\Asset Navigator.

Asset Navigator in CentraSite Control

This section provides information on the Asset Navigator feature, termed as Impact Analysis feature in CentraSite Control.

Introduction to Impact Analysis

Note:

The *Impact Analysis* feature of CentraSite Control that is used to easily navigate and visualize the associations between the catalog assets and registry objects, and hence identify the impact when updating or deleting assets in the catalog has been deprecated and will be removed in a future release. Instead, you can use the Asset Navigator interface of CentraSite Business UI helps you to easily navigate and visualize the dependencies between assets for various use cases.

The Impact Analysis feature of CentraSite Control offers the possibility of viewing associations between the registry objects and hence identifying the impact when updating or deleting an asset in the CentraSite Control.

This feature enables you to easily navigate and visualize the associations between the catalog assets and registry objects. This feature helps you to:

- Understand asset-to-objects associations by displaying the associations that exist between the catalog assets and other registry objects.
- Check that existing associations between the assets and objects are not violated when you make changes in the registry. Also, check the external links from registry objects to supporting documents.
- Determine the impact that updating or deleting an asset would have on its related objects.

You can visualize the currently defined associations for an asset with other registry objects, either in graphical or tabular form. The graphical representation is enabled through a Flash-based web browser.

Note that apart from the assets, you can also view the impact analysis for other CentraSite objects like organizations, users, groups, and roles through the object's context menu.

Graphical Visualization of Impact Analysis

To visualize the impact of an asset in graphical form, follow these steps:

1. In CentraSite Control, go to **Asset Catalog > Browse**.

A list of currently defined assets is displayed in the **Assets** pane.

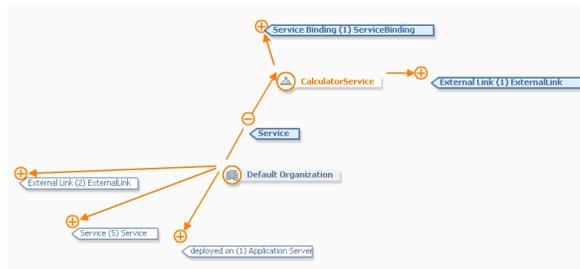
- In the **Assets** pane, right-click an asset for which you want to display the impact analysis, and click **Impact Analysis**, or select the check boxes for multiple assets, click the **Actions** menu, and click **Impact Analysis**.

You can also initiate this step using the **Actions** menu in the details page of an asset.

In the graphical view, there are four tabs (**Types**, **Associations**, **Controls**, and **Configuration**) to allow you to set the configuration parameters that control how the impact analysis is displayed.

The graphical view shows a visual representation of the selected asset, the objects referred to by the selected asset, the objects that refer to the selected asset and the associations.

Here is an example of the impact analysis diagram for the asset *CalculatorService*, which is an asset of type Service:



The asset for which the impact analysis is being displayed is shown in a box with orange colored text (in the example, **CalculatorService**). The objects that are associated with the central asset are displayed initially in boxes with dark blue text on a lighter background (for example, the node **Default Organization**).

Associations between assets are represented by orange-colored arrows. Each association has a name and a direction (indicated by the arrowhead). For example, the diagram shows the association with the name **Service** that connects the **Default Organization** node to the **CalculatorService** node. This indicates that an association of type Service connects the two nodes. The arrowhead points to the **CalculatorService** node, indicating that **Default Organization** contains a service **CalculatorService**.

The display of the associations can be expanded or collapsed as required. If an association is shown with an orange plus sign, you can click on the plus sign to expand the association; this reveals the node or nodes at the other end of the association, and the plus sign changes to a minus sign. To collapse the association, that is, to hide the other end of the association, click on the minus sign, and the display reverts to its original state.

When you expand an association, the association's background color changes to blue. If you collapse a previously expanded association, its color remains blue; this way you can identify the associations that you have already visited. Associations that have not yet been expanded are displayed with a neutral background (that is, the same background color as the drawing canvas).

The text in the box for any collapsed association shows three items of information:

- The type of the association;

- The number of currently invisible target nodes that are attached to the visible source node;
- The object type of the invisible target node(s).

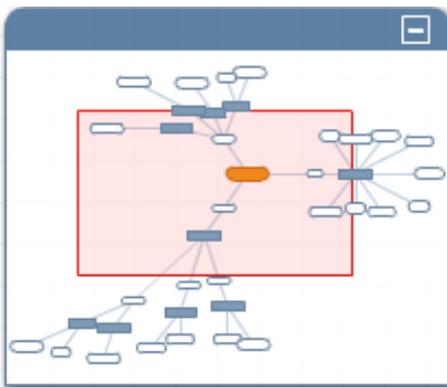
So, for example, the association labeled **deployed on (1) Application Server** in the diagram indicates an association of type **deployed on** between the visible source node **Default Organization** and a currently invisible node of type **Application Server**.

If you click on an object (as opposed to an association), a window appears with a short summary of the object's definition.

You can move the whole diagram within the web browser display by moving the cursor to an empty part of the diagram and dragging the diagram in the required direction.

You can rearrange the position of any node in the diagram by clicking on the node and dragging it to a new location on the canvas.

The display also contains a bird's eye view of the impact analysis diagram, for example:



The shaded central part is the part that is shown in detail in the full display. You can drag the shaded central part to any location in the bird's eye view, and the focus of the full display will move accordingly. You can minimize the bird's eye view by selecting the - icon. The minimized view shows just a menu bar with a + icon. To restore the view, click the + icon.

Tabular Visualization of Impact Analysis

To visualize the impact of an asset in tabular form, follow these steps:

1. In CentraSite Control, go to **Asset Catalog > Browse**.

A list of currently defined assets is displayed in the **Assets** pane.

2. In the **Assets** pane, right-click an asset for which you want to display the impact analysis, and click **Impact Analysis**

You can also select multiple assets, click the **Actions** menu, and click **Impact Analysis**.

Note:

You can also initiate this step using the **Actions** menu in the details page of an asset.

3. In the graphical view, click **Switch to Tabular View**.

The objects and associations listed in the table depend on your current filter configuration settings. You can change these settings by choosing the **Customize** button. This opens the **Customize** dialog, in which you can select the required object types to display, the required association types and the nesting depth of the associations.

Note:

Customization that you make for the tabular view apply only for this view, not for the graphical view.

To return to the graphical view, click **Switch to Graphical View**.

Configuration Settings

You can use a filter to restrict the type of objects and associations displayed, and to specify the maximum depth of nesting for the displayed associations.

Note:

Customization that you make for the graphical view apply only for the graphical view, not for the tabular view.

Built-In Filter Configurations

CentraSite provides the following built-in filter configurations that you can use to visualize the impact analysis for the various objects:

- **Asset Dependencies:** This shows associations that are of particular relevance for assets.
- **Schema Usage:** This shows associations that are of particular relevance for schema objects.
- **Organization Details:** This shows associations that are of particular relevance for organization objects.
- **Service Details:** This shows associations that are of particular relevance for service assets.
- **webMethods Assets:** This shows associations between services and webMethods Suite types like CAF and web applications.

By default, CentraSite displays the *Asset Dependencies* filter configuration. You can select the configuration you require by choosing its radio button from the **Configuration** tab. After a few moments, the display is updated according to this configuration.

If necessary, you can select the filter configuration that you want to customize, and change the filter settings accordingly. If you change any of the filter configurations and click the **Refresh Canvas** icon in the appropriate configuration menu, the display will be updated using your new settings. Any such changes you make apply also in subsequent login sessions. If you want to return to the original settings, delete the filter configuration, this deletes the current settings and restores the configuration to its original state.

Adding a Custom Filter Configuration

You may want to have various display scenarios for your graphical impact analysis. For example, in one case you might want to restrict the depth of the association tree to two levels, but in another case you might want to have no such restriction. You can save these configuration scenarios for recall at a later stage by opening the **Configuration** tab and typing a new name in the appropriate text field. This lets you save your current configuration scenario under a name of your choice. If you have defined several scenarios, you can select the one you require by choosing it in the **Configuration** tab.

In the graphical view, there are four tabs (**Types**, **Associations**, **Controls**, and **Configuration**) to allow you to set the configuration parameters that control how the impact analysis is displayed.

> To define custom filter settings

1. In CentraSite Control, go to **Asset Catalog > Browse**.

A list of currently defined assets is displayed in the **Assets** pane.

2. In the **Assets** pane, right-click an asset for which you want to display the impact analysis, and click **Impact Analysis**.

You can also select multiple assets, click the **Actions** menu, and click **Impact Analysis**.

Note:

You can also initiate this step using the **Actions** menu in the details page of an asset.

3. To display or hide specific object types, make appropriate selections in the **Types** tab.

You can select a predefined entry such as **Types > Assets > Assets** to restrict the display to show the assets rather than all objects.

To further refine the filter settings, click the **Custom** entry for the selected type. For example, for the type **Asset**, select **Types > Assets > Custom**. In the ensuing menu, select the asset types you want to display.

4. To display or hide specific association types, make appropriate selections in the **Associations** tab.

You can select a predefined entry such as **Associations > Relationship Attributes > Relationship Attributes** to restrict the display to show the relationship associations rather than all associations.

To further refine the filter settings, click the **Custom** entry for the selected type. For example, for the type **Relationship Attributes**, select **Associations > Relationship Attributes > Custom**. In the ensuing menu, select the association types you want to display.

5. To control the nesting depth of the displayed associations, make appropriate selections in the **Controls** tab.

For a nesting depth of 1, 2, 3, 4, or 5, select the appropriate numbered symbol. For a level deeper than 5, type the required value in the **Custom** field.

In certain cases, when the text area of the label exceeds the width of the text box, the excess characters are simply not displayed. In such cases, for labels that you know will contain a certain amount of text, then it is recommended that you use the **Label Width** entry and specify the desired width of the label (although it may take some trial-and-error to get the result you want).

6. Expand the **Configuration** tab. Type a new name in the text field (located in the end of the panel), and click **Save**.

Deleting a Custom Filter Configuration

Note:

You cannot permanently delete any of the predefined filter configurations. Deleting a predefined filter configuration just deletes the current settings and restores the configuration to its original settings.

> To delete custom filter settings

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the **Assets** pane, right-click an asset for which you want to display the impact analysis, and click **Impact Analysis**.

You can also select multiple assets, click the **Actions** menu, and click **Impact Analysis**.

Note:

You can also initiate this step using the **Actions** menu in the details page of an asset.

3. Click the **Configuration** tab.
A list of custom filters that you created is displayed.
4. Locate the custom filter you want to delete, and click the **Delete** symbol.
5. Click **OK** in the confirmation dialog displayed.

Zooming the Graphical Visualization

You can zoom the display between 40% and 100% of the default display size.

> To zoom the graphical display

1. In CentraSite Control, go to **Asset Catalog > Browse**.

2. In the **Assets** pane, right-click an asset for which you want to display the impact analysis, and click **Impact Analysis**.

You can also select multiple assets, click the **Actions** menu, and click **Impact Analysis**.

Note:

You can also initiate this step using the **Actions** menu in the details page of an asset.

3. To zoom in or zoom out the graphical display, click the plus sign or the minus sign in the panel labeled - 100% +.
 - Type over the 100% with the value you require, and click the mouse on a neutral part of the display.
 - Rotate the mouse wheel to change the zoom factor directly. (Not supported on all browsers).

Printing the Graphical Visualization

You can print the graphical visualization of impact analysis for an asset.

> To print the graphical visualization

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the **Assets** pane, right-click an asset for which you want to display the impact analysis, and click **Impact Analysis**.

You can also select multiple assets, click the **Actions** menu, and click **Impact Analysis**.

Note:

You can also initiate this step using the **Actions** menu in the details page of an asset.

3. Click the **Controls** tab.
4. In the area labeled **Print**, click one of the following options:
 - **Print the graphical impact analysis using the current zoom setting:** The output is printed using the current zoom settings. As a result, multiple pages is printed if the diagram is too large to fit onto one printed page.
 - **Print the graphical impact analysis on one page:** The output is scaled to fit onto one printer page.

Full-Screen Display of Graphical Visualization

You can increase the physical display area of the graphical impact analysis on your monitor by activating full-screen mode. In this mode, all menus and task bars belonging to your web browser and operating system are suppressed.

➤ **To activate the full-screen mode of graphical display**

1. In CentraSite Control, go to **Asset Catalog > Browse**.

A list of currently defined assets is displayed in the **Assets** pane.

2. In the **Assets** pane, right-click an asset for which you want to display the impact analysis, and click **Impact Analysis**, or select the check boxes for multiple assets, click the **Actions** menu, and click **Impact Analysis**.

The default is **Graphical** view.

Note:

You can also initiate this step using the **Actions** menu in the details page of an asset.

3. Click the **Controls** tab.
4. In the area labeled **Full Screen**, click the **Toggle full screen** symbol to switch the display to full-screen mode.

To exit from the displayed full-screen mode, press *Esc*.

10 Policy Management

■ Introduction to Design and Change-Time Policies	668
■ Managing Design Time Policies through CentraSite Business UI	681
■ Managing Design and Change-Time Policies through CentraSite Control	711
■ Managing Design-Time and Change-Time Policies through Command Line Interface .	758
■ Predefined Policies	776
■ Built-In Design/Change-Time Actions Reference	782
■ Configuring Email Notifications	842

Introduction to Design and Change-Time Policies

Design and change-time policies provide governance controls that you can use to effectively administer and manage web services and other assets within your SOA environment. Design-time policies enable you to control the acceptance of assets into the registry and manage their deployment into the runtime environment. You can use Design/Change-Time policies to ensure that assets entering the SOA environment conform to organizational standards and conventions, meet the architectural requirements of your enterprise, and adhere to industry best practices. You can also use policies to execute standard procedures, such as, initiating review processes, issuing notifications, and granting instance-level permissions at key points in an asset's lifecycle.

Design/change-time policies specify a set of *actions* that are to be executed when a specified *event* occurs to an instance of a specified *object type*. You use design/change-time policies to customize the behavior of CentraSite when certain events occur on assets and other objects in the registry (for example, during creation, modification, and/or deletion events). You can use design/change-time policies to perform tasks such as obtaining approvals, executing automated tests, issuing notifications, and imposing organizational standards when assets or other objects are created, modified, or deleted.

For example, a policy could instruct CentraSite to perform any of the following tasks:

- Verify that a schema name conforms to specified conventions when a new schema is added to the catalog.
- Submit a Web service to a review panel for approval before the service enters the Design state.
- Send an email notification to the IT organization when a service is deployed.
- Change the permission settings on an asset when the asset switches to the Available state.
- Assign a user in a given role to a particular group when a new user is added to CentraSite.
- Alter an asset's lifecycle path depending on the way in which the asset is classified.

Note:

The use of design/change-time policies is not supported if you are using a CentraSite Community Edition license.

A design/change-time policy has two major elements:

- *action list*: specifies the tasks (policy actions) that are to be executed.
- *scope*: specifies when the action list is to be executed.

Design and Change-Time Policy Actions

A *policy action* is a programmed task written in Java (a scripting language). A policy action can perform any type of work you require for example, sending an email, validating an attribute setting, submitting an approval request, updating a database. It can have one or more input parameters. An action that sends an email message includes input parameters that specify the text

of the message and to whom it is to be sent. A policy action returns a completion code that indicates whether it completed its task successfully.

The action list in a policy can contain one or more actions. CentraSite executes the actions in the order that they appear in the list. If an action does not complete successfully, CentraSite immediately exits the policy and skips any remaining actions in the list. Policy failures are recorded in CentraSite's policy log.

CentraSite is installed with a library of built-in policy actions that you can use to construct policies. You can also develop your own custom actions using Java

Design and Change-Time Policy Scope

The *policy scope* specifies the conditions under which CentraSite is to execute the policy. It consists of two main parameters: *Event Type* and *Object Type*.

- *The Event Type parameter:* specifies the events to which the policy applies. An event represents a specific point during a registry operation when policies can be executed. Such points include the PreCreate event (the point in time just before CentraSite saves a new instance of an object to the registry), the PostCreate event (the point immediately after CentraSite saves a new instance of an object to the registry) and the PreDelete event (the point in time immediately before CentraSite deletes an object). Other events include the points in time before and after an update operation and before and after a state change.
- *The Object Type parameter:* specifies the types of objects to which the policy applies. Policies can be applied to any type of asset and to several other types of registry objects.

Together, the event type and the object type determine when the policy executes. You can make the scope as narrow or as broad as you need. That is, you can target the policy for one particular type of event and object (for example, a PreDelete event on an XML Schema) or apply the policy to multiple events and objects (for example, a PreCreate, PreUpdate, PreDelete event on a Service, an XML Schema, a BPEL Process).

Refining a Policy's Scope with Additional Selection Criteria

You can optionally refine the scope of a policy to narrow the set of objects to which the policy applies. To do this, you include additional selection criteria based on an object's Name, Description, or Classification properties. For example, you might create a policy that applies only to Application Servers whose name includes the string myDomain.com or to Application Servers that are classified as Software AG Runtime.

The ability to execute policies based on object classification is an especially effective way to selectively apply policies to objects.

Scope of a Policy Action

A policy action also has a declared scope. The action's scope specifies the object and event types with which the action can be used. Some actions have very specific object-type and event-type requirements. For example, you can use the Validate Policy Deactivation action only during a PreStateChange on a policy object. Other actions support a broad range of object and event types.

A policy can contain only actions that support the full set of object types and event types specified by the policy's scope. For example, if you create a policy that executes on the PreCreate and PreUpdate events for XML Schemas, it can only contain actions whose scope includes the PreCreate and PreUpdate event types and the XML Schema object type.

When you create a policy, CentraSite's user interface only allows you to select actions that satisfy the specified scope of the policy. If you subsequently change the policy's scope, CentraSite does not allow you to save the updated policy unless all of its actions support the policy's new scope.

Objects on Which Design/Change-Time Policies Can Operate

Design/change-time policies operate on the objects that CentraSite manages. The types of objects to which you can apply design/change-time policies are:

- Organizations
- Users (Note that policies you apply to User objects are enforced when events occur to the User objects in the CentraSite registry, not when events occur in the external naming directory.)
- Taxonomies
- Policies
- Assets. (You can create policies that apply generally to all policy-enabled assets or to specific policy-enabled types.)
- Report Templates
- Lifecycle Models

Applying Policies to Assets

The type definition for an asset includes a property setting called **Policies Can Be Applied**. This property determines whether assets of the given type are *policy-enabled* (that is, whether design/change-time policies are to be executed against them). When this property is enabled, you can create policies that are specific to the assets of that type. Additionally, instances of that type are capable of triggering design/change-time policies that target "assets" in general.

By default, the **Policies Can Be Applied** property is enabled for an asset type. If you do not want instances of particular asset type to be affected by design/change-time policies, disable this property in the asset's type definition.

Applying Policies to Virtual Types

When an asset is an instance of a virtual type, the set of policies that CentraSite applies to the asset depends on the virtual type's **Inherit Base Type Policies** setting. If the type's **Inherit Base Type Policies** option is enabled, CentraSite applies the policies of the base type to the asset *in addition to* the policies of the virtual type. For example, the **Inherit Base Type Policies** option is, by default, enabled for virtual services. Therefore, when CentraSite enforces policies for a virtual service, it applies the set of policies that are defined for the Virtual Service object type *and* the set of policies that are defined for the Service object type (the base type for the Virtual Service type).

If you disable the **Inherit Base Type Policies** option for a virtual type, CentraSite applies to the asset only the policies that are defined for the virtual type. For example, if you disable the **Inherit Base Type Policies** option for the Virtual Service object type, CentraSite applies to virtual services, only the policies that are defined for the Virtual Service type. Policies that are defined for the Service type are not applied.

The following table summarizes how the set of policies that CentraSite enforces for a virtual type is affected by the state of the **Inherit Base Type Policies** option:

If the virtual type's "Inherit Base Type Policies" option is...	And the policy is defined for the...	The instances of the virtual type will have...
ENABLED	Base Type	Base Type Policies
ENABLED	Virtual Type	Virtual Type Policies
ENABLED	Base Type	Base Type Policies
	—AND—	—AND—
	Virtual Type	Virtual Type Policies
DISABLED	Base Type	None
DISABLED	Virtual Type	Virtual Type Policies
DISABLED	Base Type	Virtual Type Policies
	—AND—	
	Virtual Type	

For more information about virtual types and the **Inherit Base Type Policies** option and which predefined types in CentraSite are virtual types, see [“Inheriting Base Type Profiles, Lifecycle Models, and Policies” on page 263](#).

Note:

The **Inherit Base Type Policies** option does not affect policies that are assigned to the generic Asset type (that is, policies that apply to all assets). Policies that are associated with the Asset type are applied to both base types and virtual types, irrespective of the **Inherit Base Type Policies** setting.

Pre-Operation and Post-Operation Event Types

Many of the event types to which you can apply policies represent *pre-operation* events or *post-operation* events.

- Pre-operation events occur immediately before a requested operation is performed on a registry object. They include PreCreate, PreUpdate, PreStateChange, and PreDelete event types. When you apply a policy to a pre-operation event, CentraSite executes the requested operation only if the pre-operation policy executes successfully. You generally apply policies at these points

to prevent the requested operation from being executed unless an object satisfies the verification checks performed by the policy.

- Post-operation events occur immediately after a requested operation is performed on a registry object. They include PostCreate, PostUpdate, PostStateChange, and PostDelete event types. When you apply a policy to a post-operation event, CentraSite executes the policy only if the requested operation is performed successfully. Post-operation policies are often used to notify users (through email) that certain modifications have occurred in the registry to update specified attribute values and to assign permission settings on an object.

Events during Which Design/Change-Time Policies Can Be Enforced

You can apply design/change-time policies when the following events occur to an object in CentraSite:

Event	Occurs...
Pre-Create	Immediately before CentraSite commits a new object to the registry.
Post-Create	Immediately after CentraSite commits a new object to the registry.
Pre-Update	Immediately before CentraSite commits an update to an existing object in the registry.
Post-Update	Immediately after CentraSite commits an update to an existing object in the registry.
Pre-Delete	Immediately before CentraSite removes an object from the registry.
Post-Delete	Immediately after CentraSite removes an object from the registry.
Pre-State Change	Immediately before a specified lifecycle state change is made to an object.
Post-State Change	Immediately after a specified lifecycle state change is made to an object.
OnConsumerRegistration	CentraSite Control: When an asset owner accepts a pending registration request by clicking the Apply Registration Policies button in the Pending Registrations inbox. CentraSite Business UI: When a consumer requests access for an asset by clicking the Consume button in the Asset Details page.
OnTrigger	When you use the Run Policy Now button in CentraSite Control to run a policy on demand. For more information about this type of event, see Running a Policy on Demand.

Event	Occurs...
OnCollect	When a handler process calls the collector. <i>This event is intended to be used only by predefined policies that perform a collection process.</i> For more information about collectors, see Collector and Handler Policies.
OnExport	When the export handler is called during an export operation. <i>This event is intended to be used only by predefined policies that perform an export operation.</i> For more information about export handlers, see Collector and Handler Policies.
OnMove	When the move handler is called during a move operation (the movement of an asset to another user or organization). <i>This event is intended to be used only by predefined policies that perform a move operation.</i> For more information about move handlers, see Collector and Handler Policies.

Supported Object and Event Combinations

Not all object types support the full set of events. Some events occur only with certain types of objects. For example, a PreStateChange event occurs only on Assets, Policies and Lifecycle Models. If you create a policy for a PreStateChange event on a User object, that policy will never execute, because a PreStateChange event will never occur on a User object.

The following table identifies the events that each object type supports:

	Organization	Taxonomy	User	Policy	Lifecycle Model	Assets	Report Template
Pre-Create	✓	✓	✓	✓	✓	✓	✓
Post-Create	✓	✓	✓	✓	✓	✓	✓
Pre-Update	✓	✓	✓	✓	✓	✓	✓
Post-Update	✓	✓	✓	✓	✓	✓	✓
Pre-Delete	✓	✓	✓	✓	✓	✓	✓
Post-Delete	✓	✓	✓	✓	✓	✓	✓
Pre-State Change				✓	✓	✓	
Post-State Change				✓	✓	✓	
OnConsumer						✓	

	Organization	Taxonomy	User	Policy	Lifecycle Model	Assets	Report Template
Registration							
OnTrigger	✓	✓	✓	✓	✓	✓	✓
OnCollect	✓	✓	✓	✓	✓	✓	
OnExport	✓	✓		✓	✓	✓	
OnMove			✓			✓	

Actions that Design/Change-Time Policies Can Execute

An action in a design/change-time policy instructs CentraSite to perform a specific task, such as submit a request for approval, send an email notification to a group of users, perform a specified test or validate certain properties of an object.

Built-In Actions

CentraSite includes many built-in actions that you can use to compose design/change-time policies. Built-in actions are provided in the following categories:

Category	Descriptions
WS-I Compliance	Actions that test the conformance of a Web service to the basic profiles specified by the Web Services Interoperability organization (WS-I).
Design Time	Actions that you can apply when an object is initially added to CentraSite. In general, the actions in this category are designed to be used during a Pre-Create event.
Change Time	Actions that you can apply when an object within CentraSite is modified or deleted. This category contains actions that you use to obtain approvals, classify objects and perform various types of validation checks.
Global Templates	Actions that perform general tasks such as sending email notifications or setting permissions. The actions in this category can be used with nearly all event types.
ARIS	An action that notifies ARIS when changes occur to the business processes and services that have been published by ARIS to CentraSite.

Custom Actions

If you need to execute a task that is not provided by a built-in action, you can create a custom action to perform the work. A custom action consists of a Java class that performs the required task (for example, running a test, adding a required attribute, or updating an external database).

To use a custom action, you must upload the Java class to the CentraSite repository and define an *action template* for it. An action template specifies the location of your custom code and identifies the parameters that it uses. After you create the action template, your custom action appears in the CentraSite Control user interface and it can be inserted into a policy just like a built-in action.

System-Wide and Organization-Specific Policies

A design or change-time policy is either system-wide or organization-specific. System-wide policies apply to all organizations within an instance of CentraSite. Organization-specific models apply only to a specified organization.

Whether a policy is system-wide or organization-specific affects the policy's scope. When a policy is organization-specific, its scope is limited to the set of objects that belong to a specified organization. For example, if you create a policy that executes on a PreDelete event for XML schemas and you make that policy specific to organization ABC, then that policy executes only when XML schemas in organization ABC are deleted. If you make it system-wide, it executes when an XML schema in any organization is deleted.

Policy Priority

You can give a policy a priority value between 11 and 9999 (inclusive). Priority 11 is the default. Values less than 11 and greater than 9999 are reserved for system use.

The **Priority** setting contains a non-negative integer that indicates the policy's priority. A priority value of 0 represents the highest possible priority. You can give a policy a priority value between 11 and 9999 (inclusive). Priority 11 is the default. Values less than 11 (0 through 10) and greater than 9999 are reserved for system use. These priorities can only be assigned to predefined policies. You cannot assign these priority values to the regular, user-defined policies that you create using CentraSite Control.

Note:

A policy's **Priority** property is used *only* when CentraSite is given multiple policies to enforce for the same event. If CentraSite has only one policy to enforce, the **Priority** property is ignored entirely.

When an event triggers multiple policies, CentraSite examines the priorities of the selected policies and executes the policies serially, in priority order, from the lowest value to the highest value (that is, it executes the policy with the *lowest* value first). Each policy in the series is executed to completion before the next one begins.

For example, if an event were to trigger the following policies, CentraSite would execute the policies in the following order: B, A, C (as determined by the priority assignments 11, 25, and 100 respectively).

Policy

A

B

Policy

C

If two or more policies have the same priority value, their order is indeterminate. CentraSite will execute these policies in serial fashion after all lower priority policies and before any higher priority policies. However, you cannot predict their order.

Example

If CentraSite were given the following policies to enforce for an event:

Policy	Priority
Policy A	11
Policy B	25
Policy C	11
Policy D	100
Policy E (system policy)	0

It would execute the policies in the following order:

Policy	Priority
Policy E (system policy)	0
Policy A then Policy C (or vice versa) The order of these two policies cannot be controlled or predicted because they have the same priority.	11
Policy B	25
Policy D	100

Note:

The preceding example is shown for illustrative purposes. It is not a good practice to have an event trigger policies that have the same priority as exhibited by policies A and C above.

Policy Enforcement Time

The policy-enforcement process begins when a CentraSite user submits a request that acts on one of the object types. Depending on the type of request the user submits, one of the events identified can occur.

When an event occurs, CentraSite queries its database and executes policies that satisfy the following criteria:

- The policy's **Event Type** and **Object Type** settings match the given event type and object type.

– AND –

- The policy's object-selection criteria are satisfied by the object on which the event occurred.

– AND –

- The policy is scoped for the same organization as the object –OR– the policy is *system-wide*, meaning that it applies to all organizations.

Note:

It does not matter what kind of client submits the request or how the request reaches CentraSite. Design/change-time policies are applied to all requests, regardless of whether they come from the CentraSite Control user interface, a UDDI client or a JAXR-based client. Be aware that the execution of a policy can itself trigger another policy. This can occur if an action in a policy performs an operation that is within the scope of another policy.

Multiple Policies Triggered by an Event

When multiple policies have the same scope, all of those policies are triggered when an in-scope event occurs. To determine the order in which to execute these policies, CentraSite examines each policy's **Priority** setting.

Unsuccessful Design and Change-Time Policy Actions

When an action in a design/change-time policy completes its execution, it returns a completion code and a completion message. This information is written to the policy log if CentraSite is configured to log successful policies.

If the completion code indicates success, CentraSite performs the next action in the policy (if one exists) or completes the requested work on the object (for example, it commits the given change to the database).

If the completion code indicates failure, CentraSite records the error in the policy log. Then it immediately exits the policy. If the policy contains additional actions, those actions are not executed. If the policy was triggered by one of the pre-commit events (for example, during a Pre-Create, Pre-Update or Pre-State Change event) the requested operation is not performed. If the initial request had triggered multiple policies, any policy that had not yet been executed will be bypassed.

Predefined Policies Used by CentraSite

CentraSite includes a set of *predefined policies* that execute internal operations on the registry. Many of these policies relate to operations such as deleting objects, exporting objects and moving objects. By default, predefined policies are not shown in the policy list, however, you can view them by enabling the **Show Predefined Policies** option.

System policies often execute on events such as OnCollect, OnMove, and OnExport. For example, the OnMove event triggers the Default Move Handler policy, which makes the changes necessary to move an asset from one organization and user to another. If your site has special requirements, it can override certain predefined policies. For example, if you have an asset type that is not suitably exported by CentraSite's Default Export Handler policy, you can develop your own export handler policy to export assets of that type. .

Important:

If you belong to a role that includes the Manage System-Wide Design/Change-Time Policies permission, you have the ability to edit, delete, and deactivate CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be edited, deleted, or deactivated except under the direction of a technical representative from Software AG.

Execution of Design/Change-Time Policy

When an event occurs in the registry, CentraSite determines which policies are within the scope and executes those policies in priority order (from lowest assigned value to the highest assigned value). If an action within a policy fails, CentraSite immediately exits the policy. It does not execute any of the remaining actions in the policy nor does it execute any remaining policies that are within scope of the event.

If the policy was triggered by a pre-operation event (for example, a PreCreate event or a PreStateChange event) the requested operation is also not executed. For example, if a user attempts to add an XML Schema to the catalog, and the schema does not satisfy a validation policy that is triggered by the PreCreate event for XML Schemas, CentraSite rejects the user's request to add the new schema to the catalog.

Policy failures are written to CentraSite's policy log. From the Inbox page in CentraSite Control, users can view the failed policies that were logged during the events that they initiated. Administrators with View Policy Log permission can view and query the entire log using CentraSite Control's Logging feature.

CentraSite provides a special event type called an OnTrigger event. Policies that you create for this event type can be run on demand from the CentraSite Control user interface. Anyone who has View permission on an OnTrigger policy can execute the policy on demand.

When you run a policy on demand, CentraSite applies the policy directly to each object instance in the registry that:

- Is of a type specified in the policy's object scope.
- Satisfies all conditional criteria specified by the policy (that is, Name, Description, and Classification criteria that the policy specifies).
- Is an object on which the user running the policy has View permission. If the policy is organization-specific, the policy is applied to only the objects that satisfy the preceding criteria and belong to the organization specified by the policy. If the policy is system-wide, the policy is applied to all objects in the registry that satisfy the preceding criteria.

Administrators often use OnTrigger policies to assign permissions to a specified set of objects instead of manually setting permissions on individual objects using the user interface.

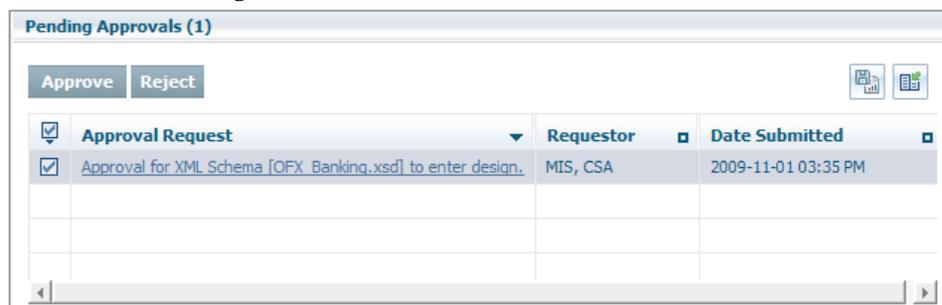
Design/Change-Time Policies Usage

Using Policies to Initiate Reviews and Approvals

Enforcing a review and approval process is a common use of a Design/Change-Time policy. To create this type of policy, you include one of CentraSite's approval actions in the policy. An approval action identifies the group of users (the approver group) whose approval is required in order to complete the policy successfully. When you configure an approval policy, you can specify whether approval is required from just one approver in the group or from all approvers in the group.

You apply approval policies to PreStateChange events. Thus, to use approval policies on assets, those assets must be under lifecycle management.

When CentraSite executes an approval policy, it initiates an approval workflow. Users who are designated approvers review the request on the **Pending Approvals** tab of their Inbox page in CentraSite Control user interface. If the approvers approve the request, CentraSite executes the requested state change. If an approver rejects the request, the policy fails and the requested state change is not executed. Pending approvals appear in the approvers inbox in CentraSite Control as shown in the figure:



Using Policies to Validate Assets

Validation is another common task that you can perform using Design and Change-Time policies. You use validation policies to ensure that assets conform to organizational norms and standards before they are accepted into the catalog or enter a critical lifecycle state. For example, you may create validation policies to:

- Ensure that a service satisfies certain naming conventions, that it exists within a specified namespace or supports specified protocols.
- Ensure that an asset has been classified by a specified taxonomy and includes required attribute settings.
- Ensure that a service complies with Web Service Interoperability (WS-I) standards (Basic Profile 1).
- Prevent an asset from being deleted unless it has reached a specified state within its lifecycle.

CentraSite provides built-in actions that you can use to perform many validation tasks. You can also create custom actions to perform validation tasks that are specific to your environment.

Using Policies to Modify Assets

You can use Design/Change-Time policies to make changes to assets when certain events occur. Setting instance-level permissions on an asset is an example of an update you may incorporate using a policy. Other kinds of changes you may incorporate include classifying an asset, setting a specified attribute when an asset completes a series of tests, and changing an asset's lifecycle state.

CentraSite provides built-in actions that you can use to perform many kinds of updates to assets. You can also create custom actions to modify assets.

Using Policies to Issue Notifications or Update External Systems

Notifying individuals when certain events occur in the registry is another common use case for Design/Change-Time policies. CentraSite includes a built-in email action that you can use for this purpose.

You may, for example, use this action to send an email to key users in your IT organization when a service switches to the Production state of its lifecycle. You may also create custom actions that would send messages to and trigger processes on external systems when certain events occur in the CentraSite registry.

Using Policies to Execute a Consumer Registration

The consumer-registration feature in CentraSite enables users to register users and applications as consumers of an asset. This enhanced feature enables you to complete the registration process without explicitly creating a consumer-registration policy and without requiring the owner of the asset to review and accept the registration request.

If you want to impose an approval process on the consumer registration feature, you might create a design-time policy with one of the CentraSite's built-in approval actions for the OnConsumerRegistration event. This policy includes the Register Consumer action, which performs the actual registration process. It can optionally include other actions, such as Set Consumer Permission action, as needed.

Using Policies to Manage the Deployment of Virtual Services

If you use virtual services that are enabled with the LCM, you must create policies that enable an administrator to deploy a virtual service to webMethods Mediator and to make a virtual service undeployable while it is being revised.

Issues to Consider When Creating Design/Change-Time Policies

The following are issues that may occur when creating Design/Change-Time policies for your CentraSite registry:

- When an event occurs in the registry, CentraSite executes *all* policies whose scope encompasses the event. Use priorities to control the order in which CentraSite executes the policies. Consider

assigning something other than the default priority of 11 to routine policies. Doing this enables you to more easily interject higher priority policies for the events associated with those policies.

- Many of the built-in actions provided with CentraSite (including the approval actions) are scoped for state-change events and can only be used with objects that are under lifecycle management. Before you create a policy, determine which policy actions you want to use and verify the event types that they support. To create certain types of policies, you may have to first apply a lifecycle model to the objects that you intend to govern with those policies.
- *Exercise caution when using OnTrigger policies!* If the policy updates objects, the scope of the policy is set precisely and targets only the set of objects that you intend to change. If used incorrectly, this type of policy can change objects in ways that cannot easily be undone.
- By default, CentraSite is configured to record only policy failures in the policy log. You can optionally configure CentraSite to log both successful and failed policies in its log. However, if you do this, the log can grow rapidly. Consider logging successful policy executions only when it is necessary for tracing or troubleshooting purposes or if you enable this option as part of your normal operations, consider purging the log on a regular basis.
- When CentraSite executes a policy, the policy's actions are executed on behalf of the user who triggered the policy (the actions are executed under that user's account). If the user does not have the permissions necessary to complete an operation initiated by a policy action, the action and the policy fails. For example, if a user triggers a policy that sets permissions on an object, the policy fails unless the user has full permission on the object. (Only users with full permission on an object are allowed to change the object's permission settings.)

Managing Design Time Policies through CentraSite Business UI

This section describes operations you can perform to manage the design time policies through CentraSite Business UI.

Creating a Design Time Policy

Pre-requisites:

To create a new Design Time policy, you must have one of the following permissions in CentraSite:

- To create policies for a specific organization, you must have the Manage Design Time Policies permission for that organization. By default, users in the CentraSite Administrator, Organization Administrator, or Policy Administrator role have this permission.
- To create system-wide policies (that is, policies that apply to all organization within an instance of CentraSite), you must have the Manage System-Wide Design Time Policies permission. By default, users in the CentraSite Administrator role, Asset Type Administrator role, and Operations Administrator role have this permission.

➤ [To Create a Design Time Policy](#)

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. Click **Add Design Time Policy**.

The Add Policy page appears.

3. In the **Basic Information** profile, specify basic information about the policy.

In this field...	Do the following...
Name	Type a name for the new policy. A policy name can contain any character (including spaces). A policy name does not need to be unique within the registry. However, to reduce ambiguity, you should avoid giving multiple policies the same name. As a best practice, Software AG recommends that you adopt appropriate naming conventions to ensure that policies are distinctly named within your organization.
Description	(Optional). Type a description for the new policy. This description appears when a user displays a list of policies in the user interface.
Version	(Optional). Specify a version identifier for the new policy. Note: The version identifier does not need to be numeric. Examples: <pre>0.0a 1.0.0 (beta) Pre-release 001 V1-2007.04.30</pre> The version identifier you type here is the policy's public, user-assigned version identifier. CentraSite also maintains an internal, system-assigned version number for the policy.
Priority	Type an integer that represents the priority of this policy with respect to other policies that might be triggered by the same event. The priority value determines the order in which the policies are enforced. The lower the Priority value, the higher the priority (that is, 0 is the highest priority and policies with this priority value are executed first). <ul style="list-style-type: none">■ Priority values 0 through 10 and values greater than 9999 are reserved for predefined policies. You cannot assign these values to the user-defined policies that you create in CentraSite Business UI.■ The default priority for a user-defined policy is 11.
Organization	Specify the organization to which the policy applies or select All if the policy applies to all organizations. Note:

In this field... Do the following...

The **Organization** list contains the names of all organizations for which you have `Manage Design Time Policies` permission. The option appears in the list if you also have `Manage System-Wide Design Time Policies` permission.

4. In the **Scope** profile, specify Applicable Types and Event Types to which the policy applies.

In this field... Do the following...

Applicable Types Specify the types of assets applicable for this policy.

Note:

When you select a base type, for example, a Service asset type that has virtual types, for example, Virtual Service, Virtual REST Service, associated to it, and the **Inherit Base Type Policies** option is enabled for certain of the virtual types, be aware that the policy you create is applied to instances of the base type *and* instances of the virtual types.

You can optionally restrict a policy to specific instances of the selected Applicable Types by specifying additional selection criteria.

Event Types Specify the types of events applicable for this policy.

Important:

The OnCollect, OnMove and OnExport events are designed to execute handler and collector processes. *Do not* use these events unless you are creating a handler or collector policy. The improper use of these event types can damage the registry.

Filter Criteria Specify the selection criteria to narrow the set of Applicable Types applicable for this policy . You can filter the Applicable Types by Name, Description, and Classification attributes.

5. In **Policy Actions**, select the actions that you want CentraSite to execute for a policy. To add a policy action, perform the following:
 - a. Based on your selection for the Scope profile, the system shows the policy actions available for this new policy on the left side menu.
 - b. Click  next to a policy action to add it to the Policy Actions page.
 - c. If you want to delete an action from the Policy Actions page, click  next to the up and down arrows in the policy action row.

Remember the following points when you select the actions for the policy:

- In the **Scope** profile, you have to select at least one Event Type. If not, when you click on a policy action, the system shows an error message and the policy action page appears empty.
- The actions shown in the **Policy Actions** list are determined by the Applicable Types and Event Types that you specify on the **Scope** profile. If you do not see an action that you need, that action is probably not compatible with all of the Applicable Types and Event Types that you selected in the **Scope** profile.
- If necessary, you can click the **Scope** profile and change your Applicable types and Event types selections.
- Ensure that the actions in the area labeled **Policy Actions** appear in the order that you want CentraSite to execute them at enforcement time. If necessary, use the up and down arrows to place the actions in the correct order.
- Be aware that actions from the WS-I category cannot be combined with other types of actions. Also be aware that when you add a WS-I action to the action list, CentraSite automatically adds dependent actions to the list as necessary.

6. Complete the policy by doing the following:

- a. Configure the parameters for each action in the **Policy Actions** area by clicking the policy action. Clicking the policy action more than once toggles the form.

Note:

Ensure that you specify values for any mandatory parameters. If not, you cannot save the form.

- b. If the policy is to be enforced during a PreStateChange or PostStateChange event, ensure that the options on the **States** profile specify the lifecycles and state changes to which the policy applies. By default, the **States** profile is disabled. When you select Pre-State Change or Post-State Change Event Type, the system automatically enables this profile.

Note:

Listing of Pre-State or Post-State Event Type depends on the **Applicable Types** and **Organization** selections.

- c. If you want to allow other users to view, edit, or delete a policy, you need the necessary permissions. You can assign permissions to a policy only after the successful creation of a policy.

7. Click **Save** to save the new policy.

Note:

You can save a policy even without adding any policy actions.

8. When you are ready to put the policy into effect, activate the policy.

Activating a Design Time Policy

Pre-requisites:

To activate a Design Time policy in CentraSite, you must have one of the following permissions:

- To activate a organization-specific policy, you must have the Manage Design Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to activate that policy unless you have permission to manage Design time policies for organization ABC.
- To activate a system-wide policy, you must have the Manage System-Wide Design Time Policies permission.

Important:

If you belong to a role that includes the Manage System-Wide Design Time Policies permission, you have the ability to activate CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be activated except under the direction of a technical representative from Software AG.

CentraSite does not begin enforcing a Design Time policy until you *activate* it.

Policies are managed by a predefined lifecycle installed with CentraSite. This lifecycle, called the Policy Lifecycle, defines the following lifecycle states: New, Productive, Suspended, and Retired.

To activate a policy, change the lifecycle state of a policy to the Productive state.

When you activate a policy, be aware that:

- You are not be allowed to activate the policy unless all of its mandatory parameters are set. When you switch the policy to the Productive state, CentraSite executes the *Validate Policy Activation* policy. This policy does not allow you to switch a policy to the Productive state if the mandatory parameters of the policy are not set.
- Some organizations require an approval to activate a policy. If your organization has an approval action associated with the activation of a policy, CentraSite does not activate the policy until the required approvals are obtained.
- If an earlier version of the policy is already active, CentraSite deactivates the old version before it activates the new one.
- When a policy becomes active, CentraSite enforces it immediately. You can suspend enforcement of a policy by switching it to the Suspended state.

Note:

When a policy whose Event Type is set to OnTrigger is activated, the Run Now policy action is enabled.

- To activate a policy, you must have permission to change the policy to the Productive state.

To determine whether a policy is active or inactive, examine the policy's decoration indicator on the Search Results page. The decoration indicates the policy's activation state as follows:

Decoration	Description
	Policy is active.
	Policy is inactive.

> To activate a Design Time policy

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. To view the Design Time policies:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, perform the following actions:
 - a. Locate **Applicable Scopes**. By default, Policy is selected. Additionally, you can also refine your search further by selecting the following options:
 - Show only Productive Policies
 - Include Predefined Policies
 - b. Add a Keyword and click the plus button next to the field to add it to the Current Search Criteria recipe.
 - c. You can also further refine your search by selecting values for the following from their respective drop-down list and clicking the plus sign next to these fields to add them to the search criteria:

Note:

The moment you click the plus sign, the result is refreshed in the Search Results page.

- Applicable Organizations
 - Applicable Types
 - Applicable Event Types
- d. This displays a list of defined Design Time policies in the Search Results page.
3. Click the policy you want to activate.

This opens the Design Time Policy Details page.

4. Examine the **Policy Actions** profile and verify that all of the parameters for the policy action are configured properly. If any of the actions are not configured, set their parameters before you continue.

- Click **Activate** to activate the policy. If you do not see the **Activate** button, it could be because you do not have permission to change the lifecycle state of a policy. Also, when a policy is Active, only the Deactivate button is available and vice versa.

You can also activate a policy by toggling the toggle icon on the Design Time activity page.

- Examine the decoration indicator of the policy on the Search Results page to verify that the state of the policy has changed.

If this state change requires approval, the decoration of the policy indicates that the policy is in the pending mode. CentraSite automatically switches the policy to the **Productive** state (and activates the policy) after all the necessary approvals are obtained.

Deactivating a Design Time Policy

Pre-requisites:

To deactivate a Design Time policy in CentraSite, you must have one of the following permissions:

- To deactivate a organization-specific policy, you must have the Manage Design Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to deactivate that policy unless you have permission to manage design time policies for organization ABC.
- To deactivate a system-wide policy, you must have the Manage System-Wide Design Time Policies permission.

Important:

If you belong to a role that includes the Manage System-Wide Design Time Policies permission, you can deactivate CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be deactivated except under the direction of a technical representative from Software AG.

Deactivating a Design Time policy causes CentraSite to suppress enforcement of the policy. You usually deactivate a policy for the following reasons:

- To suspend enforcement of a particular policy (temporarily or permanently).
- To edit a policy (for example, to modify the scope of a policy or change its action list).

To deactivate a policy, move the policy to the Suspended state. Switching the policy to this state triggers the *Automatic Policy Deactivation* policy, which deactivates the policy. If you plan to delete a policy, the policy should be in the Suspended state. Delete operation moves the policy from Suspended to Retired state and then deletes the policy. Also, when you activate a newer version for a policy, the current version of the policy moves to **Retired** state.

When you deactivate a policy, CentraSite does not deactivate a policy if it is in the process of being executed. If you attempt to deactivate a policy while it is executing, your state change request fails. If this occurs, wait for a period time and then try to deactivate the policy again.

Note:

Unless all approvals triggered by a policy is complete, you cannot deactivate that policy.

To determine whether a policy is active or inactive, examine the policy's decoration indicator on the Search Results page. The decoration indicates the policy's activation state as follows:

Decoration	Description
	Policy is active.
	Policy is inactive.

> To deactivate a Design Time policy

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. To view the Design Time policies:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, perform the following actions:
 - a. Locate **Applicable Scopes**. By default, Policy is selected. Additionally, you can also refine your search further by selecting the following options:
 - Show only Productive Policies
 - Include Predefined Policies
 - b. Add a Keyword and click the plus button next to the field to add it to the Current Search Criteria recipe.
 - c. You can also further refine your search by selecting values for the following from their respective drop-down list and clicking the plus sign next to these fields to add them to the search criteria:

Note:

The moment you click the plus sign, the result is refreshed in the Search Results page.

- Applicable Organizations
 - Applicable Types
 - Applicable Event Types
- d. This displays a list of defined Design Time policies in the Search Results page.
3. Select the policy you want to deactivate.

This opens the Design Time Policy Details page.

4. Examine the **Policy Actions** profile and verify that all of the parameters for the action are configured properly. If any of the actions are not configured, set their parameters before you continue.
5. Click **Deactivate** to deactivate the policy. (If you do not see the **Deactivate** button, it is probably because you do not have permission to change the lifecycle state of a policy.)

You can also deactivate a policy by toggling the decoration indicator for the policy in the Search Results page.

6. Examine the decoration indicator of the policy on the Search Results page to verify if the state of the policy has changed. In the policy details page, the Deactivate icon is enabled if the policy is in Productive state (active policy) and the same is indicated by means of the toggle icon in the Manage Design Time activity page.

If this state change requires approval, the decoration indicator of the policy indicates that the policy is in the pending mode. CentraSite automatically switches the policy to the **Suspended** or **Retired** state (and deactivates the policy) after all the necessary approvals are obtained. Once the policy is moved to the **Retired** state, the toggle icon is no longer visible.

Viewing Design Time Policy List

The **Design Time** activity displays the design time policies in CentraSite. This list displays policies for all organizations, not just your own. It also includes system-wide policies.

By default, the policy list displays only the user-defined policies in CentraSite . If you want to view only the active policies, you must select the **Show only Productive Policies** option.

> To view the design time policy list

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. To view the design time policies:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, perform the following actions:
 - a. Locate **Applicable Scopes**. By default, Policy is selected.
 - b. Additionally, you can also refine your search further by selecting the following options:
 - Show only Productive Policies
 - Include Predefined Policies
 - c. To further filter the design time policies in CentraSite:

To see...	Do this...
List of active design time policies	Enable the Show only Productive Policies option.
List of predefined policies	Enable the Include Predefined Policies option.
Subset of the defined design time policies matching a keyword	Type a partial string in the Keyword text field. Click the plus button next to the drop-down list or press <i>Enter</i> to add the keyword to the search recipe.
List of design time policies whose scope applies for a particular organization	<ol style="list-style-type: none">In the Applicable Organizations section, select an organization from the drop-down list.Click the plus button next to the drop-down list or press <i>Enter</i> to add the selected organization to the search recipe.
List of design time policies whose scope applies for all organizations	<ol style="list-style-type: none">In the Applicable Organizations section, select All.If any change of organization, select an organization from the drop-down list and click the plus button next to the organization name or press <i>Enter</i> to add the selected organization to the search recipe.
List of design time policies whose scope applies for a particular asset type	In the Applicable Types section, select an asset type from the drop-down list.
List of design time policies whose scope applies for a particular event type	In the Applicable Event Types section, select an event type from the drop-down list.

3. The Search Results page provides the following information about each policy:

Column	Description						
Name	The name assigned to the policy.						
Description	Additional comments or descriptive information about the policy.						
Priority	The priority value assigned to the policy.						
Organization	The organization to which this policy is applicable.						
	<table><thead><tr><th><u>This value...</u></th><th><u>Indicates that...</u></th></tr></thead><tbody><tr><td>All</td><td>The policy is system-wide and applies to all organizations.</td></tr><tr><td><i>OrgName</i></td><td>The policy applies to the specified organization.</td></tr></tbody></table>	<u>This value...</u>	<u>Indicates that...</u>	All	The policy is system-wide and applies to all organizations.	<i>OrgName</i>	The policy applies to the specified organization.
<u>This value...</u>	<u>Indicates that...</u>						
All	The policy is system-wide and applies to all organizations.						
<i>OrgName</i>	The policy applies to the specified organization.						

Column	Description						
Owner	The user to which the policy belongs.						
Version	The user-assigned version identifier for the policy.						
Last Updated Date	The date on which the policy was last updated. CentraSite automatically updates this attribute when a user modifies any of the policy's attributes.						
State	The current lifecycle state of the policy.						
Applicable Event Types	The event types applicable for the policy.						
Applicable Types	The asset types associated with the policy.						
Active	The current enforcement state of the policy.						
	<table border="1"> <thead> <tr> <th>Icon</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>The policy is active. CentraSite enforces this policy when events within the scope of the policy occur.</td> </tr> <tr> <td></td> <td>The policy is inactive. Inactive policies exist in CentraSite, but they are not enforced.</td> </tr> </tbody> </table>	Icon	Description		The policy is active. CentraSite enforces this policy when events within the scope of the policy occur.		The policy is inactive. Inactive policies exist in CentraSite, but they are not enforced.
Icon	Description						
	The policy is active. CentraSite enforces this policy when events within the scope of the policy occur.						
	The policy is inactive. Inactive policies exist in CentraSite, but they are not enforced.						

Viewing or Modifying a Design Time Policy

Pre-requisites:

To examine and modify the properties of a design time policy in CentraSite, you must have one of the following permissions:

- To examine and modify the properties of an organization-specific policy, you must have the Manage Design Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to examine and modify the properties of that policy unless you have permission to manage design time policies for organization ABC.
- To examine and modify the properties of a system-wide policy, you must have the Manage System-Wide Design Time Policies permission.

Important:

If you belong to a role that includes the Manage System-Wide Design Time Policies permission, you have the ability to modify CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be modified except under the direction of a technical representative from Software AG.

You cannot modify a policy while it is in the Productive state. To make changes to a policy, you can do any of the following:

- Create a new version of the policy, make the necessary changes to the new version and switch the new version to the Productive state when you are ready to put it into effect. Switching the new version to the Productive state will immediately put the previous version in the Retired state. (The Retired state is an end state. After you place a policy in this state, you can no longer reactivate it.)
- Create a completely new policy that includes the required changes. When you are ready to put the new policy into effect, deactivate the old policy and activate the new policy.
- Deactivate the existing policy and make the necessary changes to the policy and then activate it. While the policy is in the Deactivated state, it will not be enforced. This is because deactivating the policy results in an enforcement gap; however, you would not use this approach in a production environment.

➤ **To examine and modify details of a design time policy**

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. To view the Design Time policies:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, perform the following actions:
 - a. Locate **Applicable Scopes**. By default, Policy is selected. Additionally, you can also refine your search further by selecting the following options:
 - Show only Productive Policies
 - Include Predefined Policies
 - b. Add a Keyword and click the plus button next to the field to add it to the Current Search Criteria recipe.
 - c. You can also further refine your search by selecting values for the following from their respective drop-down list and clicking the plus sign next to these fields to add them to the search criteria:

Note:
The moment you click the plus sign, the result is refreshed in the Search Results page.

 - Applicable Organizations
 - Applicable Types
 - Applicable Event Types
 - d. This displays a list of defined Design Time policies in the Search Results page.
3. Click the policy whose details you want to examine and modify.

This opens the Design Time Policy Details page.

4. If the policy is active, deactivate it by clicking  from the menu.

You cannot modify the details of an active policy.

5. On the Design Time Policy Details page, examine and modify the attributes as required.

Field	Description
Name	The name of the policy. A policy name can contain any character (including spaces). A policy name does not have to be unique within the registry. However, to reduce ambiguity, you should avoid giving the same name to multiple policies. As a best practice, we recommend that organizations adopt appropriate naming conventions to ensure the assignment of distinct policy names.
Description	(Optional). Additional comments or descriptive information about the policy.
Version	<p>The user-assigned version ID assigned to this policy. You can use any versioning scheme for identifying different versions of a policy. The identifier does not have to be numeric. Examples:</p> <p>0.0a</p> <p>1.0.0 (beta)</p> <p>Pre-release 001</p> <p>V1-2007.04.30</p> <p>CentraSite also maintains a system-assigned version identifier for a policy. The system-assigned version identifier is independent from the version identifier that you specify in this field.</p>
Priority	<p>An integer that represents the priority of this policy with respect to other policies that might be triggered by the same event.</p> <ul style="list-style-type: none"> ■ Priority values 0 through 10 and values greater than 9999 are reserved for predefined policies. You cannot assign these values to the user-defined policies that you create in CentraSite Business UI. ■ The default priority for a user-defined policy is 11.
Scope	The settings on this profile specify the Applicable Types and Event Types that are applicable to the policy.
States	The settings on this profile specify the lifecycles and state changes to which this policy applies.

Note:

Field	Description
	The States profile is present only if the scope of the policy includes a Pre-State Change or Post-State Change event.
Policy Actions	The settings on this profile, specify the actions that CentraSite executes when the policy is enforced.

- Click **Save**.

Note:

If the selected asset and event types are not compatible with the current set of actions in the action list, CentraSite does not permit you to save the policy. You must correct the policy's action list or its scope to save the policy successfully.

- When you are ready to put the policy into effect, activate the policy.

Scope of a Design Time Policy

Scope refers to the set of properties that determine when a policy is enforced. For a design time policy, scope is determined by the **Applicable Types**, **Event Types**, and **Organization** properties of the policy, which are described here:

Property	Description
Applicable Types	<p>The list of asset types applicable to the policy.</p> <p>Note: When you select a base type, for example, a Service asset type that has virtual types, for example, Virtual Service, Virtual REST Service, associated to it, and the Inherit Base Type Policies option is enabled for certain of the virtual types, be aware that the policy you create is applied to instances of the base type <i>and</i> instances of the virtual types.</p> <p>You can optionally restrict a policy to specific instances of the selected Applicable Type by specifying additional selection criteria.</p>
Event Types	<p>The list of event types applicable to the policy.</p> <p>Note: Not all Event Types occur for all Applicable Types.</p> <p>Important: The OnCollect, OnMove, and OnExport events are designed to execute handler and collector processes. <i>Do not</i> use these events unless you are creating a handler or a collector policy. The improper use of these event types can damage the registry.</p>

Property	Description
Organization	Determines whether the policy belongs to a specific organization or is system-wide.

System-wide and Organization-specific Policy Enforcement

The **Organization** property specifies the organization to which the policy is applicable. When the **Organization** property is set to `ALL`, it indicates that the policy is *system-wide*. When the **Organization** property specifies a particular organization, it indicates that the policy is *organization-specific*.

Organization-specific Policies

An organization-specific policy is enforced on objects that belong to the same organization as the organization to which the policy applies. For example, if you have a policy that executes when user objects are updated and its **Organization** property specifies organization ABC, CentraSite executes that policy only when user objects in *organization ABC* are updated.

Remember these points when working with organization-specific policies:

- You can create organization-specific policies for any organization for which you have Manage Design Time Policies permission. For example, if you have Manage Design Time Policies permission for organization ABC and XYZ, you can create organization-specific policies for either organization.
- At enforcement time, CentraSite selects policies based on the organization to which the object belongs, and *not* the organization to which the requestor belongs. For example, if a user from organization XYZ edits an asset in organization ABC, CentraSite applies the organization policies of ABC (the organization to which the asset belongs), and not the organization policies of XYZ (the organization to which the requestor belongs).

System-wide Policies

A system-wide policy is enforced for all organizations. For example, if you create a system-wide policy that is executed when an asset is created, CentraSite enforces the policy whenever *any* user in *any* organization adds an asset to the catalog.

To create a system-wide policy, you must belong to a role that has Manage System-Wide Design Time Policies permission. In a standard CentraSite configuration, only users with the CentraSite Administrator role and the Policy Administrator role have this permission.

System-wide policies are useful for managing many types of objects. For example, they are often used to assign users to certain server-wide groups or to enforce server-wide naming conventions on objects. However, organization-specific policies are often better choices for asset-related policies, because they enable an organization to tailor its policies to its own development processes and methodologies.

Modifying Scope of a Design Time Policy

Use the **Scope** profile on the Design Time Policy Details page to specify a scope for the policy.

Scope changes are limited by the set of actions currently selected on the **Policy Actions** profile. That is, CentraSite does not allow you to save a policy if its scope includes objects and events that are not compatible with the current set of specified actions. In some cases, you might need to clear actions from the **Policy Actions** profile in order to select the applicable types and event types you need on the **Scope** profile.

➤ To modify the scope of a design time policy

1. Open the Design Time Policy Details page to view the Design Time policies. For more information, see [“Viewing Design Time Policy List” on page 689](#).
2. Select the policy that you want to modify.

This opens the Design Time Policy Details page.
3. If the policy is active, deactivate it.

You cannot modify the scope of an active policy.
4. Select the **Scope** profile and specify the following:
 - In the **Applicable Types** and **Event Types** lists, select the asset types and event types applicable to the policy.
 - *Optional.* In the **Filter criteria**, specify additional selection criteria to narrow the set of assets to which this policy is applied.
5. Click **Save**.

Note:

If the selected asset and event types are not compatible with the current set of actions in the action list, CentraSite does not permit you to save the policy. You must correct the action list of the policy or its scope to save the policy successfully.

6. When you are ready to put the policy into effect, activate the policy.

Refining the Object Scope

To further restrict the set of objects to which the policy is applicable, you can specify additional selection criteria in the **Filter criteria** section of the **Scope** profile. Using this section, you can filter objects by Name, Description, and Classification attributes.

Filtering By Name and Description

You can filter objects based on their Name and Description attributes using any of the following comparison operators:

Comparison Operator	Description
Equals	Selects objects whose Name or Description value matches a given string of characters. For example, you would use this operator if you want to apply a policy only to Taxonomy objects with the Description value Project IDs.
Contains	Selects objects whose Name or Description property includes a given string of characters anywhere within the value of the property. For example, you would type Fairfax in the Value field to use this operator if you wanted to apply a policy to Application Server objects that has the word Fairfax anywhere in their Description property.
StartsWith	Selects objects whose Name or Description property begins with a given string. For example, you would use this operator if you wanted to apply a policy only to web services whose name begins with the characters UTIL-.

When specifying match strings for the comparison operators described above, remember these points:

- Match strings *are not* case-sensitive. If you define a filter for names that start with ABC, it will select names with abc and Abc (and other variations) as well as ABC.
- Wild card characters are not supported. That is, you cannot use characters such as * or % to represent any sequence of characters. These characters, if present in the match string, are simply treated as literal characters that are to be matched.

Filtering By Classification Attribute

You can also filter objects based on the way in which they are classified. When you filter objects in this way, CentraSite applies the policy to objects that have at least one classification whose value matches a specified taxonomy category. For example, you could use a classification filter to apply a policy to those Application Servers objects that are classified as JBoss servers.

When you filter objects by classification, CentraSite inspects all of the classifications for an object at enforcement time. If any of those attributes contain the exact category specified by the selection criteria, the policy is executed.

Note:

To satisfy the selection criteria, the attribute value in the object must match the category specified in the selection criteria *exactly*. Sub-categories of the specified category *are not* considered to be matches. For example, if you have a taxonomy category called Project ABC, and that category has the subcategories Project ABC Design, Project ABC Development, and Project ABC Deployment. If you filter for category Project ABC, CentraSite will apply the policy to objects that are classified by the specific category Project ABC but not objects that are classified by the sub-categories of that category.

➤ **To refine the object scope of a design time policy**

1. Open the Design Time Policy Details page to view the Design Time policies. For more information, see [“Viewing Design Time Policy List” on page 689](#).
2. Click the policy whose object scope you want to refine.

This opens the Design Time Policy Details page.
3. Select the **Scope** profile of the policy.
4. To filter by Name or Description, perform the following in the **Filter criteria** section:

You cannot modify a policy that is in Active state. You must first deactivate a policy before you make any changes.
 - a. Select **Name** or **Description**.
 - b. Select the comparison operator.
 - c. Specify the match string.
5. To filter by object classification, perform the following steps in the **Filter criteria** section of the tab:
 - a. Select **Classification**.
 - b. Click **Browse** and select the category by which you want to filter objects.
6. To specify additional criteria, click the plus button and repeat steps 6 and 7.

Important:

If you specify multiple filters, the policy is applied only if the object matches *all the selection criteria* (that is, the selection criteria is combined using an AND operator, not an OR).

Configuration of Policies that Execute on Lifecycle State Changes

If you create a policy that executes on the Pre-State Change or Post-State Change event type, you must configure the **States** profile of the policy. These settings identify the specific state changes that triggers the policy. This setting also specifies whether the policy is to be executed before or after the object is switched to a specified state.

When creating policies that execute on state changes, remember the following points:

- Policies that are triggered by a state change execute when an object *switches to* a specified state (called the *target state*). The state of the object prior to the change is immaterial. For example, if you have a lifecycle model with the states: *Test*, *Production*, and *Offline* and you have a policy that specifies the *Offline* target state, that policy executes anytime the object switches to the *Offline* state. It does not matter whether the transition occurs from the *Test* state or the *Production* state.

- Policies that are triggered by a state change are executed regardless of whether the state change is initiated from the CentraSite Business UI, the API (for example, a custom client program) or another policy.
- You cannot specify a target state on the **States** profile unless that state has already been defined in a lifecycle model. Additionally, the lifecycle model must be active. In other words, you cannot completely configure a policy that executes on a state change until you have created and activated the lifecycle model whose state(s) triggers the policy.
- If you configure the policy to execute before the state of the object is changed (that is, on a Pre-State Change event) and if any action in the policy fails, the state change does not occur.

Assigning Actions to a Design Time Policy

The **Actions** profile of a policy on the Design Time Policy Details page specifies the list of actions that you want CentraSite to execute when it enforces the policy. CentraSite executes actions in the order in which they appear in the list.

The action list can include any built-in or custom actions that are compatible with the scope of the policy (as currently specified on the **Scope** profile of the policy).

Policy Scope and Action Scope

Similar to a policy, an action has a declared scope. The scope of an action is declared in the **Applicable Types** and **Event Types** properties in the *action template* of an action. An action template is an object that defines a policy action that is available within CentraSite.

A policy can only include actions whose scope *matches or exceeds* the own scope of the policy. For example, if you had an action ABC with the following scope:

Action ABC's Scope

Applicable Type(s):	Service
Event Type(s):	Post-Create Post-Update

You could use this action in policies 1 and 2 below, because these policies include only objects and events that are encompassed by scope of the action. However, you cannot use the action in policies 3 or 4, because these policies include objects and events that the action does not support.

Compatible with Action ABC? Policy #1 Scope

Applicable Types(s):	Service	Yes
Event Type(s):	Post-Create	Yes

Compatible with Action ABC? Policy #2 Scope

Applicable Types(s):	Service	Yes
----------------------	---------	-----

Compatible with Action ABC? Policy #2 Scope

Event Type(s): Post-Create Post-Update Yes

Policy #3**Compatible with Action ABC? Policy Scope**

Applicable Types(s): Service Report Template (*out of scope*) No

Event Type(s): Post-Create Yes

Policy #4**Compatible? Policy Scope**

Applicable Types(s): Service Yes

Event Type(s): Post-Create Post-Update No
Post-Delete (*out of scope*)

Note:

Virtual types and base types are treated as distinct object types with respect to policy action scope. You cannot insert a policy action that is scoped for a particular virtual type into a policy that is scoped specifically for the base type.

Modifying the Action List**Pre-requisites:**

To modify the action list of a design time policy in CentraSite, ensure that you have the required permissions. For more information, see [“Viewing or Modifying a Design Time Policy” on page 691](#)

» To modify the action list of a design time policy

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. To view the Design Time policies:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, perform the following actions:
 - a. Locate **Applicable Scopes**. By default, Policy is selected. Additionally, you can also refine your search further by selecting the following options:
 - Show only Productive Policies

- Include Predefined Policies
- b. Add a Keyword and click the plus button next to the field to add it to the Current Search Criteria recipe.
 - c. You can also further refine your search by selecting values for the following from their respective drop-down list and clicking the plus sign next to these fields to add them to the search criteria:

Note:

The moment you click the plus sign, the result is refreshed in the Search Results page.

- Applicable Organizations
 - Applicable Types
 - Applicable Event Types
- d. This displays a list of defined Design Time policies in the Search Results page.
3. Click the policy whose action list you want to modify.

This opens the Design Time Policy Details page.

4. If the policy is active, deactivate it.

You cannot change the action list of an active policy.

5. Select the **Policy Actions** profile to display the list of actions associated with the policy.

6. You can perform the following operations when modifying the action list for a policy:

- Add - To add a policy action to the current set of actions, click  next to a policy action in the **Policy Actions** menu.
- Delete - To delete a currently set policy action, click  next to the policy action in the policy action details page.
- Sort - To define the order in which you want CentraSite to execute the set policy actions, use the Up and Down arrows next to .

Remember these points when you modify the actions list:

- This dialog only displays actions that support the current scope of the policy. If you need to specify actions for object or event types that are outside of the current scope, you must modify the scope of the policy first (on the **Scope** profile) and then update the action list.
- Note that actions from the WS-I category cannot be combined with other types of actions. Also, when you add a WS-I action to the actions list, CentraSite automatically adds dependent actions to the list as necessary.

7. Click **Save**.
8. When you are ready to put the policy into effect, **Activate** the policy.

Configuring Policy Action Parameters

Pre-requisites:

You can configure the input parameters for a design time policy action by using the Design Time Policy Details page in CentraSite.

To configure the action parameters for a design time policy in CentraSite, ensure that you have the required permissions. For more information, see [“Viewing or Modifying a Design Time Policy” on page 691](#).

Most policy actions have input parameters that you must set to configure the enforcement behavior.

➤ To configure action parameters of a design time policy

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. To view the Design Time policies:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, perform the following actions:
 - a. Locate **Applicable Scopes**. By default, Policy is selected. Additionally, you can also refine your search further by selecting the following options:
 - Show only Productive Policies
 - Include Predefined Policies
 - b. Add a Keyword and click the plus button next to the field to add it to the Current Search Criteria recipe.
 - c. You can also further refine your search by selecting values for the following from their respective drop-down list and clicking the plus sign next to these fields to add them to the search criteria:

Note:

The moment you click the plus sign, the result is refreshed in the Search Results page.

- Applicable Organizations
- Applicable Types
- Applicable Event Types

- d. This displays a list of defined Design Time policies in the Search Results page.
3. Click the policy whose actions you want to configure.

This opens the Design Time Policy Details page.

4. If the policy is active, deactivate it.

Note:

You cannot configure the action parameters of an active policy.

5. In the **Policy Actions** profile, select the policy action you want to modify and set the parameters as required.
6. Click **Save**.
7. When you are ready to put the policy into effect, **Activate** the policy.

Setting Permissions through Design Time Policy Details

Pre-requisites:

To set instance-level permissions on a design time policy in CentraSite, ensure you have the required permissions.

- To set permissions on a organization-specific policy, you must belong to a role that has the Manage Design Time Policies for the organization to which the policy belongs or have the Full instance-level permission on the policy itself.
- To set permissions on a system-wide policy, you must belong to a role that has the Manage System-Wide Design Time Policies or have the Full instance-level permission on the policy itself.

Important:

If you belong to a role that includes the Manage System-Wide Design Time Policies permission, you have the ability to modify permissions of CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be modified except under the direction of a technical representative from Software AG.

By default, all users have View permissions on the design time policies in the registry.

Users who belong to a role that includes the Manage Design Time Policies permission for an organization have Full permission on the policies that belong to the organization. Users who belong to a role that includes the Manage System-Wide Design Time Policies permission, have Full permission on all system-wide policies. To enable other users to modify and delete policies, you must modify the policy's instance-level permission settings.

You can modify the instance-level permissions for a policy by executing a design time policy or by specifying the permissions manually on the **Permissions** action.

When setting permissions on policies, remember these points:

- You can assign permissions to any individual user or group defined in CentraSite.

Note:

If you give the permission to a user to view, modify or delete a policy, and you want that user to be able to perform these operations using CentraSite Control, ensure that the user belongs to a role that also has the Use the Policy UI permission.

- The groups to which you can assign permissions include the following system-defined groups:

Group Name	Description
Users	All users within a specified organization.
Members	All users within a specified organization and its child organizations.
Everyone	All users of CentraSite that includes guest users (if your CentraSite permits access by guests).

- If a user is affected by multiple permission assignments, the user receives the union of all the assignments. For example, if group ABC has `Modify` permission on a policy and group XYZ has `Full` permission on the same policy, users that belong to both groups will, in effect, receive `Full` permission on the policy.

➤ **To assign instance-level permissions to a design time policy**

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. To view the Design Time policies:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, perform the following actions:
 - a. Locate **Applicable Scopes**. By default, `Policy` is selected. Additionally, you can also refine your search further by selecting the following options:
 - Show only Productive Policies
 - Include Predefined Policies
 - b. Add a Keyword and click the plus button next to the field to add it to the Current Search Criteria recipe.
 - c. You can also further refine your search by selecting values for the following from their respective drop-down list and clicking the plus sign next to these fields to add them to the search criteria:

Note:

The moment you click the plus sign, the result is refreshed in the Search Results page.

- Applicable Organizations
- Applicable Types
- Applicable Event Types

- d. This displays a list of defined Design Time policies in the Search Results page.
3. Select the policy whose permissions you want to modify.

This opens the Design Time Policy Details page.

4. If the policy is active, deactivate it.

Note:

You cannot modify the permission settings of an active policy.

5. From the top menu bar, select **Permission** action button.

The Assign Permissions window opens.

6. To add users or groups to the **Users / Groups** list, in the Assign Permissions window, you can perform either of the following actions:
 - a. Use the Type ahead functionality by typing a few letters in the **Add User or Group** to bring up the possible matches for a user or group and click the plus button to select the desired match.
 - b. Click **Choose** in the **Add User or Group** field. In the **Choose Users and Groups** window, click the search icon to select the user or groups to whom you want to assign permissions and click **Ok**.
7. Once you have selected the user or group, assign specific permissions to each user or group in the **User and Group Permissions** section by selecting the checkbox for the user or group. The options are as follows:

Permission	Allows the selected user or group to...
	View the policy.
	View and edit the policy.
	View, edit, and delete the policy. This permission also allows the selected user or group to assign instance-level permissions to the policy.

8. To remove a user or group from the **User and Group Permissions** list, click  adjacent to the group name or user ID.
9. Click **Save**.
10. When you are ready to put the policy into effect, **Activate** the policy.

Setting Permissions through Design Time Policy

You can include the Set Permissions action in a design time policy to set instance-level permissions on a policy. You can use this action to automatically assign permissions to a policy during any of the following events:

- Post-Create
- Pre-State Change
- Post-State Change
- OnTrigger

Performing a Run Now Action

You can perform a Run Now action on a policy that is in **Productive** state. Perform a Run Now action executes all the policy actions associated with a policy.

> To perform a Run Now action on a Design Time policy

Note:

The Run Now button is enabled only if the Event Type is set to On-Trigger.

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. To view the Design Time policies:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, perform the following actions:
 - a. Locate **Applicable Scopes**. By default, Policy is selected. Additionally, you can also refine your search further by selecting the following options:
 - Show only Productive Policies
 - Include Predefined Policies
 - b. Add a Keyword and click the plus button next to the field to add it to the Current Search Criteria recipe.

- c. You can also further refine your search by selecting values for the following from their respective drop-down list and clicking the plus sign next to these fields to add them to the search criteria:

Note:

The moment you click the plus sign, the result is refreshed in the Search Results page.

- Applicable Organizations
- Applicable Types
- Applicable Event Types

- d. This displays a list of defined Design Time policies in the Search Results page.

3. Open the policy for which you want to perform the Run Now action and click .

In the Run Policy Log window, you will see details which includes the following:

- Policy Action Name
- Policy Message
- Policy Object Name
- Status

4. Click **Ok** to close the window.

Deleting Design Time Policies

Pre-requisites:

To delete a design time policy in CentraSite, you must have one of the following permissions:

- To delete a organization-specific policy, you must have the Manage Design/Change-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to delete that policy unless you have permission to manage design/change-time policies for organization ABC.
- To delete a system-wide policy, you must have the Manage System-Wide Design/Change-Time Policies permission.

Important:

If you belong to a role that includes the Manage System-Wide Design/Change-Time Policies permission, you have the ability to delete CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be deleted except under the direction of a technical representative from Software AG.

You delete a policy to remove it from CentraSite permanently.

CentraSite is installed with a system-wide policy called *Check State Validation Policy for Policy*. As per this policy, you cannot delete any policy that is in the **Productive** state. For other states which includes New, Suspended, and Retired states, you can delete the policy.

In addition to being in the New, Suspended, or Retired state, the following conditions must also be met in order to delete a policy:

- The policy must not be in-progress.
- The policy must be inactive.
- You must have Full permission on the policy.

> To delete design time policies

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. To view the Design Time policies:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, perform the following actions:
 - a. Locate **Applicable Scopes**. By default, Policy is selected. Additionally, you can also refine your search further by selecting the following options:
 - Show only Productive Policies
 - Include Predefined Policies
 - b. Add a Keyword and click the plus button next to the field to add it to the Current Search Criteria recipe.
 - c. You can also further refine your search by selecting values for the following from their respective drop-down list and clicking the plus sign next to these fields to add them to the search criteria:

Note:
The moment you click the plus sign, the result is refreshed in the Search Results page.

 - Applicable Organizations
 - Applicable Types
 - Applicable Event Types
 - d. This displays a list of defined Design Time policies in the Search Results page.
3. You can delete a policy in either of the following ways:

a. From the Search Results page - select one or more policies by selecting the checkbox next to the policy name(s) you want to delete and click 

b. From the Policy Details page - open the policy you want to delete and click  in the policy details page.

4. If the policy is active, deactivate it before you delete the policy.

Note:

You cannot delete an active policy.

5. When you are prompted to confirm the delete operation, click **OK**.

Note:

When you delete a policy that is an intermediate version, CentraSite also deletes all previous versions of the policy.

Important:

If you have selected several policies where one or more of them are predefined policies. For example, collector and handler policies. You can use the **Delete** button to delete the policies. However, as you are not allowed to delete predefined policies, only policies you have permission for are deleted. The same applies to any other policies for which you do not have the required permissions.

Versioning a Design Time Policy

Pre-requisites:

To create a new version of a design time policy in CentraSite, you must have one of the following permissions:

- To create a new version of a organization-specific policy, you must have the Manage Design Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to create a new version of that policy unless you have permission to manage design time policies for organization ABC. This is because the versioned policy has the same organizational scope as the original.
- To create a new version of a system-wide policy, you must have the Manage System-Wide Design Time Policies permission.

When you need to make changes to an existing policy, creating a new version of the policy is an efficient way to accomplish this task. Versioning a policy enables you to create a new version of a policy (this is an identical copy of the existing policy) and make your changes to the new version. When you are ready to put the updated policy into effect, activate the new version of the policy.

When you activate the new version, CentraSite automatically deactivates and retires the old version of the policy.

When you create a new version of a policy:

- You can only create a new version from the *latest version* of a policy. For example, if a policy already has versions 1.0, 2.0, and 3.0, CentraSite only allows you to create a new version of the policy from version 3.0. It makes no difference whether the policy that you are versioning is active or inactive. You can version a policy in either mode.
- When CentraSite creates a new version of a policy, it produces a version that is identical to the previous version, except that:
 - The new policy's system-assigned version identifier is incremented by one.
 - Ownership of the new policy is assigned to the user who created the new version.
- Like all new policies, the new version begins its lifecycle in the New state and is marked as inactive.
- CentraSite automatically establishes a relationship between the new version of the policy and the previous version. This relationship enables several capabilities and features in CentraSite that relate to versioned policies.

> To version a design time policy

1. In the CentraSite Business UI activity bar, click **Design Time**.
2. To view the Design Time policies:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, perform the following actions:
 - a. Locate **Applicable Scopes**. By default, Policy is selected. Additionally, you can also refine your search further by selecting the following options:
 - Show only Productive Policies
 - Include Predefined Policies
 - b. Add a Keyword and click the plus button next to the field to add it to the Current Search Criteria recipe.
 - c. You can also further refine your search by selecting values for the following from their respective drop-down list and clicking the plus sign next to these fields to add them to the search criteria:

Note:

The moment you click the plus sign, the result is refreshed in the Search Results page.

- Applicable Organizations
 - Applicable Types
 - Applicable Event Types
- d. This displays a list of defined Design Time policies in the Search Results page.
3. Locate the *most recent version* of the policy for which you want to create a new version.
This opens the Design Time Policy Details page.
 4. Modify the new version of the policy as necessary and save it.
 5. When you are ready to put the new version into effect, activate the new policy. CentraSite deactivates and retires the previous version.

System-Assigned and User-Assigned Version Identifiers

CentraSite maintains two version identifiers for a policy: a *system-assigned identifier* and a *user-assigned identifier*.

- The system-assigned identifier is a version number that CentraSite maintains for its own internal use. CentraSite automatically assigns this identifier to a policy when the policy is created. You cannot delete it or modify it. A system-assigned identifier for a policy is numeric and always has the format *MajorVersion.Revision*. A policy always begins with a system-assigned version identifier of 1.0. The *MajorVersion* number is incremented by one each time you create a new version of a policy (for example, 1.0, 2.0, 3.0).

A system-assigned version number of a policy is shown in the **System Version** column on the Design Time Policies page and in the **System Version** field of the policy detail page. However, if you manually provide a version number when creating a design time policy, then that value takes precedence over the system assigned version. You will see this user entered value displayed in the **System Version** column.

- The user-assigned identifier is an optional identifier that you can assign to a distinguish a specific version of a policy. This identifier does not need to be numeric. For example, you might use a value such as V2.a (beta) to identify a version.

Managing Design and Change-Time Policies through CentraSite Control

This section describes operations you can perform to manage the design and change-time policies through CentraSite Control.

Creating a Design/Change-Time Policy

Pre-requisites:

To create a new design/change-time policy, you must have one of the following permissions in CentraSite:

- To create policies for a specific organization, you must have the Manage Design/Change-Time Policies permission for that organization. By default, users in the CentraSite Administrator, Organization Administrator, or Policy Administrator role have this permission.
- To create system-wide policies (that is, policies that apply to all organization within an instance of CentraSite), you must have the Manage System-Wide Design/Change-Time Policies permission. By default, users in the CentraSite Administrator role and Operations Administrator role have this permission.

➤ **To create a design/change-time policy**

1. In CentraSite Control, go to **Policies > Design/Change-Time**.

This displays a list of defined design-time and change-time policies in the Design/Change-Time Policies page.

2. Click **Add Policy**.

3. In the **Policy Information** panel, specify the following fields:

In this field...	Do the following...
Name	Type a name for the new policy. A policy name can contain any character (including spaces). A policy name does not need to be unique within the registry. However, to reduce ambiguity, you should avoid giving multiple policies the same name. As a best practice, Software AG recommends that you adopt appropriate naming conventions to ensure that policies are distinctly named within your organization.
Description	(Optional). Type a description for the new policy. This description appears when a user displays a list of policies in the user interface.
Version	(Optional). Specify a version identifier for the new policy.

Note:
The version identifier does not need to be numeric.

Examples:

```
0.0a
1.0.0 (beta)
Pre-release 001
V1-2007.04.30
```

The version identifier you type here is the policy's public, user-assigned version identifier. CentraSite also maintains an internal, system-assigned version number for the policy.

In this field... Do the following...

- Priority** Type an integer that represents the priority of this policy with respect to other policies that might be triggered by the same event. The priority value determines the order in which the policies are enforced. The lower the **Priority** value, the higher the priority (that is, 0 is the highest priority and policies with this priority value are executed first).
- Priority values 0 through 10 and values greater than 9999 are reserved for predefined policies. You cannot assign these values to the user-defined policies that you create in CentraSite Control.
 - The default priority for a user-defined policy is 11.

4. In the **Scope** panel, specify the object and event types to which the policy applies.

In this field... Do the following...

Object Types Specify the types of objects to which this policy applies.

Note:

If the object that you select is a base type that has virtual types associated with it, and the **Inherit Base Type Policies** option is enabled for certain of its virtual types, be aware that the policy you create is applied to instances of the base type *and* instances of those virtual types.

Event Types Specify the types of events to which this policy applies.

Important:

The OnCollect, OnMove and OnExport events are designed to execute handler and collector processes. *Do not* use these events unless you are creating a handler or collector policy. The improper use of these event types can damage the registry.

Organization Specify the organization to which the policy applies or select ALL if the policy applies to all organizations.

Note:

The **Organization** list contains the names of all organizations for which you have Manage Design/Change-Time Policies permission. The option appears in the list if you also have Manage System-Wide Design/Change-Time Policies permission.

5. Click **Next**.
6. If you selected the PreStateChange or PostStateChange event in the previous panel and there is a lifecycle model for the object types that you have specified, CentraSite Control asks you to select the lifecycle states that triggers the policy. To complete this panel, do the following:

- a. If you want the policy to execute immediately *before* the state is actually changed, click **Add States** in the **Before the Object Enters State** list and select the states that causes this policy to execute. (The **Before the Object Enters State** list is present if you selected the PreStateChange event in the previous panel.)
 - b. If you want the policy to execute immediately *after* the state is changed, click the **Add States** button in the **After the Object Entered State** list and select the states that causes this policy to execute. (The **After the Object Entered State** is present if you selected the PostStateChange event in the previous panel.)
 - c. Click **Next**.
7. From the **Available Actions** list, select the actions that you want CentraSite to execute when it applies this policy. Keep the following points in mind when you select the actions for the policy:
- The actions shown in the **Available Actions** list are determined by the object types and event types that you specified on the **Scope** panel. If you do not see an action that you need, that action is probably not compatible with all of the object types and event types that you selected in the **Scope** panel.
 - If necessary, you can click **Previous** to return to the **Scope** panel and change your object-type and event-type selections.
 - Ensure that the actions in the **Selected Actions** list appear in the order that you want CentraSite to execute them at enforcement time. If necessary, use the controls above the list to place the actions in the correct order.
 - Be aware that actions from the WS-I category cannot be combined with other types of actions. Also be aware that when you add a WS-I action to the action list, CentraSite automatically adds dependent actions to the list as necessary.
8. Click **Finish** to save the new (as yet incomplete) policy.
9. Complete the new policy by doing the following:
- a. Configure the parameters for each action on the **Actions** tab.
 - b. (Optional). Specify additional selection criteria to narrow the set of objects to which this policy applies.
 - c. If the policy is to be enforced during a PreStateChange or PostStateChange event, ensure that the options on the **States** tab specify the lifecycles and state changes to which the policy applies.
 - d. If you want to allow other users to view, edit, or delete this policy, click the **Permissions** tab and assign permissions to those users.
10. When you are ready to put the policy into effect, activate the policy.

Activating a Design/Change-Time Policy

Pre-requisites:

To activate a design/change-time policy in CentraSite, you must have one of the following permissions:

- To activate a organization-specific policy, you must have the Manage Design/Change-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to activate that policy unless you have permission to manage design/change-time policies for organization ABC.
- To activate a system-wide policy, you must have the Manage System-Wide Design/Change-Time Policies permission.

Important:

If you belong to a role that includes the Manage System-Wide Design/Change-Time Policies permission, you have the ability to activate CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be deactivated except under the direction of a technical representative from Software AG.

CentraSite does not begin enforcing a Design/Change Time policy until you *activate* it.

Policies are managed by a predefined lifecycle installed with CentraSite. This lifecycle, called the Policy Lifecycle, defines the following lifecycle states: New, Productive, Suspended, and Retired.

To activate a policy, you change the policy's lifecycle state to the Productive state. This state change executes CentraSite's *Automatic Policy Activation* policy.

Note:

The *Automatic Policy Activation* policy is a hidden system policy. You cannot edit or delete this policy.

When you activate a policy, be aware that:

- You are not be allowed to activate the policy unless all of its parameters have been set. When you switch the policy to the Productive state, CentraSite executes the *Validate Policy Activation* policy. This policy does not allow you to switch a policy to the Productive state if the policy's parameters have not yet been set.
- Some organizations require an approval to activate a policy. If your organization has an approval action associated with the activation of a policy, CentraSite does not activate the policy until the required approvals are obtained.
- If an earlier version of the policy is already active, CentraSite deactivates the old version before it activates the new one.
- When a policy becomes active, CentraSite begins enforcing it immediately. You can suspend enforcement of a policy by switching it to the Suspended state.
- To activate a policy, you must have permission to change the policy to the Productive state.

To determine whether a policy is active or inactive, examine the policy's **Active** indicator on the **Policies > Design/Change-Time** page. The icon in the **Active** column indicates the policy's activation state as follows:

Icon	Description
	Policy is active.
	Policy is inactive.

The activation state of a policy is also reported next to the **State** field in the Design/Change-Time Policy Details page.

➤ To activate a design/change-time policy

1. In CentraSite Control, go to **Policies > Design/Change-Time**.

This displays a list of defined design-time and change-time policies in the Design/Change-Time Policies page.

2. Locate the policy you want to activate and select **Details** from its context menu.

This opens the Design/Change-Time Policy Details page.

3. Examine the **Actions** tabs and verify that all of the actions on this tab display the green icon in the **Parameters Set** column. If any of the actions display the red circle icon in this column, set their parameters before you continue.
4. In the **Policy Information** panel, click **Change State**.
5. In the **Change Lifecycle State** dialog box, select the **Productive** lifecycle state and click **OK**.
6. Examine the **State** field in the **Policy Information** panel to verify that the policy's state has been changed.

If this state change requires approval, the policy's decoration indicates that the policy is in the pending mode. CentraSite automatically switches the policy to the requested state and activate the policy after all the necessary approvals have been obtained.

Note:

While the policy is in pending mode, it cannot be edited.

Deactivating a Design/Change-Time Policy

Pre-requisites:

To deactivate a design/change-time policy in CentraSite, you must have one of the following permissions:

- To deactivate a organization-specific policy, you must have the Manage Design/Change-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to deactivate that policy unless you have permission to manage design/change-time policies for organization ABC.
- To deactivate a system-wide policy, you must have the Manage System-Wide Design/Change-Time Policies permission.

Important:

If you belong to a role that includes the Manage System-Wide Design/Change-Time Policies permission, you have the ability to deactivate CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be deactivated except under the direction of a technical representative from Software AG.

Deactivating a design/change-time policy causes CentraSite to suppress enforcement of the policy. You usually deactivate a policy for the following reasons:

- To suspend enforcement of a particular policy (temporarily or permanently).
- To edit a policy (for example, to modify the scope of a policy or change its action list).

To deactivate a policy, you change the policy to the Suspended state. Switching the policy to this state triggers the *Automatic Policy Deactivation* policy, which deactivates the policy. Switching the policy to the Retired state also deactivates the policy, but you do not want to switch a policy to this state unless you intend to deactivate it permanently. After you place a policy in the Retired state, you cannot reactivate it.

When you deactivate a policy, CentraSite does not deactivate a policy if it is in the process of being executed. If you attempt to deactivate a policy while it is executing, your state change request fails. If this occurs, wait for a period time and then try to deactivate the policy again.

➤ **To deactivate a design/change-time policy**

1. In CentraSite Control, go to **Policies > Design/Change-Time**.

This displays a list of defined design-time and change-time policies in the Design/Change-Time Policies page.

2. Locate the policy you want to deactivate and select **Details** from its context menu.

This opens the Design/Change-Time Policy Details page.

3. In the **Policy Information** panel, click **Change State**. (If you do not see the **Change State** button, it is probably because you do not have permission to change the lifecycle state of a policy.)
4. In the **Change Lifecycle State** dialog box, select the **Suspended** state (to deactivate it temporarily) or the **Retired** state (to deactivate it permanently), then click **OK**

5. Examine the **State** field in the **Policy Information** panel to verify that the policy's state has been changed.

If this state change requires approval, the **State** field indicates that the policy is in the pending mode. CentraSite automatically switch the policy to the requested state (and deactivate the policy) after all the necessary approvals have been obtained.

Viewing Design/Change-Time Policy List

The Design/Change-Time Policies page displays the design/change-time policies in CentraSite. This list displays policies for all organizations, not just your own. It also includes system-wide policies.

By default, the policy list shows only the user-defined policies in the registry. If you want to view the internal predefined policies, you must enable the **Show Predefined Policies** option. This option displays both user-defined and predefined policies.

You can sort the list by object type or event type. To specify the sorting order, select either **Object type** or **Event type** from the drop-down list labeled **Browse by**.

Be aware that a policy might appear multiple times in the list. For example, if you create a policy that applies to both Assets and Report Templates, the policy appears under the **Assets** heading and the **Report Templates** heading when you view the list by object type.

> To view the design/change-time policy list

1. In CentraSite Control, go to **Policies > Design/Change Time**.

This displays a list of defined design-time and change-time policies in the Design/Change-Time Policies page.

- To filter the list to see just a subset of the available policies, type a partial string in the **Search** field. CentraSite applies the filter to the **Name** column. The **Search** field is a type-ahead field, so as soon as you enter any characters, the display is updated to show only those policies whose name contains the specified characters. The wild card character % is supported.
- To see predefined policies as well as user-defined policies, enable the **Show Predefined Policies** option.

The Design/Change-Time Policies page provides the following information about each policy:

Note:

Only the first six columns described below are displayed in this list by default. You can select to display the additional columns using the **Select Columns** button.

Column	Description
Name	The name assigned to the policy.

Column	Description						
Description	Additional comments or descriptive information about the policy.						
Type	The object type(s) to which the policy applies.						
Event	The event(s) that triggers the policy.						
Organization	The organization to which the policy applies.						
	<table border="1"> <thead> <tr> <th>This value...</th> <th>Indicates that...</th> </tr> </thead> <tbody> <tr> <td>All</td> <td>The policy is system-wide and applies to all organizations.</td> </tr> <tr> <td><i>OrgName</i></td> <td>The policy applies to the specified organization.</td> </tr> </tbody> </table>	This value...	Indicates that...	All	The policy is system-wide and applies to all organizations.	<i>OrgName</i>	The policy applies to the specified organization.
This value...	Indicates that...						
All	The policy is system-wide and applies to all organizations.						
<i>OrgName</i>	The policy applies to the specified organization.						
Active	The policy's current enforcement state.						
	<table border="1"> <thead> <tr> <th>Icon</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>The policy is active. CentraSite enforces this policy when events within the scope of the policy occur.</td> </tr> <tr> <td></td> <td>The policy is inactive. Inactive policies exist in the registry, but they are not enforced.</td> </tr> </tbody> </table>	Icon	Description		The policy is active. CentraSite enforces this policy when events within the scope of the policy occur.		The policy is inactive. Inactive policies exist in the registry, but they are not enforced.
Icon	Description						
	The policy is active. CentraSite enforces this policy when events within the scope of the policy occur.						
	The policy is inactive. Inactive policies exist in the registry, but they are not enforced.						
Priority	The priority value assigned to the policy.						
Owner	The user to which the policy belongs.						
System Version	The automatically generated system-assigned version identifier for the policy.						
Version	The user-assigned version identifier for the policy.						
State	The policy's current lifecycle state.						

Viewing or Modifying a Design/Change-Time Policy

Pre-requisites:

To examine and modify the properties of a design/change-time policy in CentraSite, you must have one of the following permissions:

- To examine and modify the properties of a organization-specific policy, you must have the Manage Design/Change-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to examine and modify the properties of that policy unless you have permission to manage design/change-time policies for organization ABC.
- To examine and modify the properties of a system-wide policy, you must have the Manage System-Wide Design/Change-Time Policies permission.

Important:

If you belong to a role that includes the Manage System-Wide Design/Change-Time Policies permission, you have the ability to modify CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be modified except under the direction of a technical representative from Software AG.

You cannot modify a policy while it is in the Productive state. To make changes to a policy, you can do any of the following:

- Create a new version of the policy, make the necessary changes to the new version and switch the new version to the Productive state when you are ready to put it into effect. Switching the new version to the Productive state will immediately put the previous version in the Retired state. (The Retired state is an end state. After you place a policy in this state, you can no longer reactivate it.)
- Create a completely new policy that includes the required changes. When you are ready to put the new policy into effect, switch the old policy to the Suspended state and switch the new policy to the Productive state. When you are certain that you will no longer need to revert to the original policy, switch it to the Retired state.
- Switch the existing policy to the Suspended state, make the necessary changes to the policy and then switch it back to the Productive state. While the policy is in the Suspended state, it will not be enforced. (Because suspending the policy results in an enforcement gap, one usually does not use this approach in a production environment.)

➤ To examine and modify details of a design/change-time policy

1. In CentraSite Control, go to **Policies > Design/Change-Time**.

This displays a list of defined design-time and change-time policies in the Design/Change-Time Policies page.

2. Locate the policy whose details you want to examine and modify, and select **Details** from its context menu.

This opens the Design/Change-Time Policy Details page.

3. If the policy is active, deactivate it.

You cannot modify the details of an active policy.

4. On the Design/Change-Time Policy Details page, examine and modify the attributes as required.

Field	Description
Name	The name of the policy. A policy name can contain any character (including spaces). A policy name does not need to be unique within the registry. However, to reduce ambiguity, you should avoid giving multiple policies the same name.

Field	Description
Description	<p>As a best practice, we recommend that organizations adopt appropriate naming conventions to ensure the assignment of distinct policy names.</p> <p>(Optional). Additional comments or descriptive information about the policy.</p>
Version	<p>The user-assigned version ID assigned to this policy. You may use any versioning scheme you select for identifying different versions of a policy. The identifier does not need to be numeric. Examples:</p> <p style="margin-left: 40px;">0.0a</p> <p style="margin-left: 40px;">1.0.0 (beta)</p> <p style="margin-left: 40px;">Pre-release 001</p> <p style="margin-left: 40px;">V1-2007.04.30</p> <p>CentraSite also maintains a system-assigned version identifier for a policy. The system-assigned version identifier is independent from the version identifier that you specify in this field.</p>
Priority	<p>An integer that represents the priority of this policy with respect to other policies that might be triggered by the same event.</p> <ul style="list-style-type: none"> ■ Priority values 0 through 10 and values greater than 9999 are reserved for predefined policies. You cannot assign these values to the user-defined policies that you create in CentraSite Control. ■ The default priority for a user-defined policy is 11.
Actions	<p>The settings on this tab specify the actions that CentraSite executes when the policy is enforced.</p>
Scope	<p>The settings on this tab specify the object types and event types to which the policy applies.</p>
States	<p>The settings on this tab specify the lifecycles and state changes to which this policy applies.</p> <p>Note: The States tab is present only if the policy's scope includes a PreStateChange or PostStateChange event.</p>
Permissions	<p>The settings on this tab identify the users who have instance-level permissions on the policy.</p>

5. Click **Save**.

Note:

If the selected object and event types are not compatible with the current set of actions in the action list, CentraSite does not permit you to save the policy. You must correct the policy's action list or its scope to save the policy successfully.

- When you are ready to put the policy into effect, activate the policy.

Scope of a Design/Change-Time Policy

Scope refers to the set of properties that determine when a policy is enforced. For a design/change-time policy, scope is determined by the policy's **Object Types**, **Event Types** and **Organization** properties, which are described below.

Property	Description
----------	-------------

Object Types	<p>The list of object types to which this policy applies.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: If the object that you select is a base type that has virtual types associated with it and the Inherit Base Type Policies option is enabled for certain of its virtual types, be aware that the policy you create is applied to instances of the base type <i>and</i> instances of those virtual types.</p> </div> <p>You can optionally restrict a policy to specific instances of the selected object types by specifying additional object-selection criteria.</p>
---------------------	---

Event Types	<p>The list of event types to which the policy applies.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: Not all event types occur for all object types.</p> </div> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Important: The OnCollect, OnMove, and OnExport events are designed to execute handler and collector processes. <i>Do not</i> use these events unless you are creating a handler or collector policy. The improper use of these event types can damage the registry.</p> </div>
--------------------	---

Organization Determines whether the policy belongs to a specific organization or is system-wide.

System-wide and Organization-specific Policy Enforcement

The **Organization** property specifies the organization to which the policy applies. When the **Organization** property is set to ALL, it indicates that the policy is *system-wide*. When the **Organization** property specifies a particular organization, it indicates that the policy is *organization-specific*.

Organization-specific Policies

An organization-specific policy is enforced on objects that belong to the same organization as the organization to which the policy applies. For example, if you have a policy that executes when

user objects are updated and its **Organization** property specifies organization ABC, CentraSite only execute that policy when user objects *in organization ABC* are updated.

Points to keep in mind when working with organization-specific policies:

- You can create organization-specific policies for any organization on which you have Manage Design/Change-Time Policies permission. For example, if you have Manage Design/Change-Time Policies permission for organization ABC and XYZ, you can create organization-specific policies for either organization.
- At enforcement time, CentraSite selects policies based on the organization to which the object belongs, *not* the organization to which the requestor belongs. For example, if a user from organization XYZ edits an asset in organization ABC, CentraSite applies organization ABC's policies (the organization to which the asset belongs), not organization XYZ's policies (the organization to which the requestor belongs).

System-wide Policies

A system-wide policy is enforced for all organizations. For example, if you create a system-wide policy that executes when an asset is created, CentraSite enforces the policy whenever *any* user in *any* organization adds an asset to the catalog.

To create a system-wide policy, you must belong to a role that has Manage System-Wide Design/Change-Time Policies permission. In a standard CentraSite configuration, only users in the CentraSite Administrator role and the Policy Administrator role have this permission.

System-wide policies are useful for managing many types of objects. For example, they are often used to assign users to certain server-wide groups or to enforce server-wide naming conventions on objects. However, organization-specific policies are often better choices for asset-related policies, because they enable an organization to tailor its policies to its own development processes and methodologies.

Modifying Scope of a Design/Change-Time Policy

Pre-requisites:

To modify the scope of a design/change-time policy in CentraSite, you must have one of the following permissions:

- To modify the scope of a organization-specific policy, you must have the Manage Design/Change-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to modify the scope of that policy unless you have permission to manage design/change-time policies for organization ABC.
- To modify the scope of a system-wide policy, you must have the Manage System-Wide Design/Change-Time Policies permission.

Important:

If you belong to a role that includes the Manage System-Wide Design/Change-Time Policies permission, you have the ability to modify the scope of CentraSite's predefined policies. However,

you should not do this. These policies perform critical functions within the registry and must not be modified except under the direction of a technical representative from Software AG.

You use the **Scope** tab on the Design/Change-Time Policy Details page to specify a policy's scope.

Scope changes are limited by the set of actions currently selected on the **Actions** tab. That is, CentraSite does not allow you to save a policy if its scope includes object types and events that are not compatible with the current set of specified actions. In some cases, you might need to clear actions from the **Actions** tab in order to select the object types and event types you need on the **Scope** tab.

> To modify the scope of a design/change-time policy

1. In CentraSite Control, go to **Policies > Design/Change Time**.

This displays a list of defined design-time and change-time policies in the Design/Change-Time Policies page.

2. Locate the policy whose scope you want to modify, and select **Details** from its context menu.

This opens the Design/Change-Time Policy Details page.

3. If the policy is active, deactivate it.

You cannot modify the scope of an active policy.

4. Select the **Scope** tab and specify the following:

- In the **Object Types** and **Event Types** lists, select the object types and event types to which the policy applies.
- *Optional.* In the **Apply policy to objects that meet the following criteria**, specify additional selection criteria to narrow the set of objects to which this policy is applied.

5. Click **Save**.

Note:

If the selected object and event types are not compatible with the current set of actions in the action list, CentraSite does not permit you to save the policy. You must correct the policy's action list or its scope to save the policy successfully.

6. When you are ready to put the policy into effect, activate the policy.

Refining the Object Scope

Pre-requisites:

To refine the object scope of a design/change-time policy you must have the instance-level View permission for that particular policy. By default, all users have View permissions on the design/change-time policies in the registry.

To further restrict the set of objects to which the policy is applied, you can specify additional selection criteria in the **Apply policy to objects that meet the following criteria** section of the **Scope** tab. Using this section, you can filter objects by Name, Description, and Classification attributes.

Filtering By Name and Description

You can filter objects based on their Name and Description attributes using any of the following comparison operators:

Comparison Operator	Description
Equals	Selects objects whose Name or Description value matches a given string of characters. For example, you would use this operator if you wanted to apply a policy only to Taxonomy objects with the Description value Project IDs.
Not Equals	Selects objects whose Name or Description value <i>does not match</i> a given string of characters. For example, you would use this operator if you wanted to apply a policy to all Taxonomies <i>except</i> those with the Description value Project IDs.
Contains	Selects objects whose Name or Description property includes a given string of characters anywhere within the property's value. For example, you would use this operator if you wanted to apply a policy to Application Server objects that had the word Fairfax anywhere in their Description property.
Starts With	Selects objects whose Name or Description property begins with a given string. For example, you would use this operator if you wanted to apply a policy only to web services whose name begins with the characters UTIL-.

When specifying match strings for the comparison operators described above, keep the following points in mind:

- Match strings *are not* case-sensitive. If you define a filter for names that start with ABC, it will select names starting abc and Abc (and other variations) as well as ABC.
- Wild card characters are not supported. That is, you cannot use characters such as * or % to represent any sequence of characters. These characters, if present in the match string, are simply treated as literal characters that are to be matched.

Filtering By Classification Attribute

You can also filter objects based on the way in which they are classified. When you filter objects in this way, CentraSite applies the policy to objects that have at least one classification whose value matches a specified taxonomy category. For example, you could use a classification filter to apply a policy to those Application Servers objects that are classified as JBoss servers.

When you filter objects by classification, CentraSite inspects all of an object's classifications at enforcement time. If any of those attributes contain the exact category specified by the selection criteria, the policy is executed.

Note:

To satisfy the selection criteria, the attribute value in the object must match the category specified in the selection criteria *exactly*. Sub-categories of the specified category *are not* considered to be matches. For example, say you have a taxonomy category called Project ABC, and that category has the subcategories Project ABC Design, Project ABC Development, and Project ABC Deployment. If you filter for category Project ABC, CentraSite will apply the policy to objects that are classified by the specific category Project ABC but not objects that are classified by that category's sub-categories.

> To refine the object scope of a design/change-time policy

1. In CentraSite Control, go to **Policies > Design/Change Time**.

This displays a list of defined design-time and change-time policies in the Design/Change-Time Policies page.

2. Locate the policy whose object scope you want to refine, and select **Details** from its context menu.

This opens the Design/Change-Time Policy Details page.

3. Select the policy's **Scope** tab.
4. To filter by Name or Description, perform the following in the **Apply policy to objects that meet the following criteria** section:
 - a. Select **Name** or **Description**.
 - b. Select the comparison operator.
 - c. Specify the match string.
5. To filter by object classification, take the following steps in the **Apply policy to objects that meet the following criteria** section of the tab:
 - a. Select **Classification**.
 - b. Click **Browse** and select the category by which you want to filter objects.
6. To specify additional criteria, click the plus button and repeat steps 2 and 3.

Important:

If you specify multiple filters, the policy is applied only if the object matches *all the selection criteria* (that is, the selection criteria is combined using an AND operator, not an OR).

Configuration of Policies that Execute on Lifecycle State Changes

If you create a policy that executes on the PreStateChange or PostStateChange event type, you must configure the policy's **States** tab. The settings on this tab identify the specific state changes that triggers the policy. This tab also specifies whether the policy is to be executed before or after the object is switched to a specified state.

When creating policies that execute on state changes, keep the following points in mind:

- Policies that are triggered by a state change execute when an object *switches to* a specified state (called the *target state*). The object's state prior to the change is immaterial. For example, if you have a lifecycle model with the states: *Test*, *Production*, and *Offline* and you have a policy that specifies the *Offline* target state, that policy executes anytime the object switches to the *Offline* state. It does not matter whether the transition occurs from the *Test* state or the *Production* state.
- Policies that are triggered by a state change are executed regardless of whether the state change is initiated from the CentraSite Control UI, the API (for example, a custom client program) or another policy.
- You cannot specify a target state on the **States** tab unless that state has already been defined in a lifecycle model. Additionally, the lifecycle model must be active. In other words, you cannot completely configure a policy that executes on a state change until you have created and activated the lifecycle model whose state(s) triggers the policy.
- If you configure the policy to execute before the object's state is changed (that is, on a PreStateChange event) and any action in the policy fails, the state change does not occur.

Assigning Actions to a Design/Change-Time Policy

The **Actions** tab on the Design/Change-Time Policy Details page specifies the list of actions that you want CentraSite to execute when it enforces the policy. CentraSite executes actions in the order in which they appear in the list.

The action list can include any built-in or custom actions that are compatible with the policy's scope (as currently specified on the policy's **Scope** tab).

Policy Scope and Action Scope

Like a policy, an action has a declared scope. The scope of an action is declared in the **Object Types** and **Event Types** properties in the action's *action template*. An action template is an object that defines a policy action that is available within CentraSite.

A policy can only include actions whose scope *matches or exceeds* the policy's own scope. For example, if you had an action ABC with the following scope:

Action ABC's Scope

Object Type(s):	Service
Event Type(s):	Post-Create Post-Update

You could use this action in policies 1 and 2 below, because these policies include only objects and events that are encompassed by scope of the action. However, you could not use the action in policies 3 or 4, because these policies include objects and events that the action does not support.

Compatible with Action ABC? Policy #1 Scope

Object Types(s):	Service	Yes
Event Type(s):	Post-Create	Yes

Compatible with Action ABC? Policy #2 Scope

Object Types(s):	Service	Yes
Event Type(s):	Post-Create Post-Update	Yes

Policy #3**Compatible with Action ABC? Policy Scope**

Object Types(s):	Service Report Template (<i>out of scope</i>)	No
Event Type(s):	Post-Create	Yes

Policy #4**Compatible? Policy Scope**

Object Types(s):	Service	Yes
Event Type(s):	Post-Create Post-Update Post-Delete (<i>out of scope</i>)	No

> To examine the scope of an action

1. In CentraSite Control, go to **Policies > Action Templates**.

This displays a list of defined design-time and change-time actions in the Action Templates page.

2. Locate the action whose scope you want to examine, and select **Details** from its context menu.

This opens the Edit Action Template page.

3. Select the action's **Scope** tab.

Note:

Virtual types and base types are treated as distinct object types with respect to policy action scope. A policy action that is scoped for a particular virtual type cannot be inserted into a policy that is scoped specifically for the base type.

Modifying the Action List

Pre-requisites:

To modify the action list of a design/change-time policy in CentraSite, you must have one of the following permissions:

- To modify the action list of a organization-specific policy, you must have the Manage Design/Change-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to modify the action list of that policy unless you have permission to manage design/change-time policies for organization ABC.
- To modify the action list of a system-wide policy, you must have the Manage System-Wide Design/Change-Time Policies permission.

Important:

If you belong to a role that includes the Manage System-Wide Design/Change-Time Policies permission, you have the ability to modify the action list of CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be modified except under the direction of a technical representative from Software AG.

➤ To modify the action list of a design/change-time policy

1. In CentraSite Control, go to **Policies > Design/Change Time**.

This displays a list of defined design-time and change-time policies in the Design/Change-Time page.

2. Locate the policy whose action list you want to modify and select **Details** from its context menu.

This opens the Design/Change-Time Policy Details page.

3. If the policy is active, deactivate it.

You cannot change the action list of an active policy.

4. Select the **Actions** tab to display the list of actions associated with the policy.

5. To add actions to, delete actions from or modify the order of actions in the list, do the following:
 - a. Click **Edit Actions List**.
 - b. Use the controls in the **Edit Assigned Actions** dialog box.

When editing the list of actions:

- This dialog only displays actions that support the policy's current scope. If you need to specify actions for object or event types that are outside of the current scope, you must modify the policy's scope first (on the **Scope** tab) and then update the action list.
 - Make sure the actions in the **Assigned Actions** list appear in the order that you want CentraSite to execute them.
 - Be aware that actions from the WS-I category cannot be combined with other types of actions. Also be aware that when you add a WS-I action to the action list, CentraSite automatically adds dependent actions to the list as necessary.
- c. Click **OK**.
 6. Click **Save**.
 7. When you are ready to put the policy into effect, activate the policy.

Configuring Policy Action Parameters

Pre-requisites:

You can configure the input parameters for a design/change-time policy action by using the Design/Change-Time Policy Details page in CentraSite Control.

To configure the action parameters for a design/change-time policy in CentraSite, you must have one of the following permissions:

- To configure the action parameters for an organization-specific policy, you must have the Manage Design/Change-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to export that policy unless you have permission to manage design/change-time policies for organization ABC.
- To configure the action parameters for a system-wide policy, you must have the Manage System-Wide Design/Change-Time Policies permission.

Important:

If you belong to a role that includes the Manage System-Wide Design/Change-Time Policies permission, you have the ability to modify the action parameters of CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be modified except under the direction of a technical representative from Software AG.

Most policy actions have input parameters that you must set to configure the action's enforcement behavior.

When you display the **Actions** tab on the Design/Change-Time Policy Details page, the icon in the **Parameters Set** column indicates whether the action has input parameters that need to be set.

This Icon... Indicates that...



The action has required input parameters that have not yet been set.



All of the action's required input parameters have been set.

Note:

This icon automatically appears for actions that have no input parameters.

➤ **To configure action parameters of a design/change-time policy**

1. In CentraSite Control, go to **Policies > Design/Change Time**.

This displays a list of defined design-time and change-time policies in the Design/Change-Time Policies page.

2. Locate the policy whose actions you want to configure and select **Details** from its context menu.

This opens the Design/Change-Time Policy Details page.

3. If the policy is active, deactivate it.

You cannot configure the action parameters of an active policy.

4. On the **Actions** tab, do the following for each action in the list:

- a. Click the action whose parameters you want to examine or set.
- b. In the Edit Action Parameters page, set the parameters as necessary.

Note:

Required parameters are marked with an asterisk.

- c. Click **Save** to save the parameter settings.
5. Click **Save**.
 6. When you are ready to put the policy into effect, activate the policy.

Setting Permissions through Design/Change-Time Policy Details

Pre-requisites:

To set instance-level permissions on a design/change-time policy in CentraSite, you must have one of the following permissions:

- To set permissions on a organization-specific policy, you must belong to a role that has the Manage Design/Change-Time Policies for the organization to which the policy belongs or have the Full instance-level permission on the policy itself.
- To set permissions on a system-wide policy, you must belong to a role that has the Manage System-Wide Design/Change-Time Policies or have the Full instance-level permission on the policy itself.

Important:

If you belong to a role that includes the Manage System-Wide Design/Change-Time Policies permission, you have the ability to modify permissions of CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be modified except under the direction of a technical representative from Software AG.

By default, all users have View permissions on the design/change-time policies in the registry.

Users who belong to a role that includes the Manage Design/Change-Time Policies permission for an organization have Full permission on the policies that belong to the organization. Users who belong to a role that includes the Manage System-Wide Design/Change-Time Policies permission, have Full permission on all system-wide policies. To enable other users to modify and delete policies, you must modify the policy's instance-level permission settings.

You can modify the instance-level permissions for a policy by executing a design/change-time policy or by specifying the permissions manually on the **Permissions** tab in .

When setting permissions on policies, keep the following points in mind:

- You can assign permissions to any individual user or group defined in CentraSite.

Note:

If you give a user permission to view, modify or delete a policy, and you want that user to be able to perform these operations using CentraSite Control, ensure that the user belongs to a role that also has the Use the Policy UI permission.

- The groups to which you can assign permissions include the following system-defined groups:

Group Name	Description
Users	All users within a specified organization.
Members	All users within a specified organization and its child organizations.

Group Name	Description
------------	-------------

Everyone	All users of CentraSite <i>including guest users</i> (if your CentraSite permits access by guests).
-----------------	---

- If a user is affected by multiple permission assignments, the user receive the union of all the assignments. For example, if group ABC has Modify permission on a policy and group XYZ has Full permission on the same policy, users that belong to both groups will, in effect, receive Full permission on the policy.

➤ **To assign instance-level permissions to a design/change-time policy**

1. In CentraSite Control, go to **Policies > Design/Change Time**.

This displays a list of defined design-time and change-time policies in the Design/Change-Time Policies page.

2. Locate the policy whose permissions you want to modify, and select **Details** from its context menu.

This opens the Design/Change-Time Policy Details page.

3. If the policy is active, deactivate it.

You cannot modify the permission settings of an active policy.

4. On the Design/Change-Time Policy Details page, click the **Permissions** tab.

5. To add users or groups to the **Users / Groups** list, do the following:

- a. Click **Add Users / Groups**.

- b. Select the users and groups to which you want to assign permissions.

If you want to filter the list, type a partial string in the **Search** field. CentraSite applies the filter to the **Users/Groups** column.

Examples

String	Description
b	Displays names that contain b
bar	Displays names that contain bar
%	Displays all users and groups

- c. Click **OK**.

- To remove a user or group from the **Users / Groups** list, select the check box beside the group name or user ID and click **Delete**.
- Assign specific permissions to each user and group in the **Users / Groups** list as follows:

Permission **Allows the selected user or group to...**

View View the policy.

Note:

Disabling this permission does not prevent a user from accessing the policy. CentraSite implicitly grants users View permission on all design/change-time policies within an instance of CentraSite. This implicit permission that CentraSite grants to a user cannot be not revoked by disabling the **View** permission on this tab.

Modify View and edit the policy.

Full View, edit, and delete the policy. This permission also allows the selected user or group to assign instance-level permissions to the policy.

- Click **Save**.
- When you are ready to put the policy into effect, activate the policy.

Setting Permissions through Design/Change-Time Policy

You can include the Set Permissions action in a design/change-time policy to set instance-level permissions on a policy. You can use this action to automatically assign permissions to a policy during any of the following events:

- Post-Create
- Pre-State Change
- Post-State Change
- OnTrigger

Policies on Demand

If you create a policy for the OnTrigger event, you can use the **Run** button on the details page to run the policy on demand.

Many of the built-in actions in CentraSite support the OnTrigger event. For example, you can run the WS-I actions on demand. You can also use the OnTrigger event to execute policies that set permissions on certain types of objects or change the state of an object.

You can run a policy on demand if you have view permission on the policy.

Objects to Which the Policy is Applied

When you run a policy on demand, CentraSite queries the objects in the organization to which the policy applies and selects objects that satisfy the following conditions:

- The object is one that is within the policy's object scope.
- The object is one on which you have View permission.
- The object's name, description, and classification properties satisfy the object-selection criteria on the policy's **Scope** tab (if any).

Note:

If the policy's **Organization** property is set to ALL (meaning that it is a system-wide policy), then CentraSite queries *all organizations* for objects that satisfy the conditions listed above.

CentraSite executes the policy's actions on each object in the result set produced by this query (henceforth, referred to as the *target set*).

If an action in the policy performs an update or delete operation on the objects in the target set, be aware that these operations execute successfully if you have the appropriate Modify or Full permission on the target object. If you do not have the required permissions, the action that performs the edit or delete operation fails and the failure is reported in the policy log.

As with other policies, a policy that you execute on demand might trigger other policies. This occurs anytime an action in the policy performs an operation that is within the scope of another policy.

When you run a policy on demand, CentraSite executes the policy against each object in the target set. Results are written to the policy log and are also displayed in the results window in the user interface.

For example, if you have a policy that contains actions 1, 2, and 3 and the target set contains objects A, B, and C, the policy iterates over the objects in the target set as follows:

Iteration 1: Execute actions 1, 2, and 3 on object A

Iteration 2: Execute actions 1, 2, and 3 on object B

Iteration 3: Execute actions 1, 2, and 3 on object C

If an action returns a failure code during an iteration of the policy, CentraSite writes the failure message to the policy log and immediately exits that iteration of the policy. If the target set contains additional objects, CentraSite applies the policy to the next object in the target set.

There is one exception, namely if the Send Email Notification action returns a failure code; in this case, CentraSite writes the failure message to the policy log and performs the next action in the policy (if one exists).

Running a Policy on Demand

> To run a design/change-time policy on demand

1. In CentraSite Control, go to **Policies > Design/Change Time**.
2. Locate the policy whose details you want to view or modify and select **Details** from its context menu.

This opens the Design/Change-Time Policy Details page.

3. Examine the **Scope** tab and verify that the **Object Types** property and the criteria in the **Apply policy to objects that meet the following criteria** section of the tab (if any) identify the precise set of objects to which you want the policy applied.
4. Examine the **Actions** tab and verify that the action list contains the set of actions that you want CentraSite to execute and that the parameters for all actions in the list are set properly.
5. Click **Run**.

If the **Run** button does not appear on the Design/Change-Time Policy Details page, it is most likely because:

- The policy has not been activated.
 - The policy's Event Type property does not include the OnTrigger event.
6. When the policy completes, examine the results window to determine whether all iterations of the policy executed successfully. CentraSite writes these results to the policy log, so you can view them later.

Deleting Design/Change-Time Policies

Pre-requisites:

To delete a design/change-time policy in CentraSite, you must have one of the following permissions:

- To delete a organization-specific policy, you must have the Manage Design/Change-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to delete that policy unless you have permission to manage design/change-time policies for organization ABC.
- To delete a system-wide policy, you must have the Manage System-Wide Design/Change-Time Policies permission.

Important:

If you belong to a role that includes the Manage System-Wide Design/Change-Time Policies permission, you have the ability to delete CentraSite's predefined policies. However, you should

not do this. These policies perform critical functions within the registry and must not be deleted except under the direction of a technical representative from Software AG.

You delete a policy to remove it from CentraSite permanently.

CentraSite is installed with a system-wide policy called *Check State Validation Policy for Policy*. This policy does not allow you to delete a policy unless the policy is in the New or Retired state.

In addition to being in the New or Retired state, the following conditions must also be met in order to delete a policy:

- The policy must not be in-progress.
- The policy must be inactive.
- You must have Full permission on the policy.

➤ To delete design/change-time policies

1. In CentraSite Control, go to **Policies > Design/Change Time**.

This displays a list of defined design-time and change-time policies in the Design/Change-Time Policies page.

2. If the policy is active, deactivate it.

You cannot delete an active policy.

3. Right-click a policy you want to delete, and click **Delete**.

You can also select multiple policies, click the **Actions** menu, and click **Delete**

4. When you are prompted to confirm the delete operation, click **OK**.

Note:

When you delete a policy that is an intermediate version, CentraSite also deletes all previous versions of the policy.

Important:

If you have selected several policies where one or more of them are predefined policies. For example, collector and handler policies. You can use the **Delete** button to delete the policies. However, as you are not allowed to delete predefined policies, only policies you have permission for is deleted. The same applies to any other policies for which you do not have the required permission.

Copying a Design/Change-Time Policy

Pre-requisites:

To create a copy of a design/change-time policy in CentraSite, you must have one of the following permissions:

- To create a copy of a organization-specific policy, you must have the Manage Design/Change-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to create a copy of that policy unless you have permission to manage design/change-time policies for organization ABC. This is because the copied policy has the same organizational scope as the original.
- To create a copy of a system-wide policy, you must have the Manage System-Wide Design/Change-Time Policies permission.

A design/change-time policy can become quite complex, especially if it contains several policy actions. Instead of creating a new policy “from scratch”, it is sometimes easier to copy an existing policy that is similar to the one you need and edit the copy.

CentraSite includes a copy feature that lets you do this. It produces a copy that is identical to the original policy. Unlike a new version of a policy, a copy of a policy is not associated with the original policy in any way CentraSite treats the copy just as if it were a new policy that you created from scratch.

When you create a copy of a policy, be aware that:

- When CentraSite creates a copy of a policy, the new copy of the policy is identical to the original one except that:
 - The new policy's system-assigned version identifier is always set at 1.
 - Ownership of the new policy is assigned to the user who created the copy.
- Like all new policies, the copied policy begins its lifecycle in the New state and it is marked as inactive.
- There is no expressed relationship between the copy and the original policy (that is, CentraSite does not establish any type of association between the two policies).

In general, a copied policy is no different than a newly created policy.

➤ To copy a policy

1. In CentraSite Control, go to **Policies > Design/Change-Time**.
2. Locate the policy that you want to copy and select **Create Copy** from its context menu.
3. Modify the new policy as necessary and save it.

Versioning a Design/Change-Time Policy

Pre-requisites:

To create a new version of a design/change-time policy in CentraSite, you must have one of the following permissions:

- To create a new version of an organization-specific policy, you must have the Manage Design/Change-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to create a new version of that policy unless you have permission to manage design/change-time policies for organization ABC. This is because the versioned policy has the same organizational scope as the original.
- To create a new version of a system-wide policy, you must have the Manage System-Wide Design/Change-Time Policies permission.

When you need to make changes to an existing policy, creating a new version of the policy is an efficient way to accomplish this task. Versioning a policy enables you to create a new version of a policy (which is an identical copy of the existing policy) and make your changes to the new version. When you are ready to put the updated policy into effect, you simply activate the new version of the policy. When you activate the new version, CentraSite automatically deactivates and retires the old version of the policy.

When you create a new version of a policy:

- You can only create a new version from the *latest version* of a policy. For example, if a policy already has versions 1.0, 2.0, and 3.0, CentraSite only allows you to create a new version of the policy from version 3.0. It makes no difference whether the policy that you are versioning is active or inactive. You can version a policy in either mode.
- When CentraSite creates a new version of a policy, it produces a version that is identical to the previous version, except that:
 - The new policy's system-assigned version identifier is incremented by one.
 - Ownership of the new policy is assigned to the user who created the new version.
- Like all new policies, the new version begins its lifecycle in the New state and is marked as inactive.
- CentraSite automatically establishes a relationship between the new version of the policy and the previous version. This relationship enables several capabilities and features in CentraSite that relate to versioned policies.

➤ To version a design/change-time policy

1. In CentraSite Control, go to **Policies > Design/Change-Time**.
2. Locate the *most recent version* of the policy for which you want to create a new version and select **Create New Version** from its context menu.
3. If the policy is active, deactivate it.

You cannot version an active policy.

4. Modify the new version of the policy as necessary and save it.
5. When you are ready to put the new version into effect, activate the new policy. CentraSite deactivates and retires the previous version.

Note:

If you activate the new version of the policy while CentraSite is in the middle of executing the old version, your activation request fails. If this occurs, wait for a period time and then try to activate the new version of the policy again.

System-Assigned and User-Assigned Version Identifiers

CentraSite maintains two version identifiers for a policy: a *system-assigned identifier* and a *user-assigned identifier*.

- The system-assigned identifier is a version number that CentraSite maintains for its own internal use. CentraSite automatically assigns this identifier to a policy when the policy is created. You cannot delete it or modify it. A policy's system-assigned identifier is numeric and always has the format *MajorVersion.Revision*. A policy always begins with a system-assigned version identifier of 1.0. The *MajorVersion* number is incremented by one each time you create a new version of a policy (for example, 1.0, 2.0, 3.0).

A policy's system-assigned version number is shown in the **System Version** column on the Design/Change-Time Policies page and in the **System Version** field of the policy's detail page.

- The user-assigned identifier is an optional identifier that you can assign to a distinguish a specific version of a policy. This identifier does not need to be numeric. For example, you might use a value such as V2.a (beta) to identify a version.

A policy's system-assigned version number is shown in the **Version** column on the Design/Change-Time Policies page and in the **Version** field of the policy's detail page.

Modifying the Predefined Lifecycle Model for Policies

The predefined lifecycle model that CentraSite uses for policies is made up of four states: New, Productive, Suspended, and Retired. This lifecycle is generally adequate for most environments. However, you can make certain minor types of customizations to it if necessary.

For information about the ways in which you can customize this lifecycle model, see the chapter, Lifecycle Management.

Viewing the Policy Log

Pre-requisites:

To view this log, you must belong to a role that includes the View Policy Log permission:

- If you belong to the CentraSite Administrator role, you can view all entries in the policy log.

- If you belong to the Organization Administrator role for an organization, you can view the log entries for policies that were triggered by users in your organization.
- If you do not belong to either of these roles, but you have the View Policy Log permission, you can view the log entries for the policies that you triggered.

The policy log contains information about policy that CentraSite has executed. By default, CentraSite only logs information about policies that fail. However, you can optionally configure CentraSite to log information about policies that resulted in success, info, warning, and failure alerts.

Note:

Over time, the policy log can grow quite large, especially if you are logging information about successful policies. To prevent the policy log from growing too large, you should purge it periodically.

Note:

From your **Inbox** on the My CentraSite page, you can view the list of policies that failed during events that you triggered. You do not need special permissions to view this log.

➤ **To view the policy log**

1. In CentraSite Control, go to **Administration > Logs > Policy Log**.
2. Complete the following fields to specify which type of log entries you want to view:

In this field...	Specify...
Object Name	(Optional). A pattern string that describes the names of the objects (of Object Type) whose log entries you want to view. You can provide the exact name or use a pattern string consisting of a character sequence and the % wild card character (which represents any string of characters). For example, if you specify the pattern string 'A%', CentraSite displays entities whose names start with 'A'. Leave Entity Name empty to view all names.
Policy Type	The type of policy whose log entries you want to view. To view the log entries for design/change-time policies, select Design/Change Time from the drop-down list (if it is not already selected).
Object Type	The object type whose log entries you want to view.
Event Type	The event type whose log entries you want to view.
Policy Status	The policy execution status that you want to view. A policy's execution status is the result set of each of its action's execution result. CentraSite writes the following policy execution status to the policy log depending on the log configuration:

In this field...	Specify...
	Icon Description
	 <i>Success</i> . Displays policies that have resulted in success alert.
	 <i>Info</i> . Displays policies that have resulted in informational alert.
	 <i>Inprogress</i> . Displays policies that have resulted in inprogress alert.
	 <i>Warning</i> . Displays policies that have resulted in warning alert.
	 <i>Failure</i> . Displays policies that have resulted in failure alert.
Execution Date	<i>Optional</i> . The time period that you want to examine. Leave the From and To fields empty to view log entries for all dates.

3. Click **Search** to retrieve the specified log entries.
4. To view details for a particular entry in the returned list, click the name of the policy.

Note:

If a policy included a WS-I action, the log entry for the policy includes a link to the results of the WS-I action.

Viewing the Policy's Action Result

A policy's execution status depends on each of its action's execution result.

When an action in a design/change-time policy completes its execution, it returns a completion code and a completion message. If the completion code indicates success, informational or warning, CentraSite performs the next action in the policy (if one exists) or completes the requested work on the object (for example, it commits the given change to the database) and writes the information to the policy log. The policy log displays a success, informational and warning alert accordingly if configured to log these alerts.

However, if the completion code indicates failure, CentraSite records the error in the policy log. Then it immediately exits the policy. If the policy contains additional actions, those actions are not executed. If the policy was triggered by one of the pre-commit events (for example, during a Pre-Create, Pre-Update or Pre-State Change event) the requested operation is not performed. If the initial request had triggered multiple policies, any policy that had not yet been executed is bypassed.

The policy's execution status is a result set of each of its actions result.

The following table summarizes how a policy's execution status is affected by each of its action's execution result:

The policy's execution status have...	Icon	Policy Action A	Policy Action B	Policy Action C	Policy Action D	Policy Action E
Success		Success	Success	Success	Success	Success
Info		Success	Success	Info	Success	Success
In Progress		Info	In Progress	Success	Success	Success
Warning		Success	Success	Info	Warning	Warning
Failure		Success	Warning	Success	Success	Failure

Note:

If you have used a client jar from the version 8.2.2, then CentraSite shows policy's result status as Success in the client side but however on the server side the policy log continues to store the policy's result status as Informational or Warning. This is because, a client jar used from versions of CentraSite prior to version 8.2.5 do not support the above policy status.

Viewing Failed Policies From Your Inbox

Your Inbox on the My CentraSite page includes the **Failed Policies** link, which displays the list of policies that failed during events that you initiated.

When you click the **Failed Policies** link in your Inbox, CentraSite Control opens a two-pane screen. The upper pane displays the list of logged policy failures that occurred during events that you initiated. The lower pane displays detailed information for a selected failure.

A failure stays in your **Failed Policies** list until you explicitly clear it from the list using the **Remove from List** button or the underlying log entry is purged from the policy log.

Note:

When you clear an entry from this list using the **Remove from List**, you do not remove the entry from the underlying policy log. You simply eliminate it from your Inbox display.

Mark as Read		Remove From List			
<input checked="" type="checkbox"/>	Failed Policy	Date Triggered	Triggered By User	Failed On Action	
<input type="checkbox"/>	SchemaPolicy	2010-09-09 03:06 PM	Admin01	Initiate Approval	
<input type="checkbox"/>	Validate Policy Activation	2010-09-09 02:33 PM	Admin01	Validate Policy Activation	
<input type="checkbox"/>	Validate Policy Activation	2010-09-09 02:10 PM	Admin01	Validate Policy Activation	
<input type="checkbox"/>	Validate Policy Activation	2010-09-09 02:08 PM	Admin01	Validate Policy Activation	
<input checked="" type="checkbox"/>	Check State Validation Policy fr	2010-09-09 02:07 PM	Admin01	Validate State	

Details				
	Actions	Reason	Object	Date Triggered
	Validate State	Policy Approval for XML Schem	Not Available	2010-09-09 02:07 PM

> To view the policy log

1. In CentraSite Control, go to **Home > My CentraSite**.
2. In the **Policy Log** section of the Inbox, click **Failed Policies**.
3. Examine the list of failures in the upper pane of the **Failed Policies** window.
4. If you want to examine the details for a reported failure, click in any non-linked area of the row that contains the failure log entry.

The details for the selected failure appears in the lower pane.

This Icon...	Indicates That...
	Action had resulted in success alert.
	Action had resulted in informational alert.
	Action had resulted in inprogress alert.
	Action had resulted in warning alert.
	Action had resulted in failure alert.

Exporting Design/Change-Time Policies

Pre-requisites:

To export a design/change-time policy in CentraSite, you must have one of the following permissions:

- To export a organization-specific policy, you must have the Manage Design/Change-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to export that policy unless you have permission to manage design/change-time policies for organization ABC.
- To export a system-wide policy, you must have the Manage System-Wide Design/Change-Time Policies permission.

When exporting a design/change-time policy, keep the following points in mind:

- You can export a policy that is active or inactive. You do not need to deactivate a policy to export it.
- The export process does not export the following objects that a policy references:
 - Lifecycle State
 - Organization
- The export process does not export the policy's instance-level permissions. When an administrator imports the policy on the target instance, the import process assigns instance-level permissions.
- If the policy includes a custom action, CentraSite exports the action template for the custom action along with the action's metadata.
- If the policy's object scope includes a custom asset type, CentraSite exports that type along with the policy.
- If the policy scope is **Assets**, then the type information is exported. So any basic attributes referenced in the policies have to be edited in the target registry (for example, if the policy has assertions such as Set Attribute value or Validate Attribute value). There is one exception, namely if the policy uses classification attributes in the policy assertions; in this case, the type information is included in the export.
- The export process exports the parameter values assigned to the actions in the policy. If the parameter value is a reference to an instance of one of the following object types, you must export the referenced object and import it on the target instance of CentraSite before you import the policy:
 - Organization
 - User
 - Group

If referenced User/Group objects do not already exist on the target instance of CentraSite when you import the policy, the import removes the references to these objects and the import is successful. After the import, you can edit the policy to reference the necessary user/group

objects. If a parameter references any other type of object, the export process exports the referenced object with the policy and that object is imported as necessary into the target registry.

- If the policy executes on a `PreStateChange` or `PostStateChange` event, the lifecycle model and state information on the policy's **States** tab is exported. If the specified lifecycle model and states do not exist in the target instance of CentraSite when the policy is imported, the import process fails (that is, the policy is not imported).

➤ To export a design/change-time policy

1. In CentraSite Control, go to **Policies > Design/Change-Time**.
2. Locate the policy that you want to export and select **Export** from its context menu.
3. Specify the options in the **Export** dialog box and click **OK**.
4. Save the archive file when prompted to do so.
5. Examine the export log that is displayed by CentraSite Control and check for any errors that occurred during the export process.

Importing Design/Change-Time Policies

Pre-requisites:

To import a design/change-time policy in CentraSite, you must have one of the following permissions:

- To import an organization-specific policy, you must have the Manage Design/Change-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to import that policy unless you have permission to manage design/change-time policies for organization ABC. If the archive contains a system-wide policy or custom action templates, you must also belong to a role that has the Manage System-Wide Design/Change-Time Policies permission.
- To import a system-wide policy, you must have the Manage System-Wide Design/Change-Time Policies permission.

When importing a design-time policy, keep the following points in mind:

- If the policy, or any related object in the archive already exists in the target instance of CentraSite, the existing object is overwritten.
- When the imported policy *is added* to the registry, the import process assigns the Organization, Lifecycle State, and Activation State properties to the imported policy as follows.

Property	Value assigned to new policy
Organization	The organization of the user that is performing the import, unless the policy is a system-wide policy. If the policy is system-wide, the policy's Organization attribute will remain set to All.
Lifecycle State	The initial state for new policies.
	<p>Note: If you are using the lifecycle model for policies that is installed with CentraSite, the imported policy's state is set to <i>New</i>.</p>
Activation State	Inactive

Additionally, CentraSite assigns instance-level permissions to the imported policy just as though you created the policy manually. (In other words, the imported policy receives the same permission settings as the policies you create from scratch.)

- When an imported policy *replaces* (updates) an existing policy in the target registry, all of the policy's properties, except for its permission settings, are updated according to the policy object in the archive. This includes the policy's organizational scope, its lifecycle state, and its activation state (that is, whether the policy is active or inactive). If the referenced organization and lifecycle model does not already exist on the target registry, the import process fails. Also, be aware that the import process replaces the policy on the target regardless of whether the target policy is currently active or inactive. Due to this behavior, you might want to import only new versions of a policy, and not use the import process to directly replace a version of a policy that already exists.
- If the archive file contains a reference to an object that is not already present in the target registry or is not included in the archive file itself, the policy will not be imported.
- If design/change-time policies exist for the events that the import process initiates (for example, the creation of a policy), those policies is triggered.

➤ To import a design/change-time policy

1. In CentraSite Control, go to **Policies > Design/Change-Time**.
2. Click the **Import** icon.
3. Click **Browse** and select the zip file containing the policy that you want to import.
4. To automatically replace the policy if it already exists, select **Allow replace of existing objects**.
5. Click **OK**.
6. When the import process is complete, check the import log to make sure that the policy and its associated objects were imported successfully.

7. If the import process was successful, open the policy in CentraSite Control and do the following:
 - a. Inspect the parameter settings for each action in the policy to ensure that they are set properly. Specify appropriate values as necessary.
 - b. Inspect the other properties assigned to the policy and ensure that they are set appropriately.
 - c. Activate the policy.

Approval Policies

CentraSite's approval-management framework enables you to create policies that trigger approval processes when certain time events occur in the registry. For example, you might create a policy that requires a system architect to review and approve all assets before they are switched to a productive state.

To impose an approval process on a change time event, you create an *approval policy* for the event. An approval policy is a policy that contains one of CentraSite's built-in *approval actions*.

Note:

- In this guide, the term *approval policy* is used to generally refer to policies that you use to perform approvals. Technically speaking, an approval policy is no different than an ordinary design/change-time policy. It is simply one that includes an approval action. An approval policy can also include other actions (assuming they are within the policy's scope).
- The use of approval policies is not supported if you are using a CentraSite Community Edition license.

The Approval Actions

Action Name	Description
Initiate Approval	This action submits a request to the <i>approval group</i> .
Initiate Group-dependent Approval	This action submits a request to the approval group <i>only</i> if the requestor belongs to a specified user group.

Enforcing Approval Action

When a user performs an operation that triggers an approval policy, CentraSite initiates an approval workflow and submits the user's request to the designated group of approvers. Approvers receive the approval request in their inbox in CentraSite Control. Approvers whose user account includes a valid e-mail address also receive an email message informing them that a request is awaiting their approval. You can configure an approval action to send an email notification to other specified users, too.

Note:

For CentraSite to issue email messages, an administrator must first configure CentraSite's email server settings.

CentraSite does not execute the user's requested operation until it obtains the necessary approvals. If an approver rejects the request, CentraSite notifies the requestor and immediately exits the policy. It does not perform the user's requested operation nor does it execute any remaining actions in the approval policy. If other policies were to be executed against the user's request (that is, if the request triggered lower priority policies in addition to the approval policy) those policies are not executed.

Using the **Inbox** on the My CentraSite page in CentraSite Control, users can view the status of the requests that they have submitted for approval. Approvers also use the **Inbox** to review and authorize requests that require their approval.

Auto-Approval

When the user who submits a request is also an authorized approver for the requested operation, the request is *auto-approved*, the requestor's approval is granted implicitly.

Requests that are handled after it is auto-approved depends on whether the approval workflow is configured to execute in *Anyone* or *Everyone* mode.

- *In Anyone mode*, an auto-approval completes the approval process. Such requests do not formally initiate an approval workflow, however, they do appear in the Approval History log (the log indicates that the request was auto-approved).
- *In Everyone mode*, the requestor's approval is registered and then the request is submitted to the remaining approvers in the approval group.

Note:

The auto-approval process also occurs when an approval action is invoked and all of its specified approver groups are empty or all users in the specified groups are inactive.

Approval Modes

You configure an approval action to operate in one of the following modes:

■ Anyone

In Anyone mode, a request can be approved or rejected by any single user in the approver group. In this mode, only one user in the group is required to approve or reject the request. This is the default mode.

■ Everyone

In Everyone mode, a request must be approved by all users in the approver group (it does not matter in which order the approvals are obtained). A rejection by any approver in the group causes the request to be rejected.

Events and Objects for Approval

You can add approval policies for the following combinations of events and object types:

Event Type	Supported Object Types	Supported Approval Actions
PreStateChange	Asset Policy	<ul style="list-style-type: none">■ Initiate Approval■ Initiate Group-dependent Approval
OnConsumerRegistration	Asset	<ul style="list-style-type: none">■ Initiate Approval■ Initiate Group-dependent Approval

Using the Initiate Approval Action

You use the Initiate Approval action when you want to define an approval process that applies to *all of the users* who submit requests that trigger the policy. If you need to apply the approval process selectively, that is, if only certain groups of users require approval, or if different groups of users require authorization from different groups of approvers, use the Initiate Group-dependent Approval action instead.

The parameters required to define the action include the name of the approval flow that the action initiates, the name of the approver groups (that is, the groups of users who are allowed to approve requests that trigger the policy) and email addresses of users who should be informed of the progress of the action.

Using the Initiate Group-Dependent Approval Action

When you want a policy to initiate an approval process for some groups of requestors and not for others, or when you need to route requests to different approvers based on the user group to which a requestor belongs, you use the Initiate Group-dependent Approval action.

The parameters required to define the action include the name of the approval flow that the action initiates, the name of the approver groups (that is, the groups of users who are allowed to approve requests that trigger the policy), the names of the related triggering groups (that is, the groups of members whose requests require approval), and email addresses of users who should be informed of the progress of the action.

You can route approvals to different approver groups based on the triggering group to which the requestor belongs. For example, you could configure the action to route requests to the approver groups Approvers-A and Approvers-B when a requestor belongs to a particular triggering group.

Points to consider when using the Initiate Group-dependent Approval action:

- If a requestor does not belong to any of the groups specified in the **Triggering Groups** parameter, CentraSite does not even initiate an approval workflow. Approval is waived and CentraSite simply executes the next action in the policy. (Be aware that, because the request does not enter the approval framework, requests that are waived do not appear in the Approval History log.)
- The UI dialog allows you to combine triggering groups and approval groups into sets, where each set defines one or more triggering groups and the associated approver groups. You can

specify multiple sets and CentraSite processes each set in the order given in the dialog. When it encounters a set whose **Triggering Groups** parameter includes a user group to which the requestor belongs, it immediately initiates an approval workflow based on that set and ignores any remaining sets in the dialog. In other words, if the requestor is a member of multiple **Triggering Groups**, approval is determined by whichever of those groups appears first in the dialog.

- If a requestor is a member of both **Triggering Groups** and a member of **Approver Group** in the same triggering group/approver group combination, the request is auto-approved.

Switching the State of an Object when an Approval Request is Rejected

By default, an object's lifecycle state is not changed when an approval request is rejected. For example, let's say that object ABC is in the Tested state and an approval request is submitted to switch object ABC to the Production state. If the approval request is rejected, object ABC stays in the Tested state. For some approval work-flows, however, you might want to switch objects to a particular state when they are rejected. To do this you use the **Reject State** parameter.

Important:

If you use this option, make sure that the lifecycle model provides a transition from the state(s) that an object might be in when the approval policy executes and the state that you specify in the **Reject State** parameter. Otherwise, the approval engine will not be able to switch the target object to the specified state when a rejection occurs.

Also be aware that you can specify only one state in the **Reject State** parameter. Therefore, if an approval policy applies to objects with different lifecycle models, the **Reject State** can apply to only one of those models. For example, let's say you use the same approval policy for both XML schema and services, but these two asset types follow different lifecycle models. If you set the **Reject State** to a state in the lifecycle model for XML schema, only XML schema will switch to this state when an approval request is rejected. Services, when rejected, will simply remain in their current state. If you want to specify one reject state for XML schema and another for services, you must create a separate approval policy for each type.

Adding an Approval Policy

To create an approval policy, you must perform the following general steps:

1. Create a user group composed of the individuals who are authorized to approve the type of request that triggers the policy.
2. Create a design/change-time policy with the appropriate scope (event type and object type) and into this policy, insert an approval action.

Multiple Actions in Approval Policy

An approval policy can include actions in addition to the approval action. For example, you might create a policy like the example shown below, which validates a particular attribute in the asset and executes a custom action before it initiates the approval process.

Example

```
Validate Attribute Value  
MyCustomAction  
Initiate Approval
```

The example above illustrates how you can execute policy actions before you initiate the approval process. You can also insert actions after the approval action as long as those actions do not attempt to modify the object on which the policy is acting. When an object enters an approval process, CentraSite locks the object to prevent any modifications to the object while it is undergoing approval. The object remains locked until the approval policy *and all additional policies that are triggered by the same event* are complete.

If an approval policy includes an action that attempts to update the object after approval process has been initiated, that action fails. When this occurs, CentraSite immediately exits the policy and reverts the object to its previous state.

The following shows an approval policy that includes an action after the approval action. This policy executes successfully, because the action following the approval action simply sends out an email notification. It does not attempt to modify the asset on which the policy is acting:

Example A (correct)

```
Validate Classification  
Set Instance and Profile Permissions  
Initiate Approval  
Send Email Notification
```

The following shows an approval policy that would not execute successfully. In this example, the Set Instance and Profile Permissions action follows the Initiate Approval action. Because the asset is locked at this point in the policy, the Set Instance and Profile Permissions action fails and the asset reverts to its previous lifecycle state:

Example B (incorrect)

```
Validate Classification  
Initiate Approval  
Set Instance and Profile Permissions  
Send Email Notification
```

Tip:

As a best practice, avoid executing any additional actions after the approval action in an approval policy. If there are actions that you need to execute after approval is granted, place those actions in a separate policy that executes on the PostStateChange event.

Using Approvals with Pre-State Change Events

The Pre-State Change event occurs when you change the lifecycle state of an object.

You can use an approval policy with the Pre-State Change event to prevent users from switching the following types of objects to certain lifecycle states (for example, to the Productive state) without first getting the required approvals:

- Policy
- Asset
- Lifecycle model

➤ To use approvals with Pre-State Change events

1. Ensure that the state change(s) that triggers the policy are defined in an existing lifecycle model. If the lifecycle model, with the appropriate state, has not yet been defined, you must create it before you create the approval policy.
2. Create a design/change-time policy with the following scope:
 - **Event Type:** Pre-State Change
 - **Object Type:** Policy, Asset, or Lifecycle Model
3. In the **Before the Object Enters State** section of the policy's **States** tab, specify the state change that requires approval.
4. On the policy's **Actions** tab, specify and configure the approval action that is to be executed when an in-scope object switches to the state specified in the preceding step. If other actions are to be executed before or after the approval action, insert those actions on the **Action** tab.

CAUTION:

Only certain kinds of actions can be included *after* the approval action in an approval policy. Some actions, if they occur after the approval action, will cause the policy to fail.

Using Approvals with OnConsumerRegistration Events

The OnConsumerRegistration event occurs when the user submits the consumer registration request.

The enhanced feature enables users to register users and applications as consumers of an asset without explicitly creating a consumer-registration policy and without requiring the owner of the asset to review and accept the registration request.

If you want to impose an approval process, that is, you want designated individuals to review and approve the registration request, you create a design-time policy with one of the CentraSite's built-in approval actions for the OnConsumerRegistration event. At a minimum, this policy must include the Register Consumer action, because this action performs the work of actually registering a consumer (that is, it establishes the actual relationship between the asset and the specified consumers). It can optionally include other actions, such as Set Consumer Permission action, as needed.

> To use approvals with OnConsumerRegistration events

1. Create a design/change-time policy with the following scope:
 - **Event Type:** OnConsumerRegistration
 - **Object Type:** Asset (of any type)
2. On the policy's **Actions** tab, add the following actions. Ensure that the approval action *precedes* the Register Consumer action:
 - Initiate Approval —OR— Initiate Group-dependent Approval
 - Register Consumer
3. Configure the approval action's input parameters.
4. Insert additional actions before and after this pair of actions as necessary. The following example shows an action list that obtains the required approval, executes the registration process, and then grants instance-level permissions to the consumers that the policy registers:

```
Initiate Approval  
Register Consumer  
Set Consumer Permission
```

Approver Groups

An approver group is a user group that identifies the set of individual who are authorized to approve a submitted request. An approver group can be composed of users from any organization.

Note:

If you want approvers to be able to review the details for an object that they are asked to approve, make sure those users have View permission on the object. For example, if the users in group ABC is required to approve assets that are switched to a certain lifecycle state, make sure that the users in group ABC have View permission on the assets that they are asked to approve. Without View permission, approvers will not be able to examine the details of the assets that users submit to them for approval.

Changing the membership of an approver group *does not* affect requests that are already pending approval. When CentraSite submits a request to the approval engine, it assigns the users from the specified approver group to that request. The request retains its assigned set of approvers throughout the entire approval process.

For example, let's say that approval policy P1 uses approver group AG1 and that AG1 contains users A and B. If a user submits a request that triggers P1, users A and B become the designated approvers for that request. Let's say that while this request is waiting for approval, an administrator modifies group AG1 and replaces users A and B with users X and Y. This change have *no effect* on the request that is awaiting approval. Users A and B continues to its designated approvers. The changes to group AG1 will only affect new requests that policy P1 submits for approval.

Reviewing Requests Submitted for Approval

In the Approval History log, CentraSite maintains a record of every request that users submit for approval.

Note:

The list displays *all* requests that have been submitted on your behalf, including requests that were auto-approved.

➤ To view requests you have submitted for approval

1. In CentraSite Control, go to **Home > My CentraSite**.

This displays the My CentraSite page.

2. Click **Menu > Inbox > Approvals > Approval Requests**.

This displays the list of requests that you have submitted for approval.

The **Status** column in the **Approval Requests** list indicates the state of each request as follows:

Status	Description
Pending	The request has been submitted for approval, but has not yet been processed by the required approvers.
Approved	The request has been submitted and approved by the required approvers.
Rejected	The request has been submitted and rejected. The operation you requested was not executed.
Auto-Approved	The request was auto-approved. This occurs when you submit a request for which you are also an authorized approver.

3. To examine the details for a particular request (including a list of the individuals who are authorized to approve the request), click any non-linked area in the row that contains the request. CentraSite Control

The details for the request will appear in the **Approval Flow Information** panel.

Approving a Request

If you are an approver, CentraSite places requests in your **Pending Approvals** inbox for your review and approval.

➤ To view and approve requests in your inbox

1. In CentraSite Control, go to **Home > My CentraSite** to display the My CentraSite page.

2. Click **Menu > Inbox > Approvals > Pending Approvals** to display the list of requests that require your approval.
3. Select the request that you want to review by clicking any non-linked area within the row that contains the request.

Note:

If you want to examine the object on which the approval is requested, click the object name in the **Approval Request** column. If you have View permissions on the object, you are allowed to view the object's details.

4. In the **Comment** text box, type a comment. For example, *Request rejected. Add required specifications to this asset and resubmit.*
5. Click **Accept** or **Reject** as appropriate to approve or reject the request.

Viewing Your Approval History

The **Approval History** link in your inbox displays all requests for which you were an authorized approver (that is, the list includes any request whose approver group included you as a member).

> To view your approval history

1. In CentraSite Control, go to **Home > My CentraSite**.

This displays the My CentraSite page.

2. Click **Menu > Inbox > Approvals > Approval History**.

This displays the list of requests for which you were an authorized approver.

3. To examine the details for a particular request, including the list of other authorized approvers, select the approval workflow in the **Approval Request** column.

Viewing the Approval History Log

Use the following procedure to display the Approval History log. This log contains a record of all approval requests that have been submitted to CentraSite. To view the Approval History log, you must belong to a role that has the `View Approval History` permission. If you belong to the CentraSite Administrator role, you will see all entries in the Approval History log. Otherwise, you will see only the set of approval requests that were triggered within your organization.

> To view the Approval History log

- In CentraSite Control, go to **Administration > Logs > Approval History**.

Reverting the State of an Object that is Pending Approval

Occasionally, you might need to revert a request that has been submitted for approval. For example, if a request that has already been submitted to the approval engine requires the approval of a user who has left the company, you need to take back that request out of the approval engine and resubmit it after updating the approver group, of course.

When you have an approval request that is stuck in the pending mode, a user in the CentraSite Administrator role can use the following procedure to revert the object to its previous state so that the condition can be corrected and the object can be resubmitted for approval.

Note:

Reverting the lifecycle state of an asset does not undo any attribute changes that might have been made by policies that were executed by the original state-change event. It simply returns the asset's lifecycle property to its previous state. If other attribute changes occurred during the state-change event, you need to undo those changes manually.

> To revert the state of an object that is pending approval

1. In CentraSite Control, use one of the following steps to display the list that contains the object whose pending state you want to revert.

To revert the state of this type of object... Do this in CentraSite Control...

Asset	Go to Asset Catalog > Browse .
Design/Change-Time Policy	Go to Policies > Design/Change-Time .
Run-Time Policy	Go to Policies > Run-Time .
Lifecycle Model	Go to Administration > Lifecycles > Models .

2. Locate the object whose state you want to revert and select **Revert Pending State** from its context menu.

Note:

For assets, you can also perform the **Revert Pending State** command from the **Actions** menu on the asset detail page.

Using the Approval Service API

CentraSite provides a web service that enables you to create applications and integrate third-party workflow tools with CentraSite's approval queue. This service provides operations that enable you to obtain the list of pending requests for an approver and approve or reject those requests through a web service. The web service also provides operations for viewing the approval history log.

Managing Design-Time and Change-Time Policies through Command Line Interface

This section describes operations you can perform to manage design-time and change-time policies through CentraSite Command Line Interface.

Viewing the Action Categories List

Pre-requisites:

To view the list of available action categories through CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `list Action Categories` for this purpose.

> To display the list of action categories

- Run the command `list Action Categories`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list Action Categories [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list Action Categories -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
```

The response to this command could be:

```
Executing the command : list Action Categories
Name : Custom Action Category
Key : uddi:137738d6-c66b-11e4-8896-da7def414f92
Type : Design/Change-Time
Successfully executed the command : list Action Categories
```

Adding Custom Action Category

Pre-requisites:

To add a custom action category through CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `set Action Category` for this purpose.

➤ To add a custom action category

- Run the command `set Action Category`.

The syntax is of the format `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Action Category [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -actioncategory <ACTION-CATEGORY> [-policytype <POLICY-TYPE>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
ACTION-CATEGORY	Name of the action category. If the category name contains white spaces, enclose the name with <code>"</code> .
POLICY-TYPE	(Optional). The type of policy definition. Supported values are <code>- Design/Change-Time, Run-Time</code> . If a value is not specified, CentraSite uses the default value <code>Design/Change-Time</code> .

Important:

Once set, you cannot change the value of this parameter.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Action Category -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-actioncategory "Custom Action Category" -policytype Design/Change-Time
```

The response to this command could be:

```
Executing the command : set Action Category
```

```
Action Category named Custom Action Category is created successfully  
Successfully executed the command : set Action Category
```

Modifying Action Category Details

Pre-requisites:

To modify an existing action category through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `set Action Category` for this purpose.

You provide the key that identifies the category to be updated as an input to the command. The action category key can be obtained by using the `list Action Categories` command.

Note:

You cannot change the policy definition type. But you can rename an action category.

➤ To modify an existing action category

- Run the command `set Action Category`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Action Category [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -actioncategory <ACTION-CATEGORY> [-id <ID>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
ACTION-CATEGORY	The new name of the action category that you want to rename. If the new category name contains white spaces, enclose the name with <code>"</code> .
ID	The ID of the action category you want to rename. You can specify the UDDI key of the category using an optional prefix <code>uddi:</code> . For example: <code>uddi:137738d6-c66b-11e4-8896-da7def414f92</code>

Examples (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Action Category -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-actioncategory "New Custom Action Category" -id
uddi:137738d6-c66b-11e4-8896-da7def414f92
```

The response to this command could be:

```
Executing the command : set Action Category
Action Category named Custom Action Category is updated successfully
Successfully executed the command : set Action Category
```

Deleting Action Category

Pre-requisites:

To delete an existing action category through CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `delete Action Category` for this purpose.

When deleting an action category, keep the following points in mind:

- Deleting a category deletes all the action templates that are classified in that category.
- If the category contains an action template that is currently used by one or more active policies, then the action category cannot be deleted.

➤ To delete an existing action category

- Run the command `delete Action Category`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd delete Action Category [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -id <ID>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
ID	The ID of the action category you want to delete. You can specify the UDDI key of the category using an optional prefix <code>uddi:.</code>

Parameter	Description
	For example: uddi:137738d6-c66b-11e4-8896-da7def414f92
	Examples (all in one line): C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd delete Action Category -url http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage -id uddi:137738d6-c66b-11e4-8896-da7def414f92
	The response to this command could be: Executing the command : delete Action Category Action Category named Custom Action Category is deleted successfully Successfully executed the command : delete Action Category

Viewing the Action Templates List

Pre-requisites:

To view the list of available action templates through CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `list Action Templates` for this purpose.

➤ To display the list of available action templates

- Run the command `list Action Templates`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list Action Templates [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> [-actioncategory <ACTION-CATEGORY>] [-policytype <POLICY-TYPE>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
ACTION-CATEGORY	(Optional). Name of the action category to display a list of action templates underneath it. If the category name contains white spaces, enclose the name with <code>" "</code> .

Parameter	Description
POLICY-TYPE	(Optional). The type of policy definition to display a list of action templates classified with it. Supported values are - Design/Change-Time, Run-Time. If a value is not specified, CentraSite will list both the design-time and run-time action templates.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list Action Templates -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
```

Providing action category name:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list Action Templates -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-actioncategory "Custom Action Category"
```

Providing policy definition type:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list Action Templates -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-policytype Design/Change-Time
```

The response to this command could be:

```
Executing the command : list Action Templates
Name                : Custom Action Template
Key                 : uddi:95f79c00-52a9-11e4-969c-d398081308d4
Action Category    : Custom Action Category
Action Category Type : Design/Change-Time
Successfully executed the command : list Action Templates
```

Adding Custom Action Template

Pre-requisites:

To add a custom action template through CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `add Action Template` for this purpose.

➤ To add a custom action template

- Run the command `add Action Template`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd add Action Template [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -file <CONFIG-FILE> -implementationzip <IMPLEMENTATION-ZIP>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
CONFIG-FILE	Name of the action template configuration file that contains the input parameters.

Note:

If you are saving the file in a different location other than the default location `<CentraSiteInstall_Directory>/utilities`, provide the absolute file path.

IMPLEMENTATION-ZIP	Name of the Java implementation zip file used to construct the action template.
--------------------	---

CentraSite has a sample implementation zip file `uniquenamechecker.zip` that you can use to enforce unique asset names in the CentraSite registry. This sample implementation zip file is located in `<CentraSiteInstall_Directory>/demos/CustomActions/Java`.

Note:

If you have saved the zip file to a different location other than the default location `<CentraSiteInstall_Directory>/utilities`, provide the absolute path to the zip file.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd add Action Template -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-file c:\temp\Custom_Action_Template_Config.xml -implementationzip
c:\temp\custom_action_implementation.zip
```

The response to this command could be:

```
Executing the command : add Action Template
Action Template named Custom Action Template is created successfully
Successfully executed the command : add Action Template
```

Action Template Configuration File

You can configure the details for new custom action template in an XML configuration file. Here is a sample configuration file.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<actionTemplate>
  <!-- type of the Action Category -->
  <actioncategory>CustomActionCategory</actioncategory>
  <!-- name of the Action Template -->
  <templateName>TestActionTemplate</templateName>
  <!-- Description of the Action Template -->
  <templateDescription>This is a new action template
    for design time policy</templateDescription>
  <!-- Scope Object Types of the Action Template -->
  <templateObjectTypes>
    <objectType>Application</objectType>
  </templateObjectTypes>
  <!-- Scope Event Types of the Action Template-->
  <templateEventTypes>
    <eventType>PreCreate</eventType>
    <eventType>PreUpdate</eventType>
  </templateEventTypes>

  <parameterTemplates>

  <parameterTemplate>
    <!-- Name of the parameter-->
    <name>ValidateName</name>
    <!-- Type of the parameter-->
    <type>Object</type>
    <!-- Default values of the parameter-->
    <defaultValue>user1</defaultValue>
    <possibleValues>user2</possibleValues>
    <possibleValues>user3</possibleValues>
    <possibleValues>user4</possibleValues>
    <!-- Array value of the parameter-->
    <array>true</array>
    <!-- Is parameter Required-->
    <isRequired>true</isRequired>
  </parameterTemplate>

  <parameterTemplate>
    <!-- Name of the parameter-->
    <name>ValidateName</name>
    <!-- Type of the parameter-->
    <type>String</type>
    <!-- Default values of the parameter-->
    <defaultValue>user1</defaultValue>
    <possibleValues>user2</possibleValues>
    <possibleValues>user3</possibleValues>
    <possibleValues>user4</possibleValues>
    <!-- Array value of the parameter-->
    <array>true</array>
    <!-- Is parameter Required-->
    <isRequired>true</isRequired>
  </parameterTemplate>
</parameterTemplates>
</parameterTemplate>

  <parameterTemplate>
    <!-- Name of the parameter-->
    <name>ValidateName</name>
    <!-- Type of the parameter-->
    <type>String</type>

```

```
<!-- Default values of the parameter-->
<defaultValue>user1</defaultValue>
<possibleValues>user2</possibleValues>
<possibleValues>user3</possibleValues>
<possibleValues>user4</possibleValues>
<!-- Array value of the parameter-->
<array>true</array>
<!-- Is parameter Required-->
<isRequired>true</isRequired>
</parameterTemplate>

</parameterTemplates>
</actionTemplate>
```

The contents of the configuration file are listed below:

Tag Name	Description
actioncategory	(Mandatory). Name of the action category.
templateName	(Mandatory). Name of the action template.
templateDescription	(Optional). Description of the action template.
templateObjectTypes	(Mandatory). The type of objects to which this action applies. For a list of the object types that CentraSite supports, see “Classification of Types” on page 200 .
templateEventTypes	(Mandatory). The type of events to which this action applies. Supported values are - PreCreate, PostCreate, PreUpdate, PostUpdate, PreDelete, PostDelete, PreState Change, PostState Change, OnConsumerRegistration, OnTrigger, OnCollect, OnExport, OnMove. For details on the individual event types, see “Introduction to Design and Change-Time Policies” on page 668 .
parameterTemplates	(Optional). The parameter template with one or more parameters that serve as input to the action.
name	(Mandatory). Name of the parameter that is input to the action at enforcement.
type	(Mandatory). The data type of the parameter. For a list of the data types that CentraSite supports, see “Basic Components of Type” on page 182 .
defaultValue	(Optional). The default value for the parameter.

Tag Name	Description
possibleValues	(Optional). One or multiple possible values for the parameter.
array	(Optional). Indicates whether a parameter's data type can be an array of values. A value of <code>true</code> allows to pass the parameter's data type as an array during the enforcement of action. Supported values are - <code>true</code> , <code>false</code> .
isRequired	(Optional). Indicates whether the parameter is mandatory or optional. A value of <code>true</code> indicates that the parameter is required for the action enforcement. If the value is <code>false</code> , the parameter is optional.

Modifying Action Template Details

Pre-requisites:

To modify the details of an action template through CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `update Action Template` for this purpose.

The action template you want to modify is specified by the parameter `ID`. Any change of the template's property value must be provided using an updated configuration file. All property values provided by the configuration file, including the specified object types, event types, and the parameter templates, are overwritten.

➤ To modify the details of an action template

- Run the command `update Action Template`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd update Action Template [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -id <ID> [-file <CONFIG-FILE>] [-implementationzip <IMPLEMENTATION-ZIP>] [-forceedit <FORCE-EDIT>]`

The input parameters are:

Parameter	Description
<code>CENTRASITE-URL</code>	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .

Parameter	Description
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, Administrator.
PASSWORD	The password for the registered CentraSite user identified by the parameter USER-ID.
ID	<p>The ID of the action template you want to update. You can specify the UDDI key of the template using an optional prefix <code>uddi:</code>.</p> <p>For example:</p> <pre>uddi:95f79c00-52a9-11e4-969c-d398081308d4</pre>
CONFIG-FILE	<p>(Optional). Name of the action template configuration file that contains the input parameters to be modified.</p> <div data-bbox="527 779 1268 947"><p>Note: If you are saving the file in a different location other than <code><CentraSiteInstall_Directory>/utilities</code>, provide the absolute file path.</p></div> <div data-bbox="527 953 1268 1157"><p>Important: If the action template whose configuration you want to modify has one or more active policies associated to it, the upload of the configuration file fails, and displays a warning message.</p></div>
IMPLEMENTATION-ZIP	<p>(Optional). Name of the Java implementation zip file that contains the elements to be modified.</p> <div data-bbox="527 1255 1268 1423"><p>Note: If you have saved the zip file to a different location other than the default location <code><CentraSiteInstall_Directory>/utilities</code>, provide the absolute path to the zip file.</p></div> <div data-bbox="527 1430 1268 1703"><p>Important: If the action template whose implementation you want to modify has one or more active policies associated to it, then the zip file upload fails, and displays a warning message. In this case, you will need to enable the <code>FORCE-EDIT</code> option to update the implementation details of action template.</p></div> <p>CentraSite has a sample implementation zip file <code>uniquenamechecker.zip</code> that you can use to enforce unique asset names in the CentraSite registry. This sample</p>

Parameter	Description
FORCE-EDIT	<p>implementation zip file is located in <CentraSiteInstall_Directory>/ demos\CustomActions\Java.</p> <p>(Optional). Allows to forcibly modify the implementation details of an action template that has one or more policies in the Active state.</p> <p>Supported values are - true, false</p>

Note:

In general, you are not allowed to modify the implementation details of an action template that has at least one policy in the **Active** state.

Examples (all in one line):

Providing action template configuration file:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd update Action Template -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-id uddi:95f79c00-52a9-11e4-969c-d398081308d4 -file
c:\temp\Custom_Action_Template_Config.xml
```

Providing action implementation file:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd update Action Template -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-id uddi:95f79c00-52a9-11e4-969c-d398081308d4 -file
c:\temp\Custom_Action_Template_Config.xml -implementationzip
c:\temp\custom_action_implementation.zip -forceedit true
```

The response to this command could be:

```
Executing the command : update Action Template
Action Template named Custom Action Template is updated successfully
Successfully executed the command : update Action Template
```

Deleting Action Template

Pre-requisites:

To delete an action template through CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool delete Action Template for this purpose.

Important:

You cannot delete an action template that is currently associated with one or more active policies. You need to deactivate the associated policies before deleting it.

➤ To delete an existing action template

- Run the command `delete Action Template`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd delete Action Template [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -id <ID>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
ID	The ID of the action template you want to delete. You can specify the UDDI key of the template using an optional prefix <code>uddi:.</code>

For example:

```
uddi:95f79c00-52a9-11e4-969c-d398081308d4
```

Examples (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd delete Action Template -url http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage -id uddi:95f79c00-52a9-11e4-969c-d398081308d4
```

The response to this command could be:

```
Executing the command : delete Action Template
Action Template named Custom Action Template is deleted successfully
Successfully executed the command : delete Action Template
```

Downloading Action Template Implementation File

Pre-requisites:

To download the Java implementation file of an action template through CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `download Action Template` for this purpose.

➤ To download the Java implementation file of an action template

- Run the command `download Action Template`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd download Action Template [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -id <ID> [-templateFile <TEMPLATE-FILE>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
ID	The ID of the action template whose implementation file you want to download. You can specify the UDDI key of the template using an optional prefix <code>uddi:</code> . For example: <code>uddi:95f79c00-52a9-11e4-969c-d398081308d4</code>
TEMPLATE-FILE	Name of the output file to download the Java implementation file of action template.

Note:

The default location is `<CentraSiteInstall_Directory>/utilities`.

If you have not specified this parameter, the complete implementation details for the action template is shown in the console.

Examples (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd download Action Template
-url http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-id uddi:95f79c00-52a9-11e4-969c-d398081308d4 -templateFile
D:\temp\custom_action_implementation.zip
```

The response to this command could be:

```
Executing the command : download Action Template
Action Template named Custom Action Template is downloaded successfully
to D:\temp\custom_action_implementation.zip
Successfully executed the command : download Action Template
```

Fetching Action Template Configuration File

Pre-requisites:

To fetch the configuration file that was used for constructing an action template through CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `get Action Template` for this purpose.

➤ To fetch the configuration file of an action template

- Run the command `get Action Template`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get Action Template [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -id <ID> [-file <CONFIG-FILE>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
ID	(Mandatory). The ID of the action template whose configuration file you want to retrieve. You can specify the UDDI key of the template using an optional prefix <code>uddi:</code> . For example: <code>uddi:95f79c00-52a9-11e4-969c-d398081308d4</code>
CONFIG-FILE	(Optional). Name of the output file to download the configuration file for action template.

Note:

The default location is `<CentraSiteInstall_Directory>/utilities`.

Examples (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get Action Template -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-id uddi:95f79c00-52a9-11e4-969c-d398081308d4 -file
D:\temp\Custom_Action_Template_Config.xml
```

The response to this command could be:

```
Executing the command : get Action Template
Action Template named Custom Action Template is downloaded successfully
to D:\temp\Custom_Action_Template_Config.xml
Successfully executed the command : get Action Template
```

Modifying Policy Action Implementation

Pre-requisites:

To modify the implementation details of a policy action through CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a Java tool named `UpdatePolicyActionImplementation.jar` for this purpose.

Note:

This tool is applicable only for custom action templates. The tool does not apply to predefined action templates installed with CentraSite.

➤ To modify the implementation details of a policy action

- Run the Java tool `UpdatePolicyActionImplementation.jar`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd UpdatePolicyActionImplementation.jar <CentraSite URL> <admin user id> <password> <action category name> <action template name> <action implementation file>`

The input parameters are:

Parameter	Description
<code><CentraSite URL></code>	The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
<code><admin user id></code>	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
<code><password></code>	The password for the CentraSite user identified by the parameter <code><admin user id></code> .
<code><action category name></code>	Name of the action category that contains the template to be modified. If the category name contains white spaces, enclose the name with <code>""</code> .
<code><action template name></code>	Name of the action template you want to modify. If the template name contains white spaces, enclose the name with <code>""</code> .
<code><action implementation file></code>	Name of the action implementation file to be modified. You can change the values in this file. If you are saving the file

Parameter	Description
	in a different location other than <CentraSiteInstall_Directory>/bin, provide the absolute file path.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd
UpdatePolicyActionImplementation.jar http://localhost:53307/CentraSite/CentraSite
DOMAIN\admin pAsSw0rD "Design Time" "My Custom Policy Action"
"c:/policy_action_implementation.zip"
```

Purging Orphaned Policy Parameters

Pre-requisites:

To purge orphaned policy parameters through CentraSite Command Line Interface, you must have the CentraSite Administrator role.

Before you run the tool, we strongly recommend that you create a database backup.

In some circumstances, you may not be able to delete an object because there could be policy parameters that are internally related to the object. You can purge such policy parameters if they are no longer used in an active policy.

CentraSite provides a Java tool named `PurgePolicyParameters.jar` for this purpose. This command tool removes all orphaned policy parameters from the CentraSite registry.

The purged policy parameters are logged into the `PurgePolicyParameters.log` file.

- Run the Java tool `PurgePolicyParameters.jar`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd PurgePolicyParameters.jar <CentraSite URL> <admin user id> <password>`

The input parameters are:

Parameter	Description
CentraSite URL	The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
admin user id	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
password	The password for the CentraSite user identified by the parameter <code>admin user id</code> .

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd PurgePolicyParameters.jar
http://localhost:53307/CentraSite/CentraSite DOMAIN\admin pAsSw0rD
```

Setting Up an Eclipse Java Project for Action Implementation

Pre-requisites:

To set up an Eclipse Java project for action template implementation through CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `setup Custom Actions` for this purpose.

➤ To set up an Eclipse Java project for the implementation of action template

- Run the command `setup Custom Actions`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd setup Custom Actions [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -packageName <PACKAGE-NAME> -className <CLASS-NAME> -file <CONFIG-FILE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
PACKAGE-NAME	Name of the package that contains the Java implementation for an action template.
CLASS-NAME	Name of the Java class file name.
	Note: Project name will be the same as the Java class file name.
CONFIG-FILE	Name of the output directory to set up the Eclipse Java project.

Examples (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd setup Custom Actions -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-id uddi:95f79c00-52a9-11e4-969c-d398081308d4 -packageName
com.softwareag.policy.custom.action.impl -className CustomActionImpl -file
D:\temp\customAction\CustomActionImpl
```

The response to this command could be:

```
Executing the command : setup Custom Actions
```

```
Custom policy action setup successful
at D:\temp\customAction\CustomActionImpl
Successfully executed the command : setup Custom Actions
```

Predefined Policies

System policies are predefined, design-time policies that CentraSite uses to perform internal operations (for example, identifying the components associated with a given object) and registry-wide governance functions (for example, ensuring the validity of policies prior to activation). Policies that are classified as predefined policies are system-wide in scope and execute at priority levels that are reserved for predefined policies. System policies are applied to assets regardless of the asset type's **Policies can be applied** property.

By default, predefined policies are not displayed by CentraSite Control. To view predefined policies, you must enable the **Show Predefined Policies** option on the Design/Change-Time Policy page.

If you belong to a role that includes the Manage System-Wide Design/Change-Time Policies permission, you have the ability to edit, delete, and deactivate CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be edited, deleted, or deactivated except under the direction of a technical representative from Software AG.

The Collector and Handler Policies Provided with CentraSite

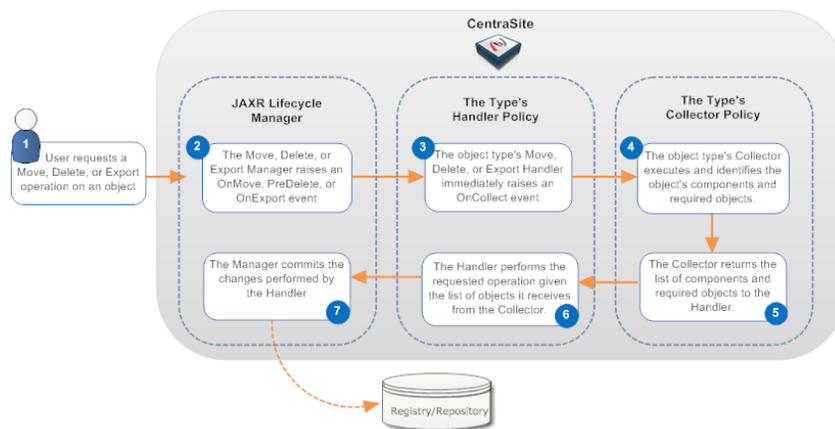
The collector and handler policies are used to delete, move, and export *composite objects* in a consistent way. Composite objects are objects that are made up, in part, of other registry objects. A Service object, for example, includes Operation objects, Binding objects, and Interface objects. When you delete, move, or export a Service object, you want CentraSite to delete, move, or export the Service object *and* its related components.

- The collector policy produces a list of the components (shared and nonshared) and required objects that are associated with a given instance of a composite object.
- The handler policy performs the delete, move, or export operation (depending on the type of handler that has been invoked) on the given object based on the list of components and required objects the handler receives from the collector.

For example, when you delete an instance of a Service object, the Service delete handler policy invokes the Service collector policy to identify the set of components and required objects associated with that particular instance of a Service (for example, its operations, bindings, interfaces, XML schemas, supporting documents, and so on). The delete handler then deletes the Service object and all of the nonshared components that were identified by the Service collector.

The following diagram illustrates how the handler and collector policies interact during a move, delete, or export operation:

The Handler and Collection Process



To understand collectors and handlers, you must understand the concepts of *shared components*, *nonshared components* and *required objects*.

The Default Collector and Handler Policies

The default collector and handler policies are used by many of the predefined types installed with CentraSite.

CentraSite also assigns the default collector and handler policies to custom types that you create.

Because CentraSite uses these policies for many of the predefined types and also assigns the handler policies to new types by default, you must not edit, delete, or deactivate them.

CentraSite uses the following policies for deleting, moving and exporting registry objects:

- Default Collector
- Default Delete Handler
- Default Move Handler
- Default Export Handler

Specialized Collector and Handler Policies for Assets

The following table lists the specialized collector and handler policies that CentraSite uses for deleting, moving, and exporting instances of certain predefined asset types.

You must not edit, delete, or deactivate any of the following policies.

Policy Name	Description
Collector Policy For BPEL Process	Performs the collection process for BPEL Process objects. For a list of the components that this collector returns, see the BPEL Process component list.

Policy Name	Description
Collector Policy for OData Service	Performs the collection process for OData Service objects. For a list of the components that this collector returns, see the OData Service component list.
Collector Policy For Process	Performs the collection process for Process objects. For a list of the components that this collector returns, see the Process component list.
Collector Policy for REST Service	Performs the collection process for REST Service objects. For a list of the components that this collector returns, see the XML/REST Service component list.
Collector Policy for Schema	Performs the collection process for XML Schema objects. For a list of the components that this collector returns, see the XML Schema component list.
Collector Policy for Virtual OData Service	Performs the collection process for Virtual OData Service objects. For a list of the components that this collector returns, see the Virtual OData Service component list.
Collector Policy for Virtual REST Service	Performs the collection process for Virtual REST Service objects. For a list of the components that this collector returns, see the Virtual XML/REST Service component list.
Collector Policy for Virtual Service	Performs the collection process for Virtual Service objects. For a list of the components that this collector returns, see the Virtual Service component list.
Collector Policy for Virtual XML Service	Performs the collection process for Virtual XML Service objects. For a list of the components that this collector returns, see the Virtual XML/REST Service component list.
Collector Policy for Web Service	Performs the collection process for Service objects. For a list of the components that this collector returns, see the Service component list.
Collector Policy for WS-Policy	Performs the collection process for WS-Policy objects. For a list of the components that this collector returns, see the WS-Policy component list.
Collector Policy for XML Service	Performs the collection process for XML Service objects. For a list of the components that this collector returns, see the XML/REST Service component list.
Collector Policy for IS Service Interface	Performs the collection process for Integration Server (IS) Service Interface objects. For a list of the components that this collector returns, see the IS Service Interface component list.
Virtual Service Export Handler	Performs the export process for Virtual Service objects.

Specialized Collector and Handler Policies for Other Registry Objects

The following table lists the set of collector and handler policies that CentraSite uses for deleting, moving, and exporting instances of certain registry objects that are not assets. You must not edit, delete, or deactivate these policies:

Policy Name	Description
Asset Type Export Handler	Performs the export process for asset type definitions.
Collector Policy For Lifecycle Model	Performs the collection process for lifecycle models.
Collector Policy For Policy	Performs the collection process for Policy objects.
Default User Move Handler	Performs the move operation on users.
Taxonomy and Category Export Handler	Performs the export process for taxonomies and their categories.

The Default Collector

The Default Collector policy identifies the components and required objects for a given object. It does this by looking for specific kinds of associations between the composite object and other objects in the registry.

How the Collector locates the components that are associated with a composite object

To identify the components of a composite object, the collector finds objects that are related to the composite object by an *aggregation relationship* or a *reverse aggregation relationship*.

- An aggregation relationship is indicated by the presence of an *Composition Using Source* Relationship attribute in the composite object. Like a regular Relationship attribute, an aggregated Relationship attribute associates an asset with other objects in the registry. However, an aggregated Relationship attribute additionally indicates that the associated objects are components of the object that contains the aggregated Relationship attribute.
- A reverse-aggregation relationship is indicated by a *Composition Using Target* Relationship attribute that is present in another registry object and points back to the object on which the collection is being performed. Like a regular Relationship attribute, a reverse-aggregated Relationship attribute also associates an asset with other objects in the registry. However, a reverse-aggregated Relationship attribute additionally indicates that the object that contains the reverse-aggregated attribute is a component of the object at the end of the relationship.

Conceptually, both the aggregated and reverse-aggregated forms of the Relationship attribute establish a parent-child relationship between the object being collected (the parent) and objects that are its components (its children). However, the aggregated form expresses the relationship from the perspective of the parent object (that is, the aggregated Relationship attribute exists in the composite object and identifies the object's components), whereas, the reverse-aggregated form expresses the relationship from the perspective of a child object (that is, the reverse-aggregated

Relationship attribute exists in a component object and identifies the composite object to which it belongs). The components that make up a composite object can be identified using aggregated Relationship attributes, reverse-aggregated Relationship attributes or a combination of the two.

Aggregated Relationship attributes are specified during the type definition for a composite object. Reverse-aggregated Relationship attributes are specified during the type definition of a component object.

How the default Collector determines whether a component is shared or nonshared

In the list of components that the Default Collector returns to a handler, a component is marked as *shared* or *nonshared*. To determine whether a component is shared or nonshared, the Default Collector checks whether the component is associated with any objects other than the one on which the collection is being performed.

- If the component is only associated with the object on which the collection is being performed, the Default Collector marks it as a nonshared object.
- If the component is associated with other objects in addition to the one on which the collection is being performed, the Default Collector marks it as a shared object.

How the default Collector locates the required objects that are associated with a composite object

In addition to components, the Default Collector also locates the required objects that are associated with a given object. It does this based on presence of required-object Relationship attributes in the composite object.

Like aggregated and reverse-aggregated Relationship attributes, a required-object Relationship attribute identifies objects that are to be collected. When a composite object contains a required-object Relationship attribute, the Default Collector collects the objects that the attribute references and marks them as required objects in the list that it returns to the handler.

In addition to those required objects that the Default Collector locates based on the required-object Relationship attributes that it finds in a composite object, the collector also returns the following items as required objects:

- The Type object associated with the object on which the collection is being performed.
- The repository items (that is, supporting documents and other attached files) associated with the object on which the collection is being performed.

Working with default Collector

The Default Collector policy is used by many of the predefined asset types installed with CentraSite. Do not edit, delete, or deactivate this policy.

Important:

The Default Collector is triggered by an OnCollect event. The sole purpose of this event is to trigger the collector policy for a given object type. Do not attempt to use the OnCollect event to create additional policies that execute before or after the Default Collector or any other collector

policy. Doing this can cause handlers to fail. Only one policy should execute when an OnCollect event occurs and that policy should be the collection policy for the object type on which the OnCollect event occurs.

The Default Delete Handler

The Default Delete Handler policy deletes the object on which the delete operation was requested and deletes all the nonshared components (as identified by the list that the collector returns to the handler) of that object.

The Default Delete Handler policy is used by many of the predefined asset types installed with CentraSite. CentraSite also uses this handler to delete policies and lifecycle models. Do not edit, delete, or deactivate this policy.

The Default Delete Handler is automatically assigned to new types that you add to CentraSite.

The Default Delete Handler is different than the other handler policies in that it executes on an event (the PreDelete event) that is also used to trigger user-defined policies. The Default Delete Handler has a priority of 1, which ensures that it executes before any user-defined policies that are also scoped for the PreDelete event.

The Default Move Handler

The Default Move Handler policy moves the object on which the move operation was requested and moves all the nonshared components (as identified by the list that the collector returns to the handler) of that object .

The Default Move Handler policy is used by many of the predefined asset types installed with CentraSite. Do not edit, delete, or deactivate this policy.

The Default Move Handler is automatically assigned to new types that you add to CentraSite.

Important:

The Default Move Handler is triggered by an OnMove event. The purpose of this event is to trigger the move handler policy for a given object type. Do not attempt to use the OnMove event to create additional policies that execute before or after the Default Move Handler (or any other move handler policy). Doing this could cause the handler to fail. Only one policy should execute when an OnMove event occurs and that policy should be the move handler policy for the object type on which the OnMove event occurs.

The Default Export Handler

The Default Export Handler policy generates an export archive file that contains the object on which the export operation was requested and all the nonshared components, shared components and required objects (as identified by the list that the collector returns to the handler) of that object .

The Default Export Handler policy is used by many of the predefined asset types installed with CentraSite. CentraSite also uses this handler to export policies. Do not delete, deactivate or modify this policy.

The Default Export Handler is automatically assigned to new types that you add to CentraSite.

Important:

The Default Export Handler is triggered by an OnExport event. The purpose of this event is to trigger the export handler policy for a given object type. Do not attempt to use the OnExport event to create additional policies that execute before or after the Default Export Handler (or before/after any other export handler policy). Doing this can cause the handler policy to fail. Only one policy should execute when an OnExport event occurs and that policy should be the export handler policy for the object type on which the OnExport event occurs.

Built-In Design/Change-Time Actions Reference

This section describes the built-in design/change-time actions that you can include in design/change-time policies for assets.

Summary of Actions in the ARIS Category

The following action templates are available in the ARIS category:

Action Template	Description
Notify ARIS Service	Notifies the ARIS APG service endpoint when: <ul style="list-style-type: none">■ A Process object in CentraSite is updated or deleted.■ A Service object (native or virtual) in CentraSite is updated or deleted, or when a user changes the state of the service to a “completed” lifecycle state (for example, the Productive state).

Summary of Actions in the Change-Time Category

The following action templates are available in the Change-Time category:

Action Template	Description
Change Activation State	Activates or deactivates a lifecycle model or a policy.
Change Deployment Status	Enables or disables the deployment of a virtual service.
Change Owner	Modifies the ownership of an asset.
Classify	Classifies an object by one or more taxonomy categories.
Delete RuntimeEvents and RuntimeMetrics	Deletes the logged events and metrics associated with a service.

Action Template	Description
Initiate Approval	Initiates an approval workflow.
Initiate Group-Dependent Approval	Initiates an approval workflow based on the group to which the requestor belongs.
Mark Pending on Runtime Policy Change	Marks a service as pending for redeployment on activation or deactivation of the applicable run-time policy.
Processing Steps Status	Enables or disables the Processing Steps profile for a virtual service.
Promote Asset	Promotes an asset to a new lifecycle stage (moving from one CentraSite instance to another CentraSite instance).
Register Consumer	Registers users and consumer applications as consumers of the requested asset.
Set Attribute Value	Assigns a value to a specified attribute in an organization, user or asset object.
Set Consumer Permission	Gives consumers instance-level permissions on the asset for which they have been registered.
Set State	Changes the lifecycle state of a lifecycle model, policy or asset.
UnClassify	Removes specified taxonomy categories from an object.
Validate Attribute Value	Validates the value of a specified attribute in an organization, user or asset against a list of allowed values.
Validate Classification	Checks whether an object is classified by a given taxonomy or taxonomy category.
Validate Lifecycle Model Activation	Checks whether a lifecycle model is ready to be activated.
Validate Policy Activation	Checks whether a policy is ready to be activated.
Validate Policy Deactivation	Verifies that a policy is not currently in-progress (that is, undergoing execution) so that it can be successfully deactivated.
Validate State	Validates the current state of a lifecycle model, policy or asset against a given list of states.

Summary of Actions in the Collector Category

The following action templates are available in the Collector category:

Important:

The actions in this category are used by the predefined collector policies that are installed with CentraSite. They are not intended to be used in user-defined policies.

Action Template	Description
Application Collector	Performs the collection process on Application objects.
BPEL Collector	Performs the collection process on BPEL Process objects.
BPM Process Project Collector	Performs the collection process on BPM Process objects.
Default Collector	Performs the collection process for types that do not have a specified collector.
IS Service Interface Collector	Performs the collection process on IS Service Interface objects.
Lifecycle Model Collector	Performs the collection process on Lifecycle Models.
OData Service Collector	Performs the collection process on OData Service objects.
Policy Collector	Performs the collection process on Policy objects (both design/change-time policies and run-time policies).
REST Service Collector	Performs the collection process on REST Service objects.
Report Collector	Performs the collection process on Report objects.
Schema Collector	Performs the collection process on XML Schema objects.
Virtual OData Service Collector	Performs the collection process on Virtual OData Service objects.
Virtual REST Service Collector	Performs the collection process on Virtual REST Service objects.
Virtual Service Collector	Performs the collection process on Virtual Service objects.
Virtual XML Service Collector	Performs the collection process on Virtual XML Service objects.
Web Service Collector	Performs the collection process on Service objects.
WS-Policy Collector	Performs the collection process on WS-Policy objects.
XML Service Collector	Performs the collection process on XML Service objects.
XPDL Collector	Performs the collection process on Process objects.

Summary of Actions in the Design-Time Category

The following action templates are available in the Design-Time category:

Action Template	Description
Create Auditable Events	Creates an audit log for a given object.
Validate Description	Validates the description of an object against a given pattern string.
Validate Name	Validates the name of an object against a given pattern string.
Validate Namespace	Checks that the target namespace attribute in a Service or XML Schema matches one of the valid namespaces in a given list.
Validate Service Binding	Checks that a Service supports the specified bindings.
Validate WSDL Size	Checks the size of the WSDL document associated with a Service to ensure that it falls within a specified range.
webMethods REST Publish	Creates a REST service in CentraSite from the published IS service interface object.

Summary of Actions in the Global Category

The following action templates are available in the Global category:

Action Template	Description
Attach Business UI Profiles for Asset Type	Appends the Business UI specific profiles to an asset type definition.
Call Web Service	Submits a given SOAP message to a specified web service.
Consumer WSDL Generator	Enables the Consumer WSDL option on the Specification profile of SOAP-based virtual services.
Default API Portal Permissions	Assigns the default permissions for API Portal objects.
Default Permission Handler	Propagates the instance-level permissions to components that are associated with an asset.

Action Template	Description
Enforce Unique Name	Ensures that the names of the Application Server type objects that are created in API Portal are unique.
Insight Deployment	Publishes API metadata to Insight Server.
Insight Undeployment	Revokes API metadata form Insight Server.
Generate Swagger 2.0 File	Generates the Swagger v2.0- compliant file when: <ul style="list-style-type: none">■ A REST API or Virtual REST API Service object (native or virtual) in CentraSite is created or updated, or when a user changes the lifecycle state of the REST API or Virtual REST API (for example, to a Productive state).■ A user executes the Generate Swagger 2.0 File policy on demand.
On Consumer Registration Request Send Email to Owner	Sends an email message to an object's owner when there is a consumer registration request for the object.
Notify Consumers	Notifies the registered API consumers when a Service object (native or virtual) in CentraSite is updated or deleted, or when a user changes the lifecycle state of the service (for example, to a Productive state).
Publish to API Portal	Publishes API metadata to API Portal repository.
Restrict API Portal Creation	Restricts the creation of API Portal objects other than the default privileged service.
Restrict Shared Composite Asset	Restricts manipulation of shared composite asset either as creation of a new asset creation or modification of an existing asset.
Send Email Notification	Sends an email message to a specified group of users.
Send Email Notification to Watchers	Sends an email notification to the watchers for an asset who are specific users asked to be notified for any modifications on that particular asset.
Set Business UI Profile Permissions	Assigns instance-level permissions to an asset's profiles in Business UI.
Set Instance and Profile Permissions	Assigns instance-level permissions to an asset and to the asset's profiles.

Action Template	Description
Set Permission for Asset's Objects	Grants View, Modify or Full permissions to assets's objects (Operation, Interface, and Binding Concept).
Set Permissions	Sets instance-level permissions on an policy.
Set Profile Permissions	Assigns instance-level permissions to an asset's profiles.
Set View Permission for Service and Service Related Object to Everyone Group	Grants View permission to all users (including guests) on a given service.
Unpublish from API Portal	Revokes API metadata form API Portal repository.
Verify Required Attributes	Ensures that all of the required attributes of an asset have a valid value.

Summary of Actions in the Handler Category

The following action templates are available in the Handler category:

Important:

The actions in this category are used by the predefined handler policies that are installed with CentraSite. They are not intended to be used in user-defined policies. For more information about the predefined handler policies, see the *CentraSite User's Guide*.

Action Template	Description
Asset Type Export Handler Action	Handler that CentraSite uses to export Type objects.
Default Delete Handler	Handler that CentraSite uses to delete instances of types that do not have a their own delete handlers.
Default Export Handler Action	Handler that CentraSite uses to export instances of types that do not have their own export handlers.
Default Move Handler	Handler that CentraSite uses to move instances of types that do not have their own move handlers (move to another user and to another organization).
Default Target Move Handler	Handler that CentraSite uses to export Target objects.
Default User Move Handler	Handler that CentraSite uses to move users to other organizations.
Event Type Delete Handler	Handler that CentraSite uses to delete Event Type objects.

Action Template	Description
Organization Export Handler	Handler that CentraSite uses to export organizations.
Reject Handler	Handler that prevents instances of a type from being deleted, exported, or moved, except as part of a composite object.
Taxonomy and Category Export Handler	Handler that CentraSite uses to export taxonomies and their categories.
Virtual Service Export Handler	Handler that CentraSite uses to export Virtual Service objects.

Summary of Actions in the WS-I Category

The WS-I category contains numerous actions from Basic Profile 1.1 and SSBP 1.0 that you can use to test a web service (of type Service or Virtual Service) for compliance with Web Service Interoperability (WS-I) standards.

Important:

A policy that contains WS-I actions must not contain any other type of action. If you need to execute other types of actions for the same event, you must place those actions in a separate policy.

Attach Business UI Profiles for Asset Type

Appends the Business UI specific profiles to an asset type definition.

Event Scope

Pre-Update

Object Scope

None.

Input Parameters

None.

Call Web Service

Submits a given SOAP message to a specified web service. You can use this action to notify external systems, through a SOAP message, of changes that occur in the registry.

If the web service returns a response, the response message is recorded to the policy log.

If the web service produces a SOAP fault or the service cannot be successfully performed for other reasons (for example, a network failure occurs), the policy action fails, and thus the policy itself fails. If the policy had been executed on a pre operation event (for example, Pre-Create, Pre-Delete), the requested operation is not executed.

Event Scope

Pre-Create

Post-Create

Pre-Update

Post-Update

Pre-Delete

Post-Delete

Pre-State Change

Post-State Change

OnTrigger

Object Scope

This action can be enforced on any object type that the policy engine supports.

Input Parameters

Service Endpoint (String). The URL of the web service that you want to call. Supported protocols are HTTP and HTTPS.

Example:

```
http://myServer:53307/wsstack/myService
```

Note:

If the web service that you want to invoke is registered in CentraSite, you can use the **Browse** button to select its URL.

HTTP Basic Auth Enabled (Boolean). Specifies whether the service is secured by Basic HTTP authentication.

If you enable this option, you can optionally specify the user ID and password that CentraSite is to submit when it invokes the service in the following parameters. If you leave these parameters empty, CentraSite submits the credentials belonging to the user who triggered this policy action.

HTTP Basic Auth Username The user ID that you want CentraSite to submit for HTTP basic authentication (if you

do not want CentraSite to submit the user ID of the user who triggered the policy).

HTTP Basic Auth Password The password associated with the user ID specified in **HTTP Basic Auth Username**.

SOAP Request Message (String). The SOAP message that CentraSite is to submit to the web service. This message can include substitution tokens, if you want to insert run-time data into it. For available tokens, see the list of **Substitution Tokens** shown in the [Send Email Notification](#) action.

```
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Body>
    <m:keylogger
      xmlns:m=" http://mycompany.example.org/key ">
      <serviceName>${entity.name}</serviceName>
      <assetType>${entity.type}</assetType>
      <key>${entity.attribute.Key}</key>
    </m:keylogger>
  </env:Body>
</env:Envelope>
```

SOAP Action (String). The SOAP action that CentraSite sets in the message. If you do not set this parameter, CentraSite sets the SOAP action to the empty string.

Connection Timeout (in milliseconds) (String). The length of time in milliseconds that CentraSite will wait for a response from the remote machine. If the timeout limit is exceeded, the policy action fails.

Content Type (String). The value that CentraSite is to assign to the Content-Type header in the SOAP request that it submits to the service.

Example:

```
application/soap+xml; charset=utf-8
```

If you do not specify **Content Type**, the value, `application/soap+xml`, is assigned to the SOAP request.

Change Activation State

Activates or deactivates a lifecycle model or a policy.

Event Scope

Post-State Change

OnTrigger

Object Scope

Lifecycle Model

Policy

Input Parameters

Change Activation State (String). The activation state to which you want to set the lifecycle model or policy as follows:

To

Active	<p>Activates the policy or lifecycle model.</p> <p>This action fails if it attempts to activate:</p> <ul style="list-style-type: none"> ■ A policy whose parameters are not set. ■ A lifecycle model that does not have an associated object type. ■ A lifecycle model whose associated object type is already assigned to another lifecycle model. <p>To prevent these types of failures from occurring, you should always execute the appropriate validation action before changing the activation state of a policy or lifecycle model.</p>
Inactive	<p>Deactivates the policy or lifecycle model.</p>

The following options are used to create policies that support the automatic deactivation of an older version of a policy or lifecycle model when a newer version is activated. In a lifecycle model for policies or lifecycle models, any state during which a policy or lifecycle is active must include a transition that places the policy or lifecycle model in one of the following activation states.

For example, the default lifecycle model for policies includes the Productive state. This is the only state in the model during which the policy is active. The Productive state includes a transition to the Retired state, which triggers a policy that switches the policy's activation state to Superseded and Retired.

Because the Productive state includes this transition, CentraSite is able to automatically deactivate an old version of a policy when a new version is activated. It simply locates and executes the transition that places the policy in one of the following states. In the case of policies, this transition is the one to the Retired state, which puts the policy in the Superseded and Retired state of activation.

Superseded	<p>Deactivates the policy and switches the policy's activation state to Superseded to indicate that the policy has been replaced by a newer version.</p>
------------	--

Retired Deactivates the policy and switches the policy's activation state to Retired to indicate that the policy is no longer available for use.

Superseded and Retired Deactivates the policy and switches the policy's activation state to Superseded and Retired to indicate that the policy has been replaced by a new version and is no longer available for use.

This action fails if it attempts to deactivate a policy that is in-progress. To prevent this type of failure from occurring, you should always execute the [Validate Policy Activation](#) action before using the Change Activation State action to deactivate a policy or lifecycle model.

Change Deployment Status

Enables or disables the deployment status of a virtual service. You use this action to specify whether the given virtual service is eligible or ineligible for deployment.

- When you enable the deployment status for a virtual service, you enable the controls on the **Deployment** profile. These controls enable authorized users to deploy, undeploy, or redeploy the virtual service.

Additionally, enabling the deployment status of a virtual service makes the virtual service eligible for automatic re-deployment when changes occur to its run-time policies.

- When you disable the deployment status for a virtual service, you disable the controls on the virtual service's **Deployment** profile (thus, preventing users from deploying, undeploying, or redeploying the virtual service).

When the deployment status for a virtual service is in the disabled state, the virtual service is not eligible for automatic re-deployment when changes occur to its run-time policies.

Note:

Disabling the deployment status of a virtual service *does not* undeploy the virtual service if it is already deployed. If the virtual service is currently deployed on a Mediator, it remains deployed there. However, administrators does not be able to undeploy or redeploy the virtual service from CentraSite Control until its deployment status is enabled.

To enable the deployment status of a virtual service, the following conditions must be satisfied:

- There must be at least one target defined in the registry.
- The Entry Protocol and Routing steps must be configured.

Typically, you use this action in combination with the [Processing Steps Status](#) action, which enables and disables the **Processing Steps** profile for a virtual service. For example, when you enable the **Deployment** profile, you generally disable the **Processing Steps** profile and vice versa.

Event Scope

Post-State Change

Object Scope

Service

Virtual Service

XML Service

Virtual XML Service

REST Service

Virtual REST Service

IS Service Interface

OData Service

Virtual OData Service

Input Parameters

Enable Deployment (Boolean). Specifies whether the virtual service is eligible for deployment (parameter set to Yes) or ineligible for deployment (parameter set to No).

Change Owner

Modifies the ownership of an asset.

Event Scope

Pre-Create

Post-State Change

Object Scope

Assets

Classify

Classifies the target object (that is, the object on which the policy was triggered) by one or more taxonomy categories. You can assign the taxonomy categories to a classification attribute of the target object, or you can assign the taxonomy categories as normal classifications of the target object.

The classifications you assign using this action will appear on the asset's **Classification** tab. The classifications you assign will also appear for the selected classification attribute.

You can select whether the classifications you specify with this action is added to the object's existing classifications or whether they replaces the object's existing classifications. This choice is

only available for multi-value classification attributes, that is, classification attributes that can reference more than one taxonomy category. If a classification attribute is a single-value classification attribute, its existing value is replaced by the new one.

Event Scope

Post-Create

Post-State Change

OnTrigger

OnConsumerRegistration

Object Scope

This action can be enforced on any object type that the policy engine supports.

Input Parameters

Classify With Attribute	(Object). (Array). This holds the parameters <code>Classification Attribute</code> and <code>Categories</code> .
Classification Attribute	(Optional). (String). This specifies the name of the object's attribute to which the following classification categories apply. If you leave this parameter empty, the classification categories is used as normal classifications of the target object.
Categories	(Taxonomy Node). (Array). The taxonomy nodes by which you want to classify the object.
Overwrite	(Boolean). If true, this specifies that you want to overwrite all existing classifications with the newly specified classifications. If false, the newly specified classifications are added to the existing classifications.

Note:

This option applies only to multi-value classification attributes. If a classification attribute is a single-value classification attribute, its existing value is replaced by the new one, regardless of the setting of the `Overwrite` parameter.

Consumer WSDL Generator

Enables the **Consumer WSDL** option on the Specification profile of SOAP-based virtual services.

Event Scope

Pre-Create

Pre-Update

Pre-State Change

OnTrigger

Object Scope

Virtual Service

Input Parameters

None.

Create Auditable Events

Creates an audit log for a given object.

Event Scope

Pre-Update

OnTrigger

Object Scope

This action can be enforced on any object type that the policy engine supports.

Default API Portal Permissions

Assigns the default permissions for API Portal objects.

Event Scope

Pre-Create

Object Scope

API Portal

Input Parameters

None.

Default Permission Handler

Propagate the asset's instance level permission to its composition and aggregation components.

Event Scope

Pre-Create

Pre-Update

Object Scope

Report Template

Assets

Input Parameters

None.

Delete RuntimeEvents and RuntimeMetrics

Deletes the events and metrics that have been logged for a service.

This action is included in the *Delete RuntimeEvents and RuntimeMetrics of Service* policy that is installed with CentraSite. This policy executes when a service is deleted. The policy ensures that the metrics and events associated with a service are removed from the run-time logs when a service is deleted.

Event Scope

Pre-Delete

Object Scope

Service

Virtual Service

XML Service

Virtual XML Service

REST Service

Virtual REST Service

IS Service Interface

OData Service

Virtual OData Service

Input Parameters

Delete Runtime Events *Boolean*. Specifies whether the events that have been logged for a service are to be deleted.

Delete Runtime Metrics *Boolean*. Specifies whether the runtime metrics that have been logged for a service are to be deleted.

Enforce Unique Name

Ensures that the names of objects that are created in CentraSite are unique.

This action is included in the *Enforce Unique Name* policy that is installed with CentraSite.

Event Scope

Pre-Create

Pre-Update

Pre-State Change

OnTrigger

Object Scope

This action can be enforced on any object type that the policy engine supports.

Input Parameters

`Enforce Across Organizations` (Boolean). If this parameter is set to True, then the unique name requirement for objects is enforced in all organizations defined in CentraSite.

`Allow Different Versions` (Boolean). If this parameter is set to True, then different versions of an object can exist in CentraSite with the same name.

Generate Swagger 2.0 File

Initiates the generation of Swagger v2.0-compliant file whenever a modification occurs on a REST API. The policy must be in the Productive state when this action is executed.

Event Scope

Post-Create

Post-Update

Post-State Change

OnTrigger

Object Scope

REST Service

Virtual REST Service

Input Parameters

None.

Insight Deployment

Enables you to publish API metadata to Insight Server gateway, thereby creating or updating the API information in Insight Server.

Event Scope

OnTrigger

Object Scope

Service

Virtual Service

XML Service

Virtual XML Service

REST Service

Virtual REST Service

IS Service Interface

OData Service

Virtual OData Service

Input Parameters

None.

Insight Undeployment

Removes specified API metadata from Insight Server gateway.

Event Scope

OnTrigger

Object Scope

Service

Virtual Service

XML Service

Virtual XML Service

REST Service

Virtual REST Service

IS Service Interface

OData Service

Virtual OData Service

Input Parameters

None.

Initiate Approval

Initiates an approval workflow.

When this action is executed, CentraSite initiates the approval process. CentraSite does not process any subsequent actions in the policy or execute the requested operation until the approvals specified by the Initiate Approval action are received.

CAUTION:

When you use this action on the Pre-State Change event, only certain kinds of actions can be executed *after* this action in an approval policy. Some actions, if they occur after this action, will cause the policy to fail.

Note:

To use the email options provided by this action, CentraSite must have a connection to an SMTP email server.

If You Migrate this Action from a Pre-8.2 Release

If you have a policy that contains this action and the policy was created prior to version 8.2, that policy continues to exhibit the old email-notification behavior (that is, it continues to send the earlier version's standard email message to approvers). If you want to use the email-notification enhancements that were introduced in version 8.2, simply edit the policy and enable the email parameters in the Initiate Approval action.

Event Scope

Pre-State Change

OnTrigger

OnConsumerRegistration

Object Scope

This action can be enforced on any object type that the policy engine supports.

Input Parameters

User (String). The user name that is used together with the Password parameter as authentication credentials for performing a lifecycle model state change on a service asset. The credentials are stored in the approval request and passed to the web service for completing the approval. The user specified must have the permissions required to perform the state change.

This parameter is only visible to users with the CentraSite Administrator role.

Password (String). The password that is used together with the `User` parameter as authentication credentials.

This parameter is only visible to users with the `CentraSite Administrator` role.

Approval Flow Name (String). The name to be given to the approval workflow that this action initiates. This name serves to identify the workflow in the `Approval History` log and in the approver's inbox.

An approval flow name can contain any combination of characters, including a space.

You can also include substitution tokens in the name to incorporate data from the target object on which the policy is acting. For a list of the allowed tokens, see the list of `Substitution Tokens` shown in the [Send Email Notification](#) action.

Approver Group (String). (Array). The user group (or groups) that identifies the set of users who are authorized to approve the requested operation.

Note:

If the user groups specified in `Approver Group` are empty at enforcement time, the user's request is auto-approved.

Approval is Needed From (String). The manner in which the approval is to be processed:

Value	Description
<code>AnyOne</code>	<i>Default.</i> The request can be approved or rejected by any single user in <code>Approver Group</code> . In this mode, only one user from the set of authorized approvers is required to approve or reject the request.
<code>EveryOne</code>	The request must be approved by all users specified in <code>Approver Group</code> . (It does not matter in which order the approvals are issued.) A single rejection will cause the request to be rejected.

Reject State The lifecycle state that is to be assigned to the object if the approval request is rejected. If this parameter is not specified, the object's lifecycle state does not change when a rejection occurs.

The lifecycle model must define a valid transition from the state that the target object is in at the time it is submitted for approval to the state specified in `Reject State`. Otherwise, the target object's state does not be switched when a rejection occurs.

Send Pending Approval Email (Boolean). Specifies whether `CentraSite` is to send an email message to specified users and groups when the request is initially submitted for approval. If you enable this option, you must set the following

parameters to specify the text of the message and to whom it is to be sent.

Note:

- If the request is auto-approved, this message is not sent.
- CentraSite automatically sends the email message to the approvers in addition to the users and groups that you specify below.

Users (Array of Users). Users who are to receive the email.

Note:

You can specify the recipients of the email using the `Users` parameter, the `Groups` parameter, or both.

Groups (Array of Groups). Groups whose users are to receive the email.

Note: CentraSite will only send the email to those users in the group whose CentraSite user account includes an email address.

Subject (String). The text that you want to appear in the subject line of the email. This text can include substitution tokens to insert run-time data into the subject line. For available tokens, see the list of Substitution Tokens shown in the [Send Email Notification](#) action.

Use Email Template (Email Template). Specifies the template that is to be used to generate the body of the email message.

Note:

- You can use the predefined template, `PendingNotification.html`, for pending-approval notifications if you do not want to create an email template of your own.
- If you use an email template to generate the body of the message, you cannot specify the body of the message using the `Custom Message` parameter. (In other words, you specify the body of the message using either the `Use Email Template` *or* the `Custom Message` parameter.)

Custom Message (TextArea). The text of the email message. This text can include substitution tokens to insert run-time data into the message. For available tokens, see the list of Substitution Tokens shown in the action.

Note:

	<p>If you use the <code>Custom Message</code> parameter to specify the body of the email message, you cannot generate the body of the message using an email template. (In other words, you specify the body of the message using either the <code>Custom Message</code> or the <code>Use Email Template</code> parameter.)</p>
<code>Format</code>	<p>(String). Specifies whether the message in the <code>Custom Message</code> parameter is formatted as HTML or plain text. For more information about using this option, see the <i>CentraSite User's Guide</i>.</p>
<code>Include owner in notification</code>	<p>(Boolean). When the parameter is enabled, CentraSite sends the email to the owner of the object (on which the notification policy is acting) in addition to the other recipients.</p>
<code>Send Approval Email</code>	<p><i>Boolean</i>. Specifies whether CentraSite is to send an email message to specified users and groups when the request is approved. If you enable this option, you must set the following parameters to specify the text of the message and to whom it is to be sent.</p>
	<p>Note:</p> <ul style="list-style-type: none"> ■ CentraSite automatically sends the email message to the user who submitted the approval request in addition to the users and groups that you specify below. ■ When the EveryOne option is specified in the <code>Approval is Needed From</code> parameter, CentraSite sends this email only after all approvers have approved the request.
<code>Users</code>	See description of <code>Users</code> parameter.
<code>Groups</code>	See description of <code>Groups</code> parameter.
<code>Subject</code>	See description of <code>Subject</code> parameter.
<code>Use Email Template</code>	See description of <code>Use Email Template</code> parameter.
	<p>Note: You can use the predefined template, <code>ApprovalNotification.html</code>, for approval notifications if you do not want to create an email template of your own.</p>
<code>Custom Message</code>	See description of <code>Custom Message</code> parameter.
<code>Format</code>	See description of <code>Format</code> parameter.
<code>Include owner in notification</code>	See description of <code>Include owner in notification</code> parameter.

Send Rejection Email *Boolean.* Specifies whether CentraSite is to send an email message to specified users and groups when the request is rejected. If you enable this option, you must set the following parameters to specify the text of the message and to whom it is to be sent.

Note: CentraSite automatically sends the email message to the approvers (except for the approver who rejected the request) and to the user who submitted the approval request in addition to the users and groups that you specify below.

Users	See description of Users parameter.
Groups	See description of Groups parameter.
Subject	See description of Subject parameter.
Use Email Template	See description of Use Email Template parameter.
	<p>Note: You can use the predefined template, <code>RejectApprovalNotification.html</code>, for rejection notifications if you do not want to create an email template of your own.</p>
Custom Message	See description of Custom Message parameter.
Format	See description of Format parameter.
Include owner in notification	See description of Include owner in notification parameter.

Initiate Group-Dependent Approval

Initiates an approval workflow based on the group to which the requestor belongs. If the requestor does not belong to any of the groups specified in the Triggering Groups array, approval is waived and the action is considered to be completed successfully.

CAUTION:

When you use this action on the Pre-State Change event, only certain kinds of actions can be executed *after* this action in an approval policy. Some actions, if they occur after this action, will cause the policy to fail.

Note:

To use the email options provided by this action, CentraSite must have a connection to an SMTP email server.

If You Migrate this Action from a Pre-8.2 Release

If you have a policy that contains this action and the policy was created prior to version 8.2, that policy continues to exhibit the old email-notification behavior (that is, it continues to send the earlier version's standard email message to approvers). If you want to use the email-notification enhancements that were introduced in version 8.2, simply edit the policy and enable the email parameters in the Approval action.

Event Scope

Pre-State Change

OnConsumerRegistration

Object Scope

This action can be enforced on any object type that the policy engine supports.

Input Parameters

User (String). The user name that is used together with the Password parameter as authentication credentials for performing a lifecycle model state change on a service asset. The credentials are stored in the approval request and passed to the web service for completing the approval. The user specified must have the permissions required to perform the state change.

This parameter is only visible to users with the CentraSite Administrator role.

Password (String). The password that is used together with the User parameter as authentication credentials.

This parameter is only visible to users with the CentraSiteAdministrator role.

Approval (Object). (Array). The list of groups whose membership determines whether the request requires approval, and if so, to which group of approvers the request is to be routed. Each object in the Approval array must contain the following information:

Parameter	Description
Triggering Groups	(String). (Array). The user group (or groups) that identifies the users whose requests must be approved.
Approval Flow Name	(String). The name to be given to the approval workflow that this action initiates. This name serves to identify the workflow when activity relating to it appears in the Approval History log or an approver's inbox. An Approval Flow Name can contain any combination of characters, including a space. You can also include substitution tokens in the name to incorporate data from the target object on which the policy is acting. For a list of the allowed tokens, see the list of

Substitution Tokens shown in the [Send Email Notification](#) action.

Approver Group (String). (Array). The user group (or groups) that identifies the set of users who are authorized to approve the requested operation.

Note:

If the user groups specified in `Approver Group` are empty at enforcement time, the user's request is auto-approved.

Approval is needed from (String). The manner in which the approval is to be processed as follows:

Value	Description
AnyOne	(Default). The request can be approved or rejected by any single user in <code>Approver Group</code> . In this mode, only one user from the set of authorized approvers is required to approve or reject the request.
EveryOne	The request must be approved by all users specified in <code>Approver Group</code> . (It does not matter in which order the approvals are issued.) A single rejection will cause the request to be rejected.

Reject State The lifecycle state that is to be assigned to the object if the approval request is rejected. If this parameter is not specified, the object's lifecycle state does not change when a rejection occurs.

The lifecycle model must define a valid transition from the state that the target object is in at the time it is submitted for approval to the state specified in `Reject State`. Otherwise, the target object's state does not be switched when a rejection occurs.

Mark Pending on Runtime Policy Change

Marks the deployed virtual services or consumer applications that are within the scope of run-time policy as pending for redeployment on activation or deactivation of the policy. After the policy is activated, the virtual services and consumer applications are automatically redeployed.

This action is included in the *Mark Pending-For-Redeployment On RuntimePolicy Change* policy that is installed with CentraSite. This policy executes when a run-time policy switches to the Productive state (which activates the policy) or Suspended state (which deactivates the policy).

If you customize the lifecycle model that CentraSite provides for policies and you add additional states to the model, you must execute this action during any transition that changes the activation state of a policy.

Event Scope

Pre-State Change

Object Scope

Policy

Input Parameters

None.

Notify ARIS Service

Notifies the ARIS APG Service endpoint with the SOAP request message provided in this action. The APG Service endpoint is picked up from the associated ARIS Application Server.

You can use this action in the following policies:

- Notify ARIS on Process Changes
- Notify ARIS on Service Changes
- Notify ARIS on Service Completion
- Notify ARIS on Service Deletion

For information about using CentraSite with ARIS in the *CentraSite Administrator's Guide*.

Event Scope

Pre-Create

Post-Create

Pre-Update

Post-Update

Pre-Delete

Post-Delete

Pre-State Change

Post-State Change

OnTrigger

Object Scope

Process

Service

Input Parameters

HTTP Basic Auth Enabled	<p>(Boolean). Specifies whether the service is secured by Basic HTTP authentication.</p> <p>If you enable this option, you can optionally specify the user ID and password that CentraSite is to submit when it invokes the service in the following parameters. If you leave these parameters empty, CentraSite will submit the credentials belonging to the user who triggered this policy action.</p> <p>HTTP Basic Auth Username The user ID that you want CentraSite to submit for HTTP basic authentication (if you do not want CentraSite to submit the user ID of the user who triggered the policy).</p> <p>HTTP Basic Auth Password The password associated with the user ID specified in HTTP Basic Auth Username.</p>
SOAP Request Message	<p>(String). The SOAP message that CentraSite is to submit to the ARIS service. This message can include substitution tokens, if you want to insert run-time data into it. For available tokens, see the list of Substitution Tokens shown in the Send Email Notification action.</p> <pre style="background-color: #f0f0f0; padding: 10px;"> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://www.idsscheer.com/age/webMethods/"> <soapenv:Header/> <soapenv:Body> <web:UpdateServiceRequest> <dbname>\${context.ARIS_DB_CONTEXT}</dbname> <language>\${user.locale}</language> <serviceDetail> <guid>\${entity.key}</guid> <name>\${entity.name}</name> <url>\${entity.URL}</url> <lifeCycleState>\${entity.state}</lifeCycleState> <owner>\${entity.owner}</owner> <description>\${entity.description}</description> <organization>\${entity.organization}</organization> <version>\${entity.version}</version> \${entity.attribute.Operations} </serviceDetail> </web:UpdateServiceRequest> </soapenv:Body> </soapenv:Envelope> </pre>
SOAP Action	<p>(String). The SOAP action that CentraSite sets in the message. If you do not set this parameter, CentraSite sets the SOAP action to an empty string.</p>
Connection Timeout (in milliseconds)	<p>(Number). The length of time (in milliseconds) that CentraSite will wait for a response from the remote machine. If the timeout limit is exceeded, the policy action fails.</p>
Content Type	<p>(String). The value that CentraSite is to assign to the Content-Type header in the SOAP request that it submits to the service.</p>

Example:

```
application/soap+xml; charset=utf-8
```

If you do not specify Content Type, the value `application/soap+xml` is assigned to the SOAP request.

Notify Consumers

Sends broadcast notifications with the message content provided in this action. The message is broadcast to the all registered consumers of the configured API.

Event Scope

OnTrigger

Object Scope

Service

Virtual Service

XML Service

Virtual XML Service

REST Service

Virtual REST Service

IS Service Interface

OData Service

Virtual OData Service

Input Parameters

Email Subject	(String). The text that you want to appear in the subject line of the broadcast message.
Email Body	(String). The content that you want to appear in the body text of the broadcast message.
Format	(String). Specifies whether the broadcast message in the Email Body parameter is formatted as HTML or plain text. The default format for the broadcast message is 'html'.
Email Template	(Email Template). Specifies the name of the template (.html) file that is to be used to generate the body of the broadcast message.

Note:

You can use the predefined template, `NotifyConsumer.html`, as your email template if you do not want to create an email template of your own.

Note:

If you use the `Email Body` parameter to specify the body of the broadcast message, you cannot generate the body of the broadcast message using an email template. (In other words, you specify the body of the broadcast message using either the `Email Body` or the **Email Template** parameter.)

On Consumer Registration Request Send Email to Owner

Sends an email message to the owner of an asset for any consumer registration request on that particular asset.

Event Scope

Pre-Create

Object Scope

Consumer Registration Request

Input Parameters

`Custom Message` (TextArea). The text of the email message. This text can include substitution tokens to insert run-time data into the message. For available tokens, see the list of Substitution Tokens shown in the [Send Email Notification](#) action.

Note:

If you use the `Custom Message` parameter to specify the body of the email message, you cannot generate the body of the message using an email template. (In other words, you specify the body of the message using either the `Custom Message` or the `Use Email Template` parameter.)

`Subject` (String). The text that you want to appear in the subject line of the email. This text can include substitution tokens to insert run-time data into the subject line. For available tokens, see the list of Substitution Tokens shown in the [Send Email Notification](#) action.

`Format` (String). Specifies whether the message in the `Custom Message` parameter is formatted as HTML or plain text.

`Use Email Template` (Email Template). Specifies the template that is to be used to generate the body of the email message. This text can include substitution tokens to insert run-time data into the subject line. For available tokens, see the list of Substitution Tokens shown in the [Send Email Notification](#) action.

Note:

- You can use the predefined template, `PendingNotification.html`, for pending-approval notifications if you do not want to create an email template of your own.

- If you use an email template to generate the body of the message, you cannot specify the body of the message using the `Custom Message` parameter. (In other words, you specify the body of the message using either `Use Email Template` or `Custom Message` parameter.)

Processing Steps Status

Enables or disables the **Processing Steps** profile for a virtual service.

- When you enable the processing steps status for a virtual service, you enable the controls on the **Processing Steps** profile for that virtual service. These controls enable authorized users to modify the processing steps for the virtual service.
- When you disable the processing steps status for a virtual service, you disable the controls on the **Processing Steps** profile. While this profile is disabled, users cannot make changes to the virtual service's processing steps.

Typically, you use this action in combination with the [Change Deployment Status](#) action, which enables and disables the **Deployment** profile for a virtual service. For example, when you enable the **Processing Steps** profile for a virtual service, you generally disable the **Deployment** profile and vice versa.

Event Scope

Post-State Change

Object Scope

Service

Virtual Service

XML Service

Virtual XML Service

REST Service

Virtual REST Service

IS Service Interface

OData Service

Virtual OData Service

Input Parameters

Enable Processing Steps	<i>Boolean.</i> Specifies whether the Processing Steps profile for a virtual service is enabled (parameter set to Yes) or disabled (parameter set to No).
-------------------------	---

Promote Asset

This policy action allows you to promote an asset instance to a different CentraSite stage. The action can be executed on a lifecycle pre-state change, post-state change, or on an OnTrigger event. The configurations cover the following options:

- **Specify a stage to promote to**

This can be either the name of a lifecycle stage or the URL of the target registry.

- **Specify optional user credentials for the target stage**

The credentials specify a user name and password of a user defined on the target registry. This user should have the required permissions to create the asset on the target registry.

- **Include referenced objects in the promotion set**

Assets that are referenced by the asset being promoted can be included in the promotion process.

- **Keep the asset owner unchanged**

You can specify that the owner of the asset in the source registry is the owner in the target registry. If this user does not exist in the target registry, the owner is the user specified in the optional user credentials described above.

This user should be able to create assets in the target organization, which can be any of the following, depending on the input parameters you specify:

- The organization mentioned in the `Target Organization` parameter.
- The organization to which the user in the target registry belongs.
- The organization to which the triggering user or the user in the `Username` parameter belongs.

- **Replace existing registry objects in the target stage**

If an asset already exists on the target stage, it may be replaced by the asset being promoted.

- **Specify a target organization name**

When the asset is promoted, it will belong to the organization specified.

- **Keep the lifecycle state**

You can specify a lifecycle state for the promoted asset on the target registry. If you do not specify a state, the promoted asset is placed in the initial state of the lifecycle model on the target registry.

- **Ignore the asset's type definition in the promotion set**

If the asset's type definition already exists on the target stage, it may be ignored in the promotion process.

- **Ignore API keys and OAuth2 tokens in the promotion set**

API keys and OAuth2 tokens that are available for the asset being promoted can be ignored in the promotion process.

Event Scope

Pre- State Change

Post-State Change

OnTrigger

Object Scope

Assets

Input Parameters

The following table lists the input parameters for the policy action.

Target Stage	(String). The name of the target stage to which the asset is promoted. If a value is specified for the parameter Target Stage URL, the value of Target Stage is used instead of the value of the parameter Target Stage URL. At least one of the parameters Target Stage or Target Stage URL must be specified, that is, they cannot both be empty.
Target Stage URL	(String). The URL of the target CentraSite registry. If a value is specified for the parameter Target Stage URL, the value of Target Stage is used instead of the value of the parameter Target Stage URL. At least one of the parameters Target Stage or Target Stage URL must be specified, that is, they cannot both be empty.
Username	(String). (Optional). The user name and password are used as authentication credentials for the target stage. The assets is created in the target by this user. If the user name and password are not supplied, the user name and password of the triggering user on the source stage is used. If this user is not defined on the target stage, the promotion fails.
Password	(String). (Optional). The user name and password are used as authentication credentials for the target stage. The assets is created in the target by this user. If the user name and password are not supplied, the user name and password of the triggering user on the source stage is used. If this user is not defined on the target stage, the promotion fails.
Target Organization	(String). (Optional). Specifies the owning organization of the asset on the target stage. This can only happen if the specified organization exists on the target.
Include Referenced Assets	Specifies whether the referenced assets (referenced through associations) of the applied asset are included for the promotion.

	<p>A selection of the <code>Include Referenced Assets</code> check box indicates that the referenced assets are promoted. A cancellation of the <code>Include Referenced Assets</code> check box indicates that only the specified asset is promoted.</p> <p>The <code>Include Referenced Assets</code> check box remains selected by default.</p>
<code>Keep Owner</code>	<p>Specifies if the current owner is the owner in the target registry. This can only happen if the owner also exists as a user on the target registry and has the permissions required to create assets.</p> <p>A selection of the <code>Keep Owner</code> check box indicates that the asset owner on the target stage is the same owner as on the source stage. A cancellation of the <code>Keep Owner</code> check box indicates that the owner is the specified user from the <code>User Name</code> parameter.</p> <p>The <code>Keep Owner</code> check box remains cancelled by default.</p>
<code>Replace Existing Assets</code>	<p>Specifies if an asset that already exists on the target stage may be replaced by the asset being promoted.</p> <p>A selection of the <code>Replace Existing Assets</code> check box indicates that an asset on the target stage can be replaced. A cancellation of the <code>Replace Existing Assets</code> check box indicates that an existing asset on the target stage cannot be replaced.</p> <p>The <code>Replace Existing Assets</code> check box remains cancelled by default.</p>
<code>Keep Lifecycle State</code>	<p>Specifies if the promoted asset should keep the lifecycle state that it has on the source stage. This can only happen if the lifecycle model used on the source stage is also defined and active on the target stage.</p> <p>A selection of the <code>Keep Lifecycle State</code> check box indicates that an asset on the target stage have the same state as on the source stage. A cancellation of the <code>Keep Lifecycle State</code> check box indicates that the promoted asset is set to a lifecycle state according to the combinations as shown in the table below.</p> <p>The <code>Keep Lifecycle State</code> check box remains cancelled by default.</p>
<code>Remove Unreferenced Associations</code>	<p>Specifies whether the referenced assets (referenced through associations) of the applied asset are excluded from the promotion.</p> <p>A selection of the <code>Remove Unreferenced Associations</code> check box indicates that the referenced assets are not promoted. A cancellation of the <code>Remove Unreferenced Associations</code> check box indicates that the referenced assets are also promoted.</p> <p>The <code>Remove Unreferenced Associations</code> check box remains cancelled by default.</p>
<code>Ignore Asset Types</code>	<p>Specifies whether the type definition of the applied asset is included for the promotion.</p>

A selection of the `Ignore Asset Types` check box indicates that the type definition will be ignored in the promotion process. A cancellation of the `Ignore Asset Types` check box indicates that the type definition will also be promoted.

The `Ignore Asset Types` check box remains cancelled by default.

`Ignore API Keys and OAuth2 Tokens` Specifies whether the API keys and OAuth2 tokens of the applied asset are included for the promotion.

A selection of the `Ignore API Keys and OAuth2 Tokens` check box indicates that the API keys and OAuth2 tokens will be ignored in the promotion process. A cancellation of the `Ignore API Keys and OAuth2 Tokens` check box indicates that the API keys and OAuth2 tokens will also be promoted.

The `Ignore API Keys and OAuth2 Tokens` check box remains cancelled by default.

As noted in the table, some of the promotion operations are only possible if the target stage contains users, organizations, and lifecycle models that are compatible with those defined on the source stage. The possible combinations are listed in the following tables.

Note:

During the promotion process, CentraSite copies the metadata of an asset from the source instance to the target instance. However, if the action is to be executed during a pre-state change event, the changes due to the other actions in the source instance are not reflected in the target instance. You need to explicitly update the asset if you want that change reflected in the target instance, too.

Important:

Before you activate a policy that includes the `Promote Asset` action, ensure that the target's specified target stage URL or target stage is active and the user credentials of target registry are valid. To check this, click the **Check Connection** button. If the connection is not active and valid, activate the target specified in *Target Stage* or *Target Stage URL*, and modify the user credentials as required.

Target Organization and Target Owner

When the asset is promoted to the target registry, it belongs to a specific organization and is owned by a specific user. The organization and owner on the target registry are not necessarily the same organization and owner as on the source registry.

The owner on the target registry can be one of the following:

- the same owner as on the source registry (called User A in the following description)
- the user specified in the `Username` parameter (called User B in the following description)
- the triggering user, that is, the user who activates the asset promotion (called User C in the following description)

The organization on the target registry can be one of the following:

- the organization specified in the `Target Organization` parameter (called Organization P in the following description)
- the organization of the user supplied in the `Username` parameter (called Organization Q in the following description)
- the organization of the triggering user (called Organization R in the following description)

Target Owner

This user is the owner under these circumstances
User A	If <code>Keep Owner</code> is specified, and User A has permission to create assets in Organization P or Q or R.
User B	If User A does not meet the requirements described in the previous row, and User B is defined.
User C	If User B not meet the requirements described in the previous row.

Target Organization

This organization is the owning organization under these circumstances
Organization P	If <code>Target Organization</code> is specified and the target owner defined in the above table has permission to create assets in this organization.
Organization Q	If Organization P does not meet the requirements described in the previous row, and User B is defined.
Organization R	If Organization Q does not meet the requirements described in the previous row.

CentraSite attempts to create the asset on the target registry using the resulting combination of target owner and target organization. If the given user does not have permission to create assets in the given organization, the promotion fails.

Keep Lifecycle State

Keep LCM State	Availability of the same LCM in the target stage	Does target have its own LCM	Result state of the promoted asset
yes	yes	na	Same state as in the source.
yes	no	yes	Initial state of the LCM in the target.
yes	no	no	No state assigned.

Keep LCM State	Availability of the same LCM in the target stage	Does target have its own LCM	Result state of the promoted asset
no	n/a	yes	Initial state of the LCM in the target.
no	n/a	no	No state assigned.

Publish to API Portal

Enables you to publish API metadata to an API Portal, thereby creating or updating the API information in the API Portal repository.

Event Scope

Pre-State Change

Post-State Change

OnTrigger

Object Scope

Service

Virtual Service

XML Service

Virtual XML Service

REST Service

Virtual REST Service

IS Service Interface

OData Service

Virtual OData Service

Input Parameters

API Portal (Optional). (String). (Array). The name of the API Portal to which the API would be published. This assumes that you have already registered the API Portal in CentraSite.

Note:

However, if this action is to be executed in a different event other than OnTrigger, for example, Pre-State Change, or Post-State Change, that is not provided by default, you *must* specify a value for this field.

Endpoint Category	(Optional). (String). (Array). The names of specific taxonomy categories by which the endpoints of the API are classified.
REST Service Attributes	(Optional). (String). (Array). A metadata bundle can be supplied with additional information of a RESTful API and published to an API Portal. You use this field to specify additional attributes of the REST API to be published to API Portal.
SOAP Service Attributes	(Optional). (String). (Array). A metadata bundle can be supplied with additional information of a SOAP-based API and published to an API Portal. You use this field to specify additional attributes of the SOAP API to be published to API Portal.

Register Consumer

Registers users, groups, and consumer applications (as specified by the requestor) as consumers of an asset. This action creates a `consumed-by` relationship between the asset and the specified consumers. Once established, this relationship is visible in the asset's **Basic Information** profile.

The following actions are typically used in conjunction with the Register Consumer action.

- The approval actions ([Initiate Approval](#) or [Initiate Group-Dependent Approval](#)) are generally used to obtain necessary approvals prior to executing the Register Consumer action.
- The [Set Consumer Permission](#) action is typically executed after the Register Consumer action to give the specified consumers access to the requested asset.

Event Scope

OnConsumerRegistration

Object Scope

Assets

Input Parameters

None.

Restrict API Portal Creation

Restricts the creation of API Portal objects other than the default privileged service.

Event Scope

Pre-Create

Object Scope

API Portal

Input Parameters

None.

Restrict Shared Composite Asset

Restricts manipulation of shared composite asset either as creation of a new asset creation or modification of an existing asset.

Event Scope

Pre-Create

Pre-Update

Object Scope

Assets

Input Parameters

None.

Send Email Notification

Sends an email message to specified users and groups when specified events occur. For example, you might use this action to alert a certain group of administrators when an asset switches to a particular state.

Note:

- To use this action, CentraSite must have a connection to an SMTP email server.
- During an iteration of the policy, if the connection to a SMTP email server fails, this policy action returns a failure code. CentraSite writes the failure message to the policy log, however, performs the next action in the policy (if one exists).

Event Scope

Pre-Create

Post-Create

Pre-Update

Post-Update

Pre-Delete

Post-Delete

Pre-State Change

Post-State Change

OnConsumerRegistration

OnTrigger

Object Scope

This action can be enforced on any object type that the policy engine supports.

Input Parameters

Users	(Array of Users). Users who are to receive the email.
	<p>Note: You can specify the recipients of the email using the <code>Users</code> parameter, the <code>Groups</code> parameter, or both.</p>
Groups	(Array of Groups). Groups whose users are to receive the email.
	<p>Note: CentraSite will only send the email to those users in the group whose CentraSite user account includes an email address.</p>
Subject	(String). The text that you want to appear in the email's subject line. This text can include substitution tokens to insert run-time data into the subject line.
Use Email Template	(Email Template). Specifies the template that is to be used to generate the body of the email message.
	<p>Note:</p> <ul style="list-style-type: none"> You can use the predefined template, <code>ChangeNotification.html</code>, as your email template if you do not want to create an email template of your own. If you use an email template to generate the body of the message, you cannot specify the body of the message using the <code>Custom Message</code> parameter. (In other words, you specify the body of the message using either the <code>Use Email Template</code> or the <code>Custom Message</code> parameter.)
Custom Message	(TextArea). The text of the email message. This text can include substitution tokens to insert run-time data into the message.
	<p>Note: If you use the <code>Custom Message</code> parameter to specify the body of the email message, you cannot generate the body of the message using the <code>Use Email Template</code> parameter. (In other words, you specify the body of the message using either the <code>Custom Message</code> or the <code>Use Email Template</code> parameter.)</p>
Format	(String). Specifies whether the custom mail message is formatted as HTML or plain text.
Include owner in notification	(Boolean). When enabled, this parameter sends the email notification to the owner of the object on which the policy is acting in addition to the users specified by the <code>Users</code> and <code>Groups</code> parameters.

For detailed information about using custom email notifications with the policy action, see [“Configuring Email Notifications” on page 842](#).

Substitution Tokens

The following list describes substitution tokens that you can use to incorporate data from the run-time instance of a policy into the email. For example, you can use tokens to return information about the object on which the policy is acting, identify the user who triggered the policy, and indicate what type of event caused the policy to fire.

Be aware that some tokens are only meaningful for certain types of objects. User objects, for example, do not have a Description attribute, so the `${entity.description}` token has no meaning for a User object. If you use a substitution token that is not supported by the policy's target object, CentraSite simply replaces the substitution token with a space at enforcement time.

If the target object includes the requested attribute, but the attribute itself has no value, CentraSite also replaces the substitution token with a space in the email message. If the requested attribute contains an array of values, CentraSite inserts the values into the email as a comma-separated list.

This token...	Inserts the following information into the parameter value at execution time...
<code>\${api.usage}</code>	A usage description for the API access token.
<code>\${entity.approver}</code>	The name of the user who approved or rejected the request. Note: This token is only meaningful in the email messages that are issued by the Initiate Approval or Initiate Group-dependent Approval actions. If it is used in a context where there is no approver or an approval request, the token is simply replaced with a space.
<code>\${entity.approvercomments}</code>	A comment provided by the approver when approving or rejecting the request. Note: This token is only meaningful in the email messages that are issued by the Initiate Approval or Initiate Group-dependent Approval actions. If it is used in a context where there is no approval request, the token is simply replaced with a space.
<code>\${entity.attribute. attributeName}</code>	A value for the attribute specified in <i>attributeName</i> . You can use this token with all of the attribute types, excluding the Classification, File, and Relationship attribute types. Important: Make sure you specify the attribute's schema name and not its display name in <i>attributeName</i> .

This token...	Inserts the following information into the parameter value at execution time...
<code>\${entity.BUIAssetURL}</code>	A deep link to the URL of the asset details page in CentraSite Business UI.
<code>\${entity.BUIBaseURL}</code>	A deep link to the URL of the CentraSite Business UI.
<code>\${entity.description}</code>	The object's description.
	Note: User objects do not have a Description attribute.
<code>\${user.displayname}</code>	The display name of the user who triggered the policy.
<code>\${entity.key}</code>	The object's key (that is, the UUID that uniquely identifies the object within the registry).
<code>\${entity.name}</code>	The object's name (in the user's locale).
<code>\${entity.owner}</code>	The name of the user who owns the object against which the policy is enforced.
<code>\${entity.type}</code>	The type of object against which the policy enforced.
<code>\${entity.state}</code>	The state of the object against which the policy is enforced. If the object is an Asset, Policy, or Lifecycle Model, this action inserts the object's current lifecycle state. For all the other object types, this token is ignored.
<code>\${entity.URL}</code>	The URL for the object on which the policy is enforced. (This is the URL that opens the object in CentraSite Control.)
<code>\${entity.version}</code>	The object's user-assigned version identifier.
<code>\${event.type}</code>	The type of event that triggered the policy.
<code>\${from.state}</code>	The state from which the object is being switched (if the policy is executing on a Pre-State Change or Post-State Change event).
<code>\${target.state}</code>	The state to which the object is being switched (if the policy is executing on a Pre-State Change or Post-State Change event).
<code>\${user.locale}</code>	The locale of the user who triggered the policy.
<code>\${user.name}</code>	The name of the user who triggered the policy.
<code>\${user.organization}</code>	The name of the organization to which the user who triggered the policy belongs.

This token...	Inserts the following information into the parameter value at execution time...
<code>\${policycontext.consumer.name}</code>	The name of the consumer requesting access to the API.
<code>\${policycontext.apikey}</code>	The access token to grant access to the API.
<code>\${request.date}</code>	The date on which the API access token was requested.
<code>\${apikey.expirationdate}</code>	The date on which the API access token expires.
<code>\${request.application.name}</code>	The name of the consumer application requesting access to the API.
<code>\${request.application.description}</code>	A short description about the consumer application requesting access to the API.
<code>\${request.reason}</code>	A reason for requesting access to the API.
<code>\${consumed.asset.name}</code>	The name of the API that is being accessed by the consumer application.

Example

```
User ${entity.owner} has added the following asset to the catalog:  
Name: ${entity.name} Description: ${entity.description}
```

Send Email Notification to Watchers

Sends an email notification to the watchers for an asset who are specific users asked to be notified for any modifications on that particular asset.

Note:

This action is applicable to CentraSite Business UI.

Event Scope

Post-Update

Post-Delete

OnTrigger

Object Scope

Assets

Input Parameters

Email Template (Email Template). Specifies the template that is to be used to generate the body of the email message.

Note:

You can use the predefined template, `NotifyUsersOnUpdate.html`, as your email template if you do not want to create an email template of your own.

Format (String). Specifies whether the mail message is formatted as HTML or plain text.

Set Attribute Value

Assigns a value to a specified attribute in an organization, user, or asset.

Event Scope

Pre-Create

Post-Create

Pre-State Change

Post-State Change

OnTrigger

OnConsumerRegistration

Object Scope

Organization

User

Assets

Input Parameters

Attribute Name (String/Non-String). The name of the attribute that you want to set.

Note: Attribute Name must be a non-arrayed String/Non-String attribute.

Attribute Value (String). (Array). An array of regular expression String values for the attribute identified by the parameter Attribute Name.

Set Business UI Profile Permissions

Assigns instance-level permissions to an asset's profiles in CentraSite Business UI.

Event Scope

Pre-Update

Object Scope

Assets

Set Consumer Permission

Assigns permission settings to the users and groups who are identified by a consumer-registration request.

The behavior of this action with respect to specific asset profiles depends on the policy's object scope.

- If you use this action in a policy that applies to multiple asset types, you can set only the asset's top-level View, Modify, or Full permissions. Consumers do not receive View or Modify permission on the individual profiles associated with the asset. You have to assign permissions to the asset's individual profiles manually.
- If you use this action in a policy that applies to one (and only one) type of asset, you can set the asset's top-level View, Modify, or Full permissions and also the View or Modify permissions on its individual profiles.

The permission settings you specify in this action will either replace or be merged with the asset's existing settings, depending on how you set the `Remove Existing Permission` parameter.

If you set `Remove Existing Permission` to true, the permission settings specified in the action *completely replace* the asset's current settings. That is, the asset's previous instance-level settings are completely cleared and the permissions specified by the action are set.

For example if an asset's initial permission settings are as follows:

```
USER A    Full
USER B    Full
```

And you specify the following permissions (with `Remove Existing Permission` set to true):

```
USER A    Full
GROUP X   Modify
```

The resulting permissions on the asset is:

```
USER A    Full
GROUP X   Modify
```

If you set `Remove Existing Permission` to false, the permission settings specified by this action are added to the asset's current settings. So, for example, if an asset has the following permission settings:

```
USER A    Full
USER B    View
```

And you specify the following permissions (with `Remove Existing Permission` set to false):

```
USER A    Modify
USER B    Full
GROUP X   Modify
```

The resulting permissions on the asset is:

```

USER A    Full
USER B    Full
GROUP X   Modify

```

Note:

The instance-level permissions that this action assigns to a user does not affect any role-based permissions that the user might already have. For example, if user ABC has `Manage Assets` permission for an organization, and that user also happens to be a member of a group to which this action assigns instance-level permissions, user ABC's `Manage Assets` permission will override the permission settings that this action assigns to him or her.

Event Scope

`OnConsumerRegistration`

Object Scope

`Assets`

Input Parameters

<code>Consumer Asset Profile Permission</code>	(Object). The instance-level permissions that are to be assigned to the users and groups (specified in the consumer registration request) for the requested asset.
<code>Remove existing permission</code>	(Boolean). Specifies whether the permission settings in the <code>Consumer Asset Profile Permission</code> parameter replace the existing permission settings or whether they are combined with the existing settings.

Set Instance and Profile Permissions

Sets instance-level permissions on an asset. You can use this action to set top-level `View`, `Modify`, or `Full` permissions on an entire asset and to set `View` or `Modify` permissions on individual profiles within an asset.

Note:

You use this action to set permissions on assets only. To set permissions on policies, you must use the [Set Permissions](#) action. If you want to assign asset permissions to consumers during the consumer registration process, use the [Set Consumer Permission](#) action.

Be aware that the behavior of this action varies depending on the policy's object scope.

- If you use this action in a policy that applies to multiple asset types, you can only use it to set the asset's top-level `View`, `Modify`, or `Full` permissions. Users do not receive `View` or `Modify` permission on the individual profiles associated with the asset. You have to assign permissions to the asset's individual profiles manually.
- If you use this action in a policy that applies to one (and only one) type of asset, you can use it to set the asset's top-level `View`, `Modify`, or `Full` permissions and also the `View` or `Modify` permissions on its individual profiles.

The permission settings you specify in this action will either replace or be merged with the asset's existing settings, depending on how you set the `Remove Existing Permission` parameter.

If you set `Remove Existing Permission` to true, the permission settings specified in the action *completely replace* the asset's current settings. That is, the asset's previous instance-level settings are completely cleared and the permissions specified by the action are set.

For example if an asset's initial permission settings are as follows:

```
USER A    Full
USER B    Full
```

And you specify the following permissions (with `Remove Existing Permission` set to true):

```
USER A    Full
GROUP X   Modify
```

The resulting permissions on the asset is:

```
USER A    Full
GROUP X   Modify
```

If you set `Remove Existing Permission` to false, the permission settings specified by this action are added to the asset's current settings. So, for example, if an asset has the following permission settings:

```
USER A    Full
USER B    View
```

And you specify the following permissions (with `Remove Existing Permission` set to false):

```
USER A    Modify
USER B    Full
GROUP X   Modify
```

The resulting permissions on the asset is:

```
USER A    Full
USER B    Full
GROUP X   Modify
```

Note:

The instance-level permissions that this action assigns to a user does not affect any role-based permissions that the user might already have. For example, if user ABC has `Manage Assets` permission for an organization, and that user also happens to be a member of a group to which this action assigns instance-level permissions, user ABC's `Manage Assets` permission will override the permission settings that this action assigns to him or her.

Event Scope

Pre-Create

Post-Create

Pre-Update

Pre-State Change

Post-State Change

OnTrigger

Object Scope

Assets

Input Parameters

User/Group Asset Permission	(Object). (Array). An array of permission settings. Each setting in the array identifies one individual user or one group and specifies the permissions for that user or group. If you specify multiple groups in this array and a user is a member of more than one group, the user will receive the permissions of all those groups combined. For example, if you assign Modify permission to Group A and Full permissions to Group B, users that are members of both groups will get Full permission on the object.
Remove existing permission	(Boolean). Specifies whether the permission settings in the parameters User/Group Asset Permission, Propagate permissions to dependent objects and replace the existing permission settings or whether they are combined with the existing settings.
Propagate permissions to dependent objects	(Boolean). Specifies whether the access permissions defined for the asset instance is automatically propagated to all dependent objects. For example, a Service asset can refer to a WSDL which in turn can refer to one or more XML Schema assets, and when you set this parameter to yes, changes in the access permissions in the Service asset is propagated to all of these dependent assets.
Propagate profile permissions	(Boolean). Specifies whether the profile permissions defined for the asset instance will be automatically propagated to all dependent assets of the same type. The restriction concerning the asset type arises because different asset types can have different sets of profiles. The use of this parameter is restricted to the following asset types: <ul style="list-style-type: none"> ■ Service ■ XML schema ■ REST Service ■ OData Service

Set Permission for Asset's Objects

Grants View, Modify, or Full permissions to assets's objects (Operation, Interface, and Binding Concept).

Event Scope

Post-Create

Object Scope

Interface

Operation

REST Method

ApplinX External Web Operation

Input Parameters

None.

Set Permissions

Grants View, Modify, or Full permissions to specified users (or to groups of users) for a policy.

Note:

You use this action to set permissions on policy objects. To set permissions on catalog assets, you must use [Set Instance and Profile Permissions](#).

Be aware that the permission settings you specify in the action will either replace or be merged with the object's existing settings, depending on how you set the `Remove Existing Permission` parameter.

If you set `Remove Existing Permission` to true, the permission settings specified in the action completely replace the object's current settings. That is, the action will clear the object's existing permission settings and replace them with the permissions you specify.

For example if a policy's initial permission settings were as follows:

```
USER A      Full
USER B      Full
GROUP ABC   Full
```

And you were to specify the following permissions with `Remove Existing Permission` set to true:

```
USER A      Full
GROUP X     Modify
```

The resulting permissions on the asset would be:

```
USER A      Full
GROUP X     Modify
```

If you set `Remove Existing Permission` to `false`, the permission settings specified in the action are *added to* the object's current settings. That is, the action will merge the new permission settings with the object's existing settings. For example, if an asset had the following permission settings:

```
USER A      Full
USER B      View
GROUP ABC   View
```

And you were to specify the following permissions with `Remove Existing Permission` set to `false`:

```
USER A      Modify
USER B      Full
GROUP X     Modify
```

The resulting permissions on the asset is:

```
USER A      Full
USER B      Full
GROUP X     Modify
GROUP ABC   View
```

Note:

The instance-level permissions that this action assigns to a user does not affect any role-based permissions that the user might already have. For example, if user ABC has `Manage Policies` permission for an organization and that user also happens to be a member of a group to which this action assigns instance-level permissions, user ABC's `Manage Policies` permission will override the permission settings that this action assigns to him or her.

Event Scope

Post-Create

Pre-State Change

Post-State Change

OnTrigger

Object Scope

Policy

Assets

Input Parameters

`User/Group Permission` (Object). (Array). An array of permission settings. Each setting in the array identifies one individual user or one group and specifies the permissions for that user or group.

If you specify multiple groups in this array and a user is a member of more than one group, the user will receive the permissions of all those groups combined. For example, if you assign `Modify` permission to Group A and

	Full permissions to Group B, users that are members of both groups will get Full permissions on the object.
Remove existing permission	(Boolean). Specifies whether the permission settings in the <code>Users</code> and <code>Groups</code> parameter replace the existing permission settings or whether they are combined with the existing settings.
Propagate permissions to dependent objects	(Boolean). Specifies whether the access permissions defined for the asset instance is automatically propagated to all dependent objects. For example, a Service asset can refer to a WSDL which in turn can refer to one or more XML Schema assets, and when you set this parameter to <code>yes</code> , changes in the access permissions in the Service asset is propagated to all of these dependent assets.

Set Profile Permissions

This action sets an asset's profile permissions for the users or groups specified without setting the asset's instance level permissions.

The users or groups specified in the parameter should have view or modify instance level permission on the asset.

Event Scope

- Post-Create
- Pre-State Change
- Post-State Change
- OnTrigger

Object Scope

- Assets

Input Parameters

User/Group Permission	(Object). (Array). An array of permission settings. Each setting in the array identifies one individual user or one group, the specified profile and the view/modify permissions for that user or group for the profile.
Remove existing permission	(Boolean). Specifies whether the permission settings in the <code>User/Group Permission</code> parameter replace the existing permission settings or whether they are combined with the existing settings.

Set State

Initiates a lifecycle state change for a lifecycle model, policy, asset or Process object.

When you use this action, be aware that:

- The state change performed by this action triggers Pre-State Change or Post-State Change policies if such policies exist for the specified state change.
- When CentraSite executes this action at enforcement time, it attempts to change the target object to the state you have specified. If this state is not a valid transition from the object's current state, the action fails.
- If the target object is already in the specified state at enforcement time, this action does nothing. It does not initiate a state change. It simply exits and returns a successful completion code (that is, this condition is not considered an error).

Event Scope

Post-State Change

OnTrigger

OnConsumerRegistration

Object Scope

Policy

Lifecycle Model

Assets

Input Parameters

Change State To (String). The value to which you want to set the object's state.

Set View Permission for Service and Service Related Object to Everyone Group

Grants the View permission on a given service to the Everyone group. When permission is given to Everyone, all users, including guests, are able to view the service and its related interface, operation and binding objects. This policy action enables UDDIv2 clients to access the service without providing an authtoken.

This action is included in the *UDDIv2 Inquiry Policy* policy that is installed with CentraSite. This policy executes when a service or virtual service is created. This policy is disabled by default.

Event Scope

Post-Create

OnTrigger

Object Scope

Assets

Input Parameters

None.

UnClassify

Removes specified taxonomy categories from an object.

You can use this action to unclassify an object generally or specifically. If you want to unclassify an object by removing from it all categories for an entire taxonomy, use the `Taxonomies` parameter to specify the taxonomy name. If you want to unclassify an object by removing just one particular category from its classification attributes, you use the `Categories` parameter to specify a specific category name. Both parameters can be used in the same action.

This action is executed against all classification attributes in the target object.

If the target object is not classified by any of the taxonomies or classifiers specified in the `Taxonomies` or `Categories` parameter, the action simply exits and returns a successful completion code. This condition is not considered to be an error.

Event Scope

Post-State Change

OnTrigger

OnConsumerRegistration

Object Scope

This action can be enforced on any object type that the policy engine supports.

Input Parameters

<code>Taxonomies</code>	(String). (Array). The names of the taxonomies whose categories are to be removed from the target object.
<code>Categories</code>	(String). (Array). The names of specific categories that are to be removed from the target object.

Unpublish from API Portal

Removes specified API metadata from the API Portal repository.

Event Scope

Pre-Delete

Post-Delete

Pre-State Change

Post-State Change

OnTrigger

Object Scope

Service

Virtual Service

XML Service

Virtual XML Service

REST Service

Virtual REST Service

IS Service Interface

OData Service

Virtual OData Service

Input Parameters

API Portal

(Optional). (String). (Array). The name of the API Portal from that API-repository the API metadata would be removed.

Note:

However, if this action is to be executed in a different event other than OnTrigger, for example, Pre-State Change or Post-State Change, that is not provided by default, you *must* specify a value for this field.

Validate Attribute Value

Validates the value of a specified attribute in an organization, user, or asset against a list of allowed values.

Event Scope

Pre-Update

Pre-State Change

Pre-Delete

OnTrigger

OnConsumerRegistration

Object Scope

Organization

User

Assets

Input Parameters

Attribute Name The name of the attribute that you want this action to test. The attribute's data type can be Boolean, Date and Time, Duration, Email, IP Address, Multiline String, Number, and URL or URI.

Note:Attribute Name must be a non-arrayed attribute.

Possible Attribute Value (String). (Array). An array of regular expression String values. If the value of the attribute specified in **Attribute Name** matches any entry in **Possible Attribute Values**, the action succeeds.

The regular expressions you specify in **Possible Attribute Values** must support the regular expression specification for Java.

The data types of possible attribute values can be Boolean, Date and Time, Duration, Email, IP Address, Multiline String, Number, and URL or URI.

You can include substitution tokens in this parameter to incorporate data from the target object on which the policy is acting. For a list of the allowed tokens, see the list of Substitution Tokens shown in the [Send Email Notification](#) action.

Validate Classification

Checks whether an object is classified by a given taxonomy or taxonomy category. This action examines all classification attributes in the target object.

If you just want to check that the target object has been classified by a given taxonomy, simply specify the taxonomy in the **Taxonomies** parameter. Leave the **Categories** parameter empty. The action will succeed if the object is classified by *any* category in the taxonomy (that is, the action succeeds if the object includes at least one **Classification** attribute whose value represents a category that belongs the specified taxonomy).

If you want to check that the target object has been classified by a specific category in a taxonomy, specify the exact category in the **Categories** parameter. Leave the **Taxonomies** parameter empty. The action will succeed only if the object has been classified by the exact category you specify (that is, the object includes at least one **Classification** attribute whose value is set to that specific category).

If you specify multiple taxonomies and categories in the **Taxonomies** and **Categories** parameters, be aware that action will succeeds if the target object is classified according to *any* taxonomy specified in the **Taxonomies** parameter or *any* category specified in the **Categories** parameter. If you need to verify that an object has been classified by several different taxonomies or categories, you must test for each required taxonomy or category using a separate **Validate Classification** action.

Event Scope

Pre-Update

Pre-Delete

Pre-State Change

OnTrigger

OnConsumerRegistration

Object Scope

This action can be enforced on any object type that the policy engine supports.

Input Parameters

Taxonomies	(String). (Array). The names of the taxonomies by which the object must be classified.
Categories	(String). (Array). The names of specific taxonomy nodes by which the target object must be classified.

Validate Description

Validates the description of an object against a given pattern string.

Event Scope

Pre-Create

Pre-Update

Pre-State Change

OnTrigger

Object Scope

This action can be enforced on any object type that the policy engine supports.

Input Parameters

Allowed Description Pattern	(String). Specifies a regular expression that the description must satisfy. The regular expressions you specify in Allowed Description Pattern must support the regular expression specification for Java.
-----------------------------	--

The regular expression can include substitution tokens to incorporate data from the target object on which the policy is acting. For a list of the allowed tokens, see the list of Substitution Tokens shown in the [Send Email Notification](#) action.

Validate Lifecycle Model Activation

Verifies that a lifecycle model is ready to be activated by checking that the following conditions exist for the lifecycle model:

- That the lifecycle model is associated with at least one object type.
- That the object types associated with the lifecycle model are not already assigned to an active lifecycle model in your organization. (This check ensures that, within your organization, each object type is associated with no more than one lifecycle model.)

The action does not succeed unless both conditions are satisfied.

You should include this action in any policy that is triggered by a lifecycle state change that subsequently activates the lifecycle model. Executing this action before the state change occurs ensures that the state change (and subsequent activation) does not occur unless the lifecycle model is capable of being activated.

This action is executed by the default *Validate Lifecycle Activation* policy that is installed with CentraSite. The Validate Lifecycle Activation policy executes on the Pre-State Change that occurs when a lifecycle model switches to the Productive lifecycle state. The Validate Lifecycle Activation action in this policy ensures that a lifecycle model is not switched to the Productive state (and consequently, activated) unless the model has been properly associated with one or more object types.

Event Scope

Pre-State Change

Object Scope

Lifecycle Model

Input Parameters

None.

Validate Name

Validates the name of an object against a given pattern string.

Event Scope

Pre-Create

Pre-Update

Pre-State Change

OnTrigger

Object Scope

This action can be enforced on any object type that the policy engine supports.

Input Parameters

Allowed Name Pattern (String). Specifies a regular expression that the object name must satisfy. The regular expressions you specify in **Allowed Name Pattern** must support the regular expression specification for Java.

The regular expression can include substitution tokens to incorporate data from the target object on which the policy is acting. For a list of the allowed tokens, see the list of Substitution Tokens shown in the [Send Email Notification](#) action.

Validate Namespace

Checks that the `targetnamespace` attribute in a web service or XML schema matches one of the valid namespaces in a given list.

Event Scope

Pre-Create

Pre-Update

Pre-State Change

OnTrigger

Object Scope

Service

Virtual Service

XML Service

Virtual XML Service

REST Service

Virtual REST Service

IS Service Interface

OData Service

Virtual OData Service

XML Schema

Event Type

IS Type Definition

IS Specification

Input Parameters

Allowed Namespaces (String). (Array). An array of regular expressions representing the valid namespaces. For this action to succeed, the value of the `targetnamespace` attribute in the service WSDL or XML schema must satisfy one of the regular expressions in the array.

The regular expressions you specify in `Allowed Namespaces` must support the regular expression specification for Java.

The regular expression can include substitution tokens to incorporate data from the target object on which the policy is acting. For a list of the allowed tokens, see the list of Substitution Tokens shown in the [Send Email Notification](#) action.

Validate Policy Activation

Verifies that a policy is ready to be activated by checking that the following conditions exist for the policy:

- That all of the required parameters in the policy's action list have been set.
- That all of the actions in the action list are supported by the policy's specified scope. That is, the policy does not contain any action whose scope includes an object type or event type that is outside the scope of the policy itself.
- That a policy that contains one or more WS-I actions contains *only* WS-I actions.
- That a policy that executes on a Pre-State Change or Post-State Change specifies the lifecycle states that triggers the policy.
- Whether a previous version of the policy is already active, and if so, it verifies that the policy can be switched to a state in which it is retired or superseded.

The action does not succeed unless all conditions are satisfied.

You should include this action in any policy that is triggered by a lifecycle state change that subsequently activates the policy. Executing this action before the state change occurs ensures that state change (and subsequent activation) does not occur unless the policy is capable of being activated.

This action is executed by the default *Validate Policy Activation* policy that is installed with CentraSite. The Validate Policy Activation policy executes on the Pre-State Change event that occurs when a policy switches to the Productive lifecycle state. The Validate Policy Activation action in this policy ensures that a policy is not switched to the Productive state (and consequently activated) unless the policy's action parameters have been set.

Event Scope

Pre-State Change

Object Scope

Policy

Input Parameters

None.

Validate Policy Deactivation

Verifies that a policy is not currently in-progress (that is, undergoing execution) and can therefore be successfully deactivated. If the policy is in-progress when this action is executed, this action fails.

You should include this action in any policy that is triggered by a lifecycle state change that subsequently deactivates the policy. Executing this action before the state change occurs helps ensure that the stage change (and subsequent policy deactivation) does not take place if the target policy is in-progress.

Note:

A policy that initiates an approval workflow is considered to be in-progress until the required approvals are obtained for the workflow. Therefore, if the Validate Policy Deactivation action is triggered for a policy that is associated with one or more pending approval workflows, the action fails.

This action is executed by the default *Validate Policy Deactivation* policy that is installed with CentraSite. The Validate Policy Deactivation policy executes on the Pre-State Change event that occurs when a policy switches to the Revising or Retired state. The Validate Policy Deactivation action in this policy ensures that a policy is not switched to the Revising or Retired state (and consequently, deactivated) while it is undergoing execution.

Event Scope

Pre-State Change

Object Scope

Policy

Input Parameters

None.

Validate Service Binding

Checks that a web service supports the specified bindings.

Event Scope

Pre-Create

Pre-Update

Pre-State Change

OnTrigger

Object Scope

Service

Virtual Service

XML Service

Virtual XML Service

REST Service

Virtual REST Service

IS Service Interface

OData Service

Virtual OData Service

Input Parameters

Binding Types (String). (Array). An array containing the list of binding types that the web service must support. The action will succeed only if the web service supports all of the bindings specified in **Binding Types**.

Validate State

Validates the current state of a lifecycle model, policy, or asset against a given list of states.

Event Scope

Pre-Delete

OnTrigger

OnConsumerRegistration

Object Scope

Lifecycle Model

Policy

Assets

Input Parameters

Allowed States (String). (Array). An array that specifies the states for which you want the target object checked. If the state of the object matches any entry specified in **Allowed States**, the action succeeds.

Validate WSDL Size

Checks the size of the WSDL document associated with a web service to ensure it falls within a specified range.

Event Scope

Pre-Create

Pre-Update

Pre-State Change

OnTrigger

Object Scope

Service

Virtual Service

XML Service

Virtual XML Service

REST Service

Virtual REST Service

IS Service Interface

OData Service

Virtual OData Service

Input Parameters

WSDL Size	(Number). The size limit (expressed in the units specified by the Size Unit parameter).
Comparator	(String). A relational operator that specifies how the size of the WSDL document is to be compared to the value in WSDL Size .
Size Unit	(String). The units in which WSDL Size is expressed. Valid values are KB (for Kilobytes) or MB (for Megabytes).

Verify Required Attributes

Ensures that all of the required attributes of an asset have a valid value.

Event Scope

Pre-Create

Pre-Update

Object Scope

Assets

Input Parameters

None.

webMethods REST Publish

Creates a REST service from the published IS Service Interface object.

The action is included in the *webMethods REST Publish* policy that is installed with CentraSite. This policy automatically executes when the webMethods Designer publishes an IS Service Interface object.

Important:

This IS Service Interface object should be classified under the concept called WMAssetType -> Integration Server Asset -> TypeOfIntegrationServiceInterface -> REST Service.

Event Scope

Post-Create

Pre-Update

Object Scope

IS Service Interface

Input Parameters

None.

Configuring Email Notifications

Certain policy actions, such as the Send Email Notification action and the approval actions, send email messages to users when specified events occur. For example, you might use the Send Email Notification action to alert a certain group of administrators when an asset switches to a particular state. Or, you might issue an email alert to certain users when an approval request is rejected.

Setting the Email-Related Parameters in an Email Notification Action

Actions that send email notifications to users (such as the Send Email Notification action) require you to specify the following input parameters:

Set this parameter...	To specify...
Users — AND/OR —	The users to whom the email message is to be sent. You can use the <code>Users</code> parameter to specify individual users, the <code>Groups</code> parameter to specify groups of users or both.
Groups	
Subject	The text that will appear on the subject line of the email.
Use Email Template — OR —	The body of the message. You can specify the body of the email by typing a message directly into the <code>Custom Message</code> parameter or by using an email template.
Custom Message	

You can include *substitution tokens* in the subject line and/or the body of the email message. Substitution tokens enable you to incorporate run-time information into the email. For example, you can use the `${user.name}` token to insert (into the email message or the subject line) the name of the user who triggered the policy. For a complete list of the supported substitution tokens, see the description of the action Send Email Notification.

Using Custom Messages in an Email Notification Action

One way to specify the body of the email message is to simply type the message directly into the `Custom Message` parameter. If you specify the body of the message in this way, you must set the `Format` parameter to indicate whether the message is to be sent as plain text or as an HTML document. When you use HTML format, you must enclose the message text in the `<html>` and `<body>` tags.

Example of a Plain Text Message

```
Virtual Service ${entity.name} has been placed in production by ${user.name}.
To view the virtual service, go to: ${entity.URL}
```

Example of an HTML Message

```
<html>
  <body>
    <p>Virtual Service <b>${entity.name}</b> has been placed in production by
    <b>${user.name}</b>.</p>
    <p>To view the virtual service, go to:
    <a href="${entity.URL}">${entity.URL}</a></p>
  </body>
</html>
```

Using Email Templates with Policy Actions

To generate the body of an email message from an email template, the email template must exist in CentraSite's repository. If the template does not already exist, you must create it and upload it to the repository using the `EmailTemplateManager` command line utility. Several predefined templates are available for you to use with various policy actions.

The following describes how to create an email template and upload it to the repository. It also describes how to edit a template, download a template, delete a template and obtain a list of the templates that already exist in the repository:

Note:

To work with email templates, you must have access to the command line on the machine where CentraSite is installed. Additionally, if you want to upload templates or delete templates, you must have a CentraSite user account that belongs to the CentraSite Administrator role. To view the list of email templates or to download a template, you simply need a CentraSite user account. (In other words, your CentraSite account does not require any explicit permissions. All CentraSite users have View permission on email templates.)

Managing Email Templates through Command Line

This section describes operations you can perform to manage email templates through Command Line Interface.

Creating a Custom Email Template

An email template is a text file that contains an HTML document. Your HTML document should include the `<html>` and `<body>` tags as shown in the example below. The inclusion of a `<head>` tag is optional. CentraSite does not require this tag in an email template.

Example:

```
<html>
<body>
  <p>Virtual Service <b>${entity.name}</b> has been placed in production by
  <b>${user.name}</b>.</p>
  <p>To view the virtual service, go to:
  <a href="${entity.}">${entity.URL}</a></p>
</body>
</html>
```

Adding a Custom Email Template

Pre-requisites:

To add a custom email template through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `ConfigEmailTemplates` for this purpose.

> To add a custom email template

- Run the command `ConfigEmailTemplates`.

The syntax is of the format `C:\SoftwareAG\CentraSite\utilities>ConfigEmailTemplates.cmd -dbuser <USERNAME> -dbpassword <PASSWORD> [-dburl <CENTRASITE-URL>] -t <TEMPLATE-FILE>`

The input parameters are:

Parameter	Description
<code>-dburl</code>	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
<code>-dbuser</code>	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
<code>-dbpassword</code>	The password for the registered CentraSite user identified by the parameter <code>-dbuser</code> .
<code>-t</code>	The absolute or relative path to the template file. If relative, the path should be relative to the location from where the command is executed.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>ConfigEmailTemplates.cmd -dburl
http://localhost:53307/CentraSite/CentraSite -dbuser Administrator -dbpassword manage
-t c:\temp\EmailTempConfig.xml
```

Viewing the Email Templates List

Pre-requisites:

To view the list of available email templates through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `ConfigEmailTemplates` for this purpose.

> To view the list of email templates

- Run the command `ConfigEmailTemplates`.

The syntax is of the format `C:\SoftwareAG\CentraSite\utilities>ConfigEmailTemplates.cmd -dbuser <USERNAME> -dbpassword <PASSWORD> [-dburl <CENTRASITE-URL>] -list`

The input parameters are:

Parameter	Description
-dburl	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
-dbuser	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
-dbpassword	The password for the registered CentraSite user identified by the parameter <code>-dbuser</code> .
-list	A list of the available email templates.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>ConfigEmailTemplates.cmd -dburl
http://localhost:53307/CentraSite/CentraSite -dbuser Administrator -dbpassword manage
-list
```

Downloading Email Template

Pre-requisites:

To download an email template through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `ConfigEmailTemplates` for this purpose.

➤ To download an email template

- Run the command `ConfigEmailTemplates`.

The syntax is of the format `C:\SoftwareAG\CentraSite\utilities>ConfigEmailTemplates.cmd -dbuser <USERNAME> -dbpassword <PASSWORD> [-dburl <CENTRASITE-URL>] -download <EMAIL-TEMPLATE> -tolocation <LOCATION>`

The input parameters are:

Parameter	Description
-dburl	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
-dbuser	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
-dbpassword	The password for the registered CentraSite user identified by the parameter <code>-dbuser</code> .

Parameter	Description
-download	Name of an email template you want to download. If the template name contains white spaces, enclose the name with "".
-tolocation	A location to save the downloaded email template.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>ConfigEmailTemplates.cmd -dburl
http://localhost:53307/CentraSite/CentraSite -dbuser Administrator -dbpassword manage
-download CustomPendingNotification.html
```

Deleting an Email Template

Pre-requisites:

To delete an email template through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `ConfigEmailTemplates` for this purpose.

> To delete an email template

- Run the command `ConfigEmailTemplates`.

The syntax is of the format `C:\SoftwareAG\CentraSite\utilities>ConfigEmailTemplates.cmd -dbuser <USERNAME> -dbpassword <PASSWORD> [-dburl <CENTRASITE-URL>] -delete <EMAIL-TEMPLATE>`

The input parameters are:

Parameter	Description
-dburl	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
-dbuser	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
-dbpassword	The password for the registered CentraSite user identified by the parameter <code>-dbuser</code> .
-delete	Name of an email template you want to delete. If the template name contains white spaces, enclose the name with "".

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>ConfigEmailTemplates.cmd -dburl
http://localhost:53307/CentraSite/CentraSite -dbuser Administrator -dbpassword manage
-delete CustomPendingNotification.html
```

Managing Email Templates through Java Class

This section describes operations you can perform to manage email templates through Java class.

Script File for EmailTemplateManager Utility

The `EmailTemplateManager` is a Java class whose `main()` method executes when you execute the `EmailTemplateManager` from the command line. To ensure that the `CLASSPATH` and other environment variables are set properly when you execute this utility, you must create a script file that calls the `EmailTemplateManager` as described below.

Creating the EmailTemplateManager Script File for Windows (Batch file)

Create a script file that looks as follows if `CentraSite` is running under Windows.

```
@echo off
set JAVAEXE=fullPathToJava.exe set REDIST=CentraSiteHomeDirectory\redist
set BASEDIR=%~dp0
cd /d %REDIST%

REM build CLASSPATH with all files from jar directory
set LOCAL_CLASSPATH=
for %%I in (*.jar) do call
  "CentraSiteHomeDirectory\bin\cfg\lcp.cmd" %%I
%JAVAEXE% -cp %LOCAL_CLASSPATH% com.centrasite.util.EmailTemplateManager %*
cd /d %BASEDIR%
```

Example:

```
@echo off
REM
REM Run Email Template Manager Utility
REM
set JAVAEXE=D:\software\java\jdk1.5.0_12\bin\java
set REDIST=C:\SoftwareAG\CentraSite\redist
set BASEDIR=%~dp0
cd /d %REDIST%
REM build CLASSPATH with all files from jar directory
set LOCAL_CLASSPATH=
for %%I in (*.jar) do call "C:\SoftwareAG\CentraSite\bin\cfg\lcp.cmd" %%I
%JAVAEXE% -cp %LOCAL_CLASSPATH% com.centrasite.util.EmailTemplateManager %*
cd /d %BASEDIR%
```

Creating the EmailTemplateManager Script File for Unix (C-shell script)

Create a script file that looks as follows if `CentraSite` is running under Unix.

```
set javaexe="fullPathToJava.exe"
set redist="fullPathToJava.exe/redist"
set mainjar="CentraSiteUtils.jar"
```

```

set delim='\:'
cd "$redist"
set cl=""
foreach j ( `ls *.jar` )
  if ($cl != "") set cl=${cl}${delim}
  set cl=${cl}${j}
end
setenv CLASSPATH ${mainjar}${delim}${cl}
$javaexe com.centrasite.util.EmailTemplateManager $*

```

Example:

```

#!/bin/csh
#
# Run Email Template Manager Utility
#
set javaexe="/.../softwareag/cjp/v16/bin/java"
set redist="/.../softwareag/CentraSite/redist"
set mainjar="CentraSiteUtils.jar"
set delim='\:'
# build CLASSPATH with all files from jar directory
cd "$redist"
set cl=""
foreach j ( `ls *.jar` )
  if ($cl != "") set cl=${cl}${delim}
  set cl=${cl}${j}
end
setenv CLASSPATH ${mainjar}${delim}${cl}
$javaexe com.centrasite.util.EmailTemplateManager $*

```

Executing the EmailTemplateManager Script File

To run the `EmailTemplateManager`, execute your script file on the machine where `CentraSite` is installed.

Note:

`CentraSite` must be running when you execute the script file.

When you execute your script file, you must include input parameters on the command line to specify what you want the `EmailTemplateManager` to do. You must also include parameters to specify your `CentraSite` user ID and password. For example, to delete a template from the directory, you would run your script as follows:

```
yourScriptFile -delete templateName -dbuser yourCSUserID -dbpassword yourPassword
```

Example:

```
myScript -delete myEmailTemplate.html -dbuser jcallen -dbpassword j45Hk19a
```

The `EmailTemplateManager` utility assumes that the `CentraSite` registry/repository is running at `http://localhost:53307` (that is, it assumes that it is installed on the same machine as the `EmailTemplateManager` utility). If you installed the registry/repository component on a different machine than the `CentraSite` Application Server Tier (CAST) or if you configured the registry/repository to run on a different port than 53307, you will need to use the `-dburl` parameter to specify the address of the registry/repository when you run your script.

```
myScript -delete myEmailTemplate.html -dburl http://rubicon:53307/CentraSite/CentraSite
-dbuser jcallen -dbpassword j45Hk19a
```

Adding a Custom Email Template

Pre-requisites:

To add a custom email template through Java class, you must have the CentraSite Administrator role.

> To add a custom email template

1. Create an email template in the machine where CentraSite is installed.
2. Create a script file for the EmailTemplateManager utility.
3. Execute the script file with the following parameters:

```
yourScriptFile -ttemplateFile -dbuser yourCSUserID -dbpassword yourPassword
```

Example:

```
myScript -t d:\myDirectory\myEmailTemplate.html -dbuser jcallen -dbpassword j45Hk19a
```

Viewing the Email Templates List

Pre-requisites:

To view the list of available email template through Java class, you must have the CentraSite Administrator role.

> To view the list of email templates

1. Create a script file for the EmailTemplateManager utility.
2. Execute the script file with the following parameters:

```
yourScriptFile -list -dbuser yourCSUserID -dbpassword yourPassword
```

Example:

```
myScript -list -dbuser jcallen -dbpassword j45Hk19a
```

Downloading an Email Template

Pre-requisites:

To download an email template through Java class, you must have the CentraSite Administrator role.

> To download an email template

1. Create a script file for the EmailTemplateManager utility.
2. Execute the script file with the following parameters:

```
yourScriptFile -download templateName -tolocation targetDirectory -dbuser yourCSUserID  
-dbpassword yourPassword
```

Example:

```
myScript -download myEmailTemplate.html -tolocation d:\myDir\mySubDir -dbuser jcallen  
-dbpassword j45Hk19a
```

Deleting an Email Template**Pre-requisites:**

To delete an email template through Java class, you must have the CentraSite Administrator role.

> To delete an email template

1. Create a script file for the EmailTemplateManager utility.
2. Execute the script file with the following parameters:

```
yourScriptFile -delete templateName -dbuser yourCSUserID -dbpassword yourPassword
```

Example:

```
myScript -delete myEmailTemplate.html -dbuser jcallen -dbpassword j45Hk19a
```


11 Report Management

■ Introduction to Reports	854
■ Predefined Reports	855
■ Configuring JDBC Support for BIRT Reports	867
■ Reserved Identifiers	869
■ Managing Reports and Report Templates through CentraSite Business UI	869
■ Managing Reports and Report Templates through CentraSite Control	881
■ Managing Reports and Report Templates through Command Line Interface	888

Introduction to Reports

Note:

Beginning with version 9.9, CentraSite does not support the Generate Reports functionality in CentraSite Control. Instead, you can use the enhanced interface of CentraSite Business UI which supports generating, downloading, and scheduling reports.

Whether you need to document or present information about your objects, you can use predefined or custom reports to do so. Reports provide a way for you to view and share information derived from the objects you have created.

You can schedule, generate, update, list, and delete reports in CentraSite in the following ways:

- CentraSite Business UI
- Command Line Interface

About Report Templates and Reports

Report template contains instructions for gathering object metadata from CentraSite registry repository, and formatting it into an object-specific report. A report template could either be used as a stand-alone or associated with one or more objects for generating specific reports.

Reports mainly use CentraSite registry repository as data source. In addition to using CentraSite as data source, CentraSite supports the relational database management systems, such as, DB2, Microsoft SQL Server, Oracle and Sybase. The reporting framework has extended support that enables some reports to communicate with these SQL databases using the Java Database Connectivity (JDBC), and report data. The JDBC-compliant reports are designed to access and retrieve data from the JDBC-compliant databases. For more information about JDBC-compliant reports, see “[Configuring JDBC Support for BIRT Reports](#)” on page 867.

CentraSite includes a set of predefined reports that provide frequently used information about assets, API invocations, and events. These reports can be saved, shared with other users, and reused to monitor and track the API specific usage information and consumer specific usage information of required APIs. For information about reports that are provided with CentraSite, see “[Predefined Reports](#)” on page 855.

In addition to using the predefined reports, you can create and design your own custom reports that will contain a specific content and format. Such reports can be created, for example, using the Eclipse plug-ins delivered with CentraSite. The BIRT Report Designer creates a `.rptdesign` file, referred to as the `Template File`, and describes a predefined `rptdesign` layout for creating the custom report templates. Documentation describing how to create BIRT (Business Intelligence and Reporting Tools) reports is contained in the Eclipse online help that is delivered with the plug-ins. For general information about the BIRT technology, see the description of the BIRT project in the **Projects** section of the Eclipse site at <http://www.eclipse.org/birt/>.

You can create a custom report in CentraSite using the command tool `add Report`. For information on the usage of CentraSite command tool, see “[Adding Custom Report](#)” on page 889.

Sharing Reports with API Portal

You can share dynamic, read-only reports with your API Portal users without giving them access to all of your data.

API Portal users will be able to see the latest version of your report when they use the shareable link.

You can share a report with an instance of API Portal registered in CentraSite using the command tool `share Report`. For information on the usage of CentraSite command tool, see [“Sharing Report with API Portal” on page 893](#).

You can stop sharing the report at any time. To stop sharing the report, you must modify the property line `com.centrasite.report.Classifications`, and remove the value shared in the customization file, `centrasite.xml`.

After you make changes to the report configuration, you must execute the command tool `update Report`. For information on the usage of CentraSite command tool, see [“Modifying Report Details” on page 891](#).

Building Reports with Indexing and Aggregated Data

CentraSite's data store supports the use of indexing and aggregate tables that can query large databases and provide faster access to frequently-accessed data and any query. The aggregate tables reduce the amount of data that must be aggregated and sorted at runtime, and help generate reports at a much faster speed.

Note:

Consumer index and aggregation only work with the flat storage mode. Flat storage indicates an enhanced mode of data storage.

CentraSite includes the following reports that are designed to use indexing and aggregated data:

- Runtime Asset Usage (Aggregation-based)
- Runtime Consumer Usage (Aggregation-based)

For information about these reports, see [“Predefined Reports” on page 855](#).

Predefined Reports

CentraSite provides a wide range of out-of-the-box reports for monitoring, tracking and capturing API usage. Some of these report templates are available for connecting to JDBC-compliant databases, such as an Oracle.

Note:

CentraSite Community Edition license does not support runtime reports.

API Invocations (daily)

Displays the total number of runtime invocations made for an API for the current date.

JDBC Compliant: No

Report Parameters

None.

API Invocations (weekly)

Displays the total number of runtime invocations made for an API for the current week.

JDBC Compliant: No

Report Parameters

None.

API Invocations (monthly)

Displays the total number of runtime invocations made for an API for the current month.

JDBC Compliant: No

Report Parameters

None.

API Invocations for Target (daily)

Displays the total number of runtime invocations made for a specific target (API) in the last 30 days from the current date.

JDBC Compliant: Yes

Report Parameters:

Parameter	Description
Target	Listbox. Specifies the target that identifies the API whose total number of runtime invocations during a specified time period you want to view in your report. Note: The Target parameter is not available if there is only target defined in CentraSite.

API Invocations for Target (weekly)

Displays the total number of runtime invocations made for a specific target (API) in the last 4 weeks including current week from the current date.

JDBC Compliant: Yes

Report Parameters:

Parameter	Description
Target	Listbox. Specifies the target that identifies the API whose total number of runtime invocations during a specified time period you want to view in your report.
	<p>Note: The Target parameter is not available if there is only target defined in CentraSite.</p>

API Invocations for Target (monthly)

Displays the total number of runtime invocations made for a specific target (API) in the last 3 months including current month from the current date.

JDBC Compliant: Yes

Report Parameters:

Parameter	Description
Target	Listbox. Specifies the target that identifies the API whose total number of runtime invocations during a specified time period you want to view in your report.
	<p>Note: The Target parameter is not available if there is only target defined in CentraSite.</p>

Asset Change History

Displays a list of all changes that have occurred on an asset before the specified timestamp.

JDBC Compliant: No

Report Parameters:

Parameter	Description
Asset Key	String. Specifies the UDDI V3 key of the asset whose historical changes you want to view in your report. Example: uddi:207ff1cc-25c5-544c-415c-5d98ea91060c
Show Asset Changes Newer Than	Timestamp. Lists all changes that were made to this asset on any date before the specified timestamp.

Asset Changes Since Date

Displays a list of all changes that have occurred on the assets of an organization since the specified timestamp.

JDBC Compliant: No

Report Parameters:

Parameter	Description
Organization	Listbox. Specifies the organization that owns the asset(s) whose recent changes you want to view in your report. Note: The Organization parameter is not available if there is only organization defined in CentraSite.
Show Asset Changes Newer Than	Timestamp. Lists all changes that were made to the assets of this organization since the specified timestamp.

Asset Dependencies

Displays a list of all incoming and outgoing relationships for an asset.

JDBC Compliant: No

Report Parameters:

Parameter	Description
Asset Key	String. Specifies the UDDI V3 key of the asset whose dependencies you want to view in your report. Example: uddi:207ff1cc-25c5-544c-415c-5d98ea91060c

New Assets Since Date

Displays a list of all new assets that were created in an organization since the specified timestamp.

JDBC Compliant: No

Report Parameters:

Parameter	Description
Organization	Listbox. Specifies the organization whose newly created assets you want to view in your report. Note: The Organization parameter is not available if there is only organization defined in CentraSite.
Show Assets Newer Than	Timestamp. Lists all assets that were recently created in this organization since the specified timestamp.

Policy Compliance

Displays a list of policy information that were logged for an asset.

If CentraSite is configured to log only failed policy executions, the report includes only failed policy executions for the asset. On the other hand, if CentraSite is configured to log both the successful and failed policy executions, the report includes all policy executions for the asset.

JDBC Compliant: No

Report Parameters:

Parameter	Description
Asset Key	String. Specifies the UDDI V3 key of the asset whose policy information you want to view in your report. Example: uddi:207ff1cc-25c5-544c-415c-5d98ea91060c

Runtime Asset Error

Displays a list of runtime error messages that were generated for the asset(s).

JDBC Compliant: Yes

Report Parameters:

Parameter	Description
Target	Listbox. Specifies the target that identifies the asset(s) whose runtime error messages you want to view in your report. Note:

Parameter	Description
	The Target parameter is not available if there is only target defined in CentraSite.
Organization	Listbox. Specifies the organization that owns the asset(s) you want to view in your report. Note: The Organization parameter is not available if there is only organization defined in CentraSite.

Runtime Asset Quality of Service

Displays the quality of service information about runtime assets.

JDBC Compliant: Yes

Report Parameters:

Parameter	Description
Target	Listbox. Specifies the target that identifies the asset(s) whose statistical information on the quality of service, uptime, and security you want to view in your report. Note: The Target parameter is not available if there is only target defined in CentraSite.
Organization	Listbox. Specifies the organization that owns the asset(s) you want to view in your report. Note: The Organization parameter is not available if there is only organization defined in CentraSite.

Runtime Asset Usage

Displays the runtime usage information of the asset(s) over the specified time period.

JDBC Compliant: Yes

Report Parameters:

Parameter	Description
Target	Listbox. Specifies the target that identifies the asset(s) whose runtime usage information for the specified time period you want to view in your report. Note: The Target parameter is not available if there is only target defined in CentraSite.
Organization	Listbox. Specifies the organization that owns the asset(s) you want to view in your report. Note: The Organization parameter is not available if there is only organization defined in CentraSite.
From and To	Timestamp. Specifies the time period to collect the runtime usage information of asset(s).

Runtime Asset Usage (Aggregation-based)

Displays the runtime usage information of the asset(s) using aggregated data at run-time. This report using aggregated data generates the same results as the previous report, however performs queries much faster and results in a much better user experience.

JDBC Compliant: Yes

Report Parameters:

Parameter	Description
Target	Listbox. Specifies the target that identifies the asset(s) whose runtime usage information you want to view using the aggregated data at run-time. Note: The Target parameter is not available if there is only target defined in CentraSite.
Organization	Listbox. Specifies the organization that owns the asset(s) you want to view in your report. Note: The Organization parameter is not available if there is only organization defined in CentraSite.
From and To	Timestamp. Specifies the time period to collect the runtime usage information of asset(s).

Runtime Consumer Usage

Displays the consumer usage information of the asset(s) over the specified time period.

JDBC Compliant: Yes

Report Parameters:

Parameter	Description
Target	Listbox. Specifies the target that identifies the asset(s) whose per consumer usage information for the specified time period you want to view in your report. Note: The Target parameter is not available if there is only target defined in CentraSite.
Organization	Listbox. Specifies the organization that owns the asset(s) you want to view in your report. Note: The Organization parameter is not available if there is only organization defined in CentraSite.
From and To	Timestamp. Specifies the time period to collect the per consumer usage information of asset(s).

Runtime Consumer Usage (Aggregation-based)

Displays the consumer usage information of the asset(s) using aggregated data at run-time. This report using aggregated data generates the same results as the previous report, however performs queries much faster and results in a much better user experience.

JDBC Compliant: Yes

Report Parameters:

Parameter	Description
Target	Listbox. Specifies the target that identifies the asset(s) whose per consumer usage information you want to view using the aggregated data at run-time. Note: The Target parameter is not available if there is only target defined in CentraSite.
Organization	Listbox. Specifies the organization that owns the asset(s) you want to view in your report.

Parameter	Description
	<p>Note: The Organization parameter is not available if there is only organization defined in CentraSite.</p>
From and To	Timestamp. Specifies the time period to collect the per consumer usage information of asset(s).

Runtime Policy Errors

Displays a list of policy violation error messages that were generated for the asset(s).

JDBC Compliant: Yes

Report Parameters

Parameter	Description
Target	<p>Listbox. Specifies the target that identifies the asset(s) whose policy violation error messages you want to view in your report.</p> <p>Note: The Target parameter is not available if there is only target defined in CentraSite.</p>

Runtime Services

Displays the detailed information about virtual services in the specified organization.

JDBC Compliant: No

Report Parameters:

Parameter	Description
Organization	<p>Listbox. Specifies the organization that owns the virtual service(s) you want to view in your report.</p> <p>Note: The Organization parameter is not available if there is only organization defined in CentraSite.</p>

Service Details

Displays the detailed information about a service.

JDBC Compliant: No

Report Parameters:

Parameter	Description
Organization	Listbox. Specifies the organization that owns the service(s) you want to view in your report. Note: The Organization parameter is not available if there is only organization defined in CentraSite.

Service Events

Displays a list of runtime events that were logged for the virtual service(s) for the given number of days.

JDBC Compliant: Yes

Report Parameters:

Parameter	Description
Target	Listbox. Specifies the target that identifies the virtual service(s) whose runtime events for the given number of days you want to view in your report. Note: The Target parameter is not available if there is only target defined in CentraSite.
UDDI Key for a Virtual Service	String. Specifies the UDDI V3 key of the virtual service whose runtime events you want to view in your report. Example: <code>uddi:207ff1cc-25c5-544c-415c-5d98ea91060c</code>
Number of Days	Number. Specifies the number of days that you want to collect runtime events for the virtual service.

Service Errors

Displays a list of error messages that were generated for the virtual service(s) for the given number of days.

JDBC Compliant: Yes

Report Parameters:

Parameter	Description
Target	Listbox. Specifies the target that identifies the virtual service(s) whose error messages for the given number of days you want to view in your report. Note: The Target parameter is not available if there is only target defined in CentraSite.
Number of Days	Number. Specifies the number of days that you want to collect error messages for the virtual service(s).

SLA Violations

Displays the violations to the service level agreement (SLA) that occurred for the virtual service(s).

JDBC Compliant: Yes

Report Parameters:

Parameter	Description
Target	Listbox. Specifies the target that identifies the virtual service(s) whose SLA violations you want to view in your report. Note: The Target parameter is not available if there is only target defined in CentraSite.
Organization	Listbox. Specifies the organization that owns the virtual service(s) you want to view in your report. Note: The Organization parameter is not available if there is only organization defined in CentraSite.

SOA Maturity

Displays the current state of SOA adoption of an organization.

JDBC Compliant: No

Report Parameters:

Parameter	Description
Organization	Listbox. Specifies the organization whose SOA maturity level you want to view in your report.

Parameter	Description
	Note: The Organization parameter is not available if there is only organization defined in CentraSite.

Top Ten Consumers

Displays the top ten consumers based on runtime invocations for the virtual service(s).

JDBC Compliant: Yes

Report Parameters:

Parameter	Description
Target	Listbox. Specifies the target that identifies the virtual service(s) whose top ten consumers you want to view in your report. Note: The Target parameter is not available if there is only target defined in CentraSite.
Organization	Listbox. Specifies the organization that owns the virtual service(s) you want to view in your report. Note: The Organization parameter is not available if there is only organization defined in CentraSite.

Top Ten Services

Displays the top ten services based on runtime invocations.

JDBC Compliant: Yes

Report Parameters:

Parameter	Description
Target	Listbox. Specifies the target that identifies the top ten virtual service(s). Note: The Target parameter is not available if there is only target defined in CentraSite.
Organization	Listbox. Specifies the organization that owns the virtual service(s).

Parameter	Description
	<p>Note: The Organization parameter is not available if there is only organization defined in CentraSite.</p>

Unreferenced Assets

Displays a list of all assets that have no incoming relationships.

JDBC Compliant: No

Report Parameters:

Parameter	Description
Organization	<p>Listbox. Specifies the organization whose list of asset(s) without any incoming relationships you want to view in your report.</p> <p>Note: The Organization parameter is not available if there is only organization defined in CentraSite.</p>

Configuring JDBC Support for BIRT Reports

CentraSite reports mainly use CentraSite Registry Repository (CRR) as data source. In addition to using CentraSite as data source, CentraSite reports can also access SQL databases using the Java Database Connectivity (JDBC).

CentraSite provides a set of report templates that are available to access an Oracle database, but will continue to use CentraSite as the default data source. For a list of the supported JDBC-compliant report templates, see [“Predefined Reports” on page 855](#).

These JDBC-compliant report templates can switch the data source dynamically on the basis of the selected target (Mediator).

To enable the access to an Oracle database, you must add the following XML fragment in the customization file, **centrasite.xml**. You can find this file on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\custom\conf`.

```
<RuntimeDatastore>
  <SqlDatabase
    targetName='Exact name of CentraSite target'
    url='URL of the SQL database (please see below)'
    driver='Database system specific driver (please see below)'
    user='User with the necessary permissions to access the SQL database'
    password='Password of the user'>
  </SqlDatabase>
</RuntimeDatastore>
```

Note:

Add an element `<SqlDatabase>` to the configuration for each target populating the metrics and event data to the SQL database, which should be accessed.

The **centrasite.xml** file should look as follows:

```
<?xml version='1.0' encoding='UTF-8'?>
<GUIConfiguration>
  ...
  <RuntimeDatastore>
    <SqlDatabase
      targetName='Exact name of CentraSite target'
      url='URL of the SQL database (please see below)'
      driver='Database system specific driver (please see below)'
      user='User with the necessary permissions to access the SQL database'
      password='Password of the user'
    </SqlDatabase>
    ...
  </RuntimeDatastore>
  ...
</GUIConfiguration>
```

You may include multiple `<SqlDatabase>` entries in the `<RuntimeDatastore>` configuration. The name of the target, defined by the attribute `targetName`, is case-sensitive, and therefore the name of the target and the name of the corresponding Mediator must exactly match.

The value of the attribute `password` will be encoded after accessing the SQL database with a BIRT report for the first time. A password with prefix `@secure:` is not permitted.

The driver and the URL format depends on the SQL database you are accessing. CentraSite supports the following drivers and URL formats:

DB2

- Driver: `com.softwareag.common.jdbc.driver.DB2ProxyDriver`
- URL format: `jdbc:wm:db2://<server_name>:<port>;databasename=<database_name>`

Microsoft SQL Server

- Driver: `com.softwareag.common.jdbc.driver.MSSQLProxyDriver`
- URL format: `jdbc:wm:sqlserver://<server_name>:<port>;databasename=<database_name>`

Oracle

- Driver: `com.softwareag.common.jdbc.driver.OracleProxyDriver`
- URL format: `jdbc:wm:oracle://<server_name>:<port>;sid=<database_name>`

Sybase

- Driver: `com.softwareag.common.jdbc.driver.SybaseProxyDriver`
- URL format: `jdbc:wm:sybase://<server_name>:<port>`

wherein, the substitution tokens <server_name>, <port>, and <database_name> are placeholders that should be replaced by real values.

Reserved Identifiers

The following identifiers are reserved and must not be used as part of a report template:

- __USRKEY
- ROKEY
- TARGETNAME
- MAIN_CS_ODA_DATA_SET, MAIN_CS_ODA_DATA_SET_1, ...
MAIN_CS_ODA_DATA_SET_9
- MAIN_SQL_ODA_DATA_SET, MAIN_SQL_ODA_DATA_SET_1, ...
MAIN_SQL_ODA_DATA_SET_9
- \$SQLDB_URL\$
- \$SQLDB_DRIVER\$
- \$SQLDB_USER\$
- \$SQLDB_BASE64PWD\$

Managing Reports and Report Templates through CentraSite Business UI

This section describes operations you can perform to manage reports and report templates through CentraSite Business UI.

Viewing the Report Template List

You use the Report Templates page to display the list of templates defined in CentraSite.

➤ To view the list of report templates

- In the CentraSite Business UI activity bar, click **Generare Reports**.

A list of defined report templates is displayed in the area labeled **Reports Templates**.

In addition, the Generate Reports page displays the set of actions that are available for working with a template.

Action	Description
Cancel	Opens the Welcome to CentraSite Business UI page.

Action	Description
View	Generates report using the BIRT reporting engine.
Download as	Downloads the generated report in the required format.
Schedule	Schedules the report template to automatically generate and distribute reports at recurring intervals.

Setting Permissions on Report Templates

To set permissions on a report template, you must have the Manage Report Templates permission or the Full instance-level permission on the template itself.

By default, all CentraSite users can view (and generate reports from) the report templates that you create through CentraSite Control. However, only you (as the owner of the report template) and users who belong to a role with the Manage Report Templates permission are initially allowed to modify and delete the templates through CentraSite Business UI.

To avoid that all users can view and generate reports from the report template that you have created or to enable other users to modify and delete a template that you have created, you must modify the template's instance-level permission settings.

The following general guidelines apply when setting permissions on templates:

- You can assign permissions to any individual user or group defined in CentraSite.
- The groups to which you can assign permissions include the following system-defined groups:

Group Name	Description
Users	All users within a specified organization.
Members	All users within a specified organization and its child organizations.
Everyone	All users of all organizations and child organizations, <i>including guest users</i> (if your CentraSite permits access by guests).

- If a user is affected by multiple permission assignments, the user receives the union of all the assignments. For example, if group ABC has Modify permission on a template and group XYZ has Full permission on the same template, users that belong to both groups will, in effect, receive Full permission on that template.
- If you have assigned users the Modify or Full permission on a template and you want them to be able to manage (modify or delete) the template in CentraSite Control, be sure that the users have Use the Reports UI permission.

> To assign permissions to a report template

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list.

The default search scope is **Assets**.

A list of defined assets in CentraSite (for which you have the View permission) is displayed in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the report templates, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Report Templates** option button, and follow these steps:
 - a. Click the chevron next to **Report Templates** option button.

A list of defined report templates in CentraSite is displayed.

- b. In the list of report templates, select the report template you want to assign the user and group permissions.
 - c. Click **OK**.
5. On the Actions bar of the Report Template Details page, click the **Permissions** icon.

This opens the **Assign Permissions** dialog box in which you want to assign the user and group permissions.

6. To add users or groups to the **User and Group Permissions** list, do one of the following:
 - a. Type a partial string in the **Add User or Group** text box. CentraSite applies the filter to the users and groups in registry.
 - b. Select the user or group to which you want to assign permissions.
 - c. Click the plus button next to the text box or press Enter to add the user or group to the **User and Group Permissions** list.

-OR-

- a. Click **Choose**. This opens the **Choose Users and Groups** dialog.
 - b. Type a partial string in the **Add User or Group** text box. CentraSite applies the filter to the users and groups in registry.
 - c. Click the **Search** icon.
 - d. Select the users and groups to which you want to assign permissions.
 - e. Click **OK**.

7. To remove a user or group from the **User and Group Permissions** list, select the **Delete** icon beside the group name or user ID.
8. Assign specific permissions to each user and group in the **User and Group Permissions** list as follows:

Permission	Allows the selected user or group to...
View	View the report template.
Modify	View and edit the report template.
Full	View, edit, and delete the report template.

9. Click **Save**.

Generating Reports Through Activity Bar

All CentraSite users have implicit (and irrevocable) view permission on report templates. This permission enables you to schedule and generate reports for any report template that is defined in CentraSite. To generate a report from the Generate Reports page, you do not explicitly need a UI permission.

The objects for which you can generate reports are:

Object	Description
Organization	An object that represents an organization in CentraSite.
Taxonomy	An object that represents a classification system in CentraSite.
Asset Type	An object that identifies a set of profiles which are collections of attributes.
Asset	An object that represents an entry in the asset catalog.
Policy	An object that identifies a set of actions that are executed when a specified event occurs to a specified object.

> To generate a report from the activity bar

1. In the CentraSite Business UI activity bar, click **Reports**.
2. In the displayed list of report templates, select the report template you want to use for report generation.
3. Click **View**.

If you have selected a template which includes required parameters, you are prompted to provide the required parameters in the **Report Parameter** dialog box to generate the report successfully.

4. In the **Report Parameter** dialog box, provide the required information for each of the displayed data fields.

The data fields are displayed based on the report you wish to generate.

5. Click **OK**.

CentraSite generates the required report using BIRT reporting engine, and then displays the generated report in CentraSite Business UI.

You can also select to generate and save the report simultaneously.

Generate Report from the Asset Details Page

All CentraSite users have implicit (and irrevocable) view permission on report templates. This permission enables you to schedule and generate reports for any report template that is defined in CentraSite. To generate a report from a particular asset details page, you must have the `View` instance-level permission on the asset itself.

The objects for which you can generate reports are:

Object	Description
Organization	An object that represents an organization in CentraSite.
Taxonomy	An object that represents a classification system in CentraSite.
Asset Type	An object that identifies a set of profiles which are collections of attributes.
Asset	An object that represents an entry in the asset catalog.
Policy	An object that identifies a set of actions that are executed when a specified event occurs to a specified object.

➤ To generate a report from the asset details page

1. In CentraSite Business UI, click the **Browse** link in the upper-left corner.
2. On the Search Results page, click the asset's link.
3. On the actions bar of the Asset Details page, click the **View Report** icon.
4. In the displayed list of report templates, select the report template you want to use for report generation.

5. Click **View**.

If you have selected a template which includes required parameters, you are prompted to provide the required parameters in the **Report Parameter** dialog box to generate the report successfully.

6. In the **Report Parameter** dialog box, provide the required information for each of the displayed data fields.

The data fields are displayed based on the report you wish to generate.

7. Click **OK**.

CentraSite generates the required report using BIRT reporting engine, and then displays the generated report in CentraSite Business UI.

You can also select to generate and save the report simultaneously.

Downloading Reports

You can download and save reports for a later use.

You may consider downloading a report if you want to:

- Work with the report data. For example, you can download your report to Excel format and then continue to work with the data in Excel format.
- Print the report in a different format. For example, you can download the report to the PDF file format and then print it.
- Save a copy of the report as another file type. For example, you can download a report to Word format and save it as another file type.

> To download a report

1. In the CentraSite Business UI activity bar, click **Reports**.
2. In the displayed list of report templates, select the report template you want to use for report generation.
3. Click **Download As**.

CentraSite displays the supported formats to download the report.

Format	Description
Comma separated value (CSV) File	The Comma-Separated Value (.csv) format renders a report as a flattened representation of data from a report in a standardized, plain-text format

Format	Description
	that is easily readable and usable with many other applications, for example spread sheet applications.
PDF document	The PDF (.pdf) format renders a report to files that can be opened in Adobe Acrobat and other third-party PDF viewers.
Word document	The Word (.docx) format renders a report as a Word document that is compatible with Microsoft Word.
Excel document	The Excel (.xlsx) format renders a report as an Excel document that is compatible with Microsoft Excel.

4. In the **Download As** list, select a format to download the report.

If you have selected a template which includes required parameters, you are prompted to provide the required parameters in the **Report Parameter** dialog box to generate the report successfully.

5. In the **Report Parameter** dialog box, provide the required information for each of the displayed data fields.

The data fields displayed depend on the report you wish to generate.

6. Click **OK**.

When you do this, you are prompted to do one of the following:

- Click **Open** to display the report in the selected format. For example, if you have previously selected the **Excel document** format, the generated report is displayed in an Excel spreadsheet.
- Click **Save** to save the newly generated report in CentraSite registry. For example, if you have previously selected the **Excel document** format, the generated report is saved as an .xslv file.

Note:

Some of the report parameters, for example, Organization, can have the Name value that starts with a special character: + (plus sign), - (minus sign or hyphen), @ (at sign), = (equal sign), / (forward slash), % (percentage), and " (double quote). When generating report for an organization (for example, +Organization ABC) using the download option as **Comma separated value (CSV) File** in CentraSite Business UI, the Name value will be prefixed with ' (single quote) character ('+Organization ABC) to prevent the formula injection in Microsoft Excel.

Scheduling Reports

You can schedule report templates to automatically generate and distribute reports at regular intervals. You can schedule the report template to generate reports daily, weekly, or monthly.

The scheduler triggers the report to be automatically generated and emailed as an attachment, in PDF format, to the user's email address.

➤ **To schedule a report**

1. In the CentraSite Business UI activity bar, click **Reports**.
2. In the displayed list of report templates, select the report template you want to schedule.
3. Click **Schedule**.

The Schedule Report Details page is displayed.

4. In the area labeled **Basic Information**, provide the required information for each of the displayed data fields.

Field	Description
Name	Name of the scheduled report. The report name can contain any characters (including spaces).
Description	<i>Optional.</i> The description for the scheduled report.

If you have selected a template which includes required parameters, you are prompted to provide the required parameters in the **Report Parameter** dialog box to generate the report successfully.

5. In the **Report Parameter** dialog box, provide the required information for each of the displayed data fields.

The data fields are displayed based on the report you wish to generate.

6. In the area labeled **Frequency**, select the frequency with which you want to schedule the report.

Field	Description
Occurs Daily	Recurs every N day(s) - The frequency with which you want the scheduled report to automatically execute on a regular basis. For example, if you specify Recurs every 10 day(s) , then your report executes every 10 days.
Occurs Weekly	Recurs every N week(s) on these weekday(s) - The exact days of the week when you want the scheduled report to execute automatically. Possible values - Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday

Field	Description
	For example, if you specify Recurs every 2 week(s) on these weekday(s) - Monday, Friday then your report will execute every 2 weeks on the specified days, that is, Monday and Friday.
Occurs Monthly	<ul style="list-style-type: none"> ■ Recurs on day N of every N month - The exact day of the month when you want the scheduled report to execute automatically. For example, if you specify Recurs on day 5 of every 2 month(s), then your report will execute on day 5 of every 2 months. ■ Recurs the first, second, third, fourth or last weekday of every N month - The ordinal with which you want the scheduled report to execute automatically. Possible values - first, second, third, forth, last. For example, if you specify Recurs the first Monday of every 3 Month(s), then your report will execute the first Monday of every 3 months.

The following table summarizes the effects of scheduled report configuration for seven exemplary cases:

If You Want the Schedule to Execute This Often	Occurs	Frequency (N)	Day (N) of a Month	Weekday	Ordinal
Every day	Daily	1	n/a	n/a	n/a
Every specific number of days (for example, every 7 days)	Daily	7	n/a	n/a	n/a
Every week on a certain day (for example, every week on Wednesday)	Weekly	1	n/a	Wednesday	n/a
Every specific number of weeks on a certain day (for example, every 2 weeks on Friday)	Weekly	2	n/a	Friday	n/a
Every month on a specific date, (for example, the 10th of every month)	Monthly	1	10	n/a	n/a

If You Want the Schedule to Execute This Often	Occurs	Frequency (N)	Day (N) of a Month	Weekday	Ordinal
Every month on a certain day (for example, the third Monday of every month)	Monthly	1	n/a	Monday	third
Every specific number of months on a certain day (for example, the last Friday of every 3rd month)	Monthly	3	n/a	Friday	last

7. Click **Save**.

Displaying a List of Scheduled Reports

If you have the CentraSite Administrator role, you can view and manage any scheduled report.

If you have a role other than the CentraSite Administrator role, you can view and manage only your scheduled reports.

> To view the list of scheduled reports

1. In the CentraSite Business UI, click your user name that is located in the upper-right corner of the header area.

This opens the User Preferences page.

2. Locate the **Scheduled Reports** panel.
3. In the displayed list of scheduled reports, hover over the scheduled report for which you want to display the details.

For each report, CentraSite specifies the display name and the schedule frequency.

If you have the CentraSite Administrator role, you see the owning user of a report, in addition to the above details.

Modifying Scheduled Report Details

If you have the CentraSite Administrator role, you can view and manage any scheduled report.

If you have a role other than the CentraSite Administrator role, you can view and manage only your scheduled reports.

> To modify a scheduled report

1. In CentraSite Business UI, click your user name that is located in the upper-right corner of the header area.

This opens the User Preferences page.

2. Locate the **Scheduled Reports** panel.
3. In the displayed list of scheduled reports, hover over the scheduled report for which you want to display the details.

This displays icons for one or more actions that you can perform on the scheduled report.

4. Click the **Edit** icon to modify the scheduled report configuration.

Deleting Scheduled Reports

If you have the CentraSite Administrator role, you can view and manage any scheduled report.

If you have a role other than the CentraSite Administrator role, you can view and manage only your scheduled reports.

> To delete a scheduled report

1. In CentraSite Business UI, click your user name in the upper-right corner of the header area.

This opens the User Preferences page.

2. Locate the **Scheduled Reports** panel.
3. In the displayed list of scheduled reports, hover over the scheduled report you want to delete.

This displays icons for one or more actions that you can perform on the scheduled report.

4. Click the **Delete** icon.
5. Click **OK** in the confirmation dialog box.

Changing the Default Scheduler for Generating Reports

By default, CentraSite contains the default settings for report scheduler. The default scheduler is set to run at 1 AM each night and use 8 threads to execute the scheduled reports. If necessary, you can specify a different scheduler.

CentraSite uses the global values set for `ReportSchedulingTime` and `ReportSchedulingThreads` to execute the default report scheduler. By default, `ReportSchedulingTime` is set to 1 AM and `ReportSchedulingThreads` is set to 8.

To reschedule the scheduler to run at a different time (other than the default 1 AM) and the number of threads (other than the default 8), change the global values set for `ReportSchedulingTime` and `ReportSchedulingThreads` in the **centrasite.xml** file.

➤ To modify the default report scheduler

1. Open the customization file, `centrasite.xml`, in a rich text editor.

You can find the **centrasite.xml** file on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\custom\conf`.

2. Locate the property named `<ReportSchedulingSettings>` in the configuration file.

The property `<ReportSchedulingSettings>` is commented and would look like the following:

```
<ReportSchedulingSettings>
  <!-- Specify hour or day when reports are checked for daily
  execution - example 22 for 10 PM -->
  <ReportSchedulingTime>1</ReportSchedulingTime>
  <!-- Number of threads to be used for reports execution -->
  <ReportSchedulingThreads>8</ReportSchedulingThreads>
</ReportSchedulingSettings>
```

3. Clear the comment for the property `<ReportSchedulingSettings>`.
4. Change the default values set for the report scheduler properties - `ReportSchedulingTime` and `ReportSchedulingThreads`.

The property details are:

Property	Description
<code>ReportSchedulingTime</code>	Specifies the hour of day when reports are checked for daily execution. For example, if you want to execute a report at 10 PM, then you would specify a value of 22.
<code>ReportSchedulingThreads</code>	Specifies the number of threads to be used for reports execution.

5. Save the configuration file.
6. Restart Software AG Runtime for the changes to take effect.

If at a later time, you want to switch to the default scheduler settings, comment the property `<ReportSchedulingSettings>`, and then restart Software AG Runtime.

Managing Reports and Report Templates through CentraSite Control

This section describes operations you can perform to manage reports and report templates through CentraSite Control.

Adding Report Template

To create report templates, you must have the Manage Report Templates permission. This permission allows you to modify and delete any report template (except certain predefined reports installed by CentraSite). By default, users in the CentraSite Administrator role and the Operations Administrator role have this permission.

If you do not belong to a role that includes the Manage Report Templates permission, you can modify and delete a report template if you have the:

- appropriate instance-level permissions (Modify or Full) on the report template.
 - AND —
- Use Reports UI permission.

CentraSite reports are designed using the BIRT (Business Intelligence and Reporting Tools), an open source, Eclipse-based reporting system that comprises an Eclipse-based report designer and a runtime component, that is, a viewer. Report template are deployed as BIRT report design files (`.rptdesign`). To create a custom report template, you must perform the following high-level steps:

1. Create a new `rptdesign` template file using the BIRT Report Designer.

During this step, you create a new `rptdesign` template file, where you specify the query for generating a report and select the fields and layout by which you would like to expose the report.

With the BIRT Report Designer, create a `rptdesign` file. This `rptdesign` file is referred to as the `Template File` in CentraSite Control.

The `rptdesign` file contains a parameter named `ROKEY`. This parameter helps to retrieve the object-specific metadata from the CentraSite data store.

2. Create a new Report Template in CentraSite Control.

During this step, you specify the details of the template and upload the `rptdesign` file with which you would like to generate the report.

You can add a report template to CentraSite Control, making it available to create new reports using this template.

> To add a report template

1. In CentraSite Control, go to **Reports > Report Templates**.

A list of defined templates is displayed in the Report Templates page.

2. In the **Report Templates** page, click **Add Report Template**.
3. In the **Add Report Template** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Template File	Click Browse to upload the rptdesign template file.
Name	Name of the report template. A report template name can contain any characters (including spaces). Note: A report template does not need to be unique within the organization. However, to reduce ambiguity, you should avoid giving multiple report templates the same name. As a best practice, we recommend that organizations adopt appropriate naming conventions to ensure the assignment of distinct report template names.
Description	(Optional). The description for the report template. This description appears when a user displays the list of templates in the Report Templates page.

4. Click **OK**.

The newly created report template is displayed in the Report Templates page.
5. To specify the object types that are allowed to use the template for association purposes, click the **Applicable to Object Types** tab, and then click **Applicable to Object Types**. Select one or more object types you want to associate with the template, or select them all by using the **All Object Types** check box.
6. To enable other users to view, modify, and delete a template that you have created, you must modify the template's permission settings. To do this, click the **Permissions** tab, and click **Add Users/Groups**. Select one or more users and groups to which you want to assign permissions. If you want to filter the list, type a partial string in the **Search** field. ContraSite applies the filter to the **Name** column. Click **OK**. Select the **View**, **Modify**, and **Full** check boxes to assign specific permissions to each user and group in the **Users/Groups** list.
7. To define custom properties for this report template, click the **Object-Specific Properties** tab, and click **Add Property**. In the **Add Object-Specific Properties** dialog box, provide a property name and a value, and click **OK**. You can add multiple values for a single property. You can optionally specify a namespace to fully identify the property. The property name can contain any character (excluding spaces).

8. Click **Save**.

Associating Report Template with Object Types

Note:

This functionality applicable only to CentraSite Control has been deprecated and will be removed in a future release.

You can associate a report template to multiple object types. When you do this, you can generate a report for the objects that are classified by the object types associated with this template.

The following general guidelines apply when associating a template with multiple object types:

- You can either select the check boxes for the object types individually, or select them all by using the **All Object Types** check box.
- By default, CentraSite displays the **All Object Types** check box as selected. When this check box is selected, the check boxes for the list of object types are automatically displayed as selected and disabled.

This default selection does not associate the template with the selected object types. Instead, it only allows the template to be applied to the selected object types (that is, CentraSite allows you to generate reports for the objects that are classified by the associated object types).

- If you clear the **All Object Types** check box and select ALL of the object types as selected, then there does not exist an association and the taxonomy is applicable to all of the selected object types.
- If you clear the **All Object Types** check box and the list of ALL the object types, then there does not exist an association and the taxonomy is still applicable to all the object types.

➤ To associate a report template with object types

1. In CentraSite Control, go to **Reports > Report Templates**.
2. Right-click a report template you want to associate with object types, and click **Details**.
3. In the Report Template Details page, click the **Applicable to Object Types** tab.
4. Select one or more object types you want to associate with this template.

Note:

To associate the report template with all asset types (including new asset types that might be added to the registry in the future), enable the **All Asset Types** option. Note that this option associates the template with asset types only. It does not associate the template with non-asset types, such as Organizations, Users, and Taxonomies, which can also be used with report templates.

5. Click **Save**.

Setting Permissions on Report Templates

To set permissions on a report template, you must have the Manage Report Templates permission or the Full instance-level permission on the template itself.

Note:

This functionality applicable only to CentraSite Control has been deprecated and will be removed in a future release.

By default, all CentraSite users can view (and generate reports from) the report templates that you create. However, only you (as the owner of the report template) and users who belong to a role with the Manage Report Templates permission are initially allowed to modify and delete the templates.

To avoid that all users can view and generate reports from the report template that you have created or to enable other users to modify and delete a template that you have created, you must modify the template's instance-level permission settings.

The following general guidelines apply when setting permissions on templates in CentraSite Control:

- You can assign permissions to any individual user or group defined in CentraSite.
- The groups to which you can assign permissions include the following system-defined groups:

Group Name	Description
Users	All users within a specified organization.
Members	All users within a specified organization and its child organizations.
Everyone	All users of all organizations and child organizations, <i>including guest users</i> (if your CentraSite permits access by guests).

- If a user is affected by multiple permission assignments, the user receives the union of all the assignments. For example, if group ABC has Modify permission on a template and group XYZ has Full permission on the same template, users that belong to both groups will, in effect, receive Full permission on that template.
- If you have assigned users the Modify or Full permission on a template and you want them to be able to manage (modify or delete) the template in CentraSite Control, be sure that the users have Use the Reports UI permission.

> To assign permissions to a report template

1. In CentraSite Control, go to **Reports > Report Templates**.

2. Right-click a template you want to assign the user and group permissions, and click **Details**.
3. In the Report Template Details page, click the **Permissions** tab.
4. Click **Add Users/Groups**.
5. Select one or more users and groups to which you want to assign permissions.
6. To filter the list, type a partial string in the **Search** field. Click **OK**.

CentraSite applies the filter to the **Name** column.

Examples

String	Description
b	Displays names that contain b.
bar	Displays names that contain bar.
%	Displays all users and groups.

7. Use the **View**, **Modify**, and **Full** check boxes to assign specific permissions to each user and group in the **Users/Groups** list as follows:

Permission	Allows the selected user or group to...
View	View the report template.
Modify	View and edit the report template.
Full	View, edit, and delete the report template. This permission also allows the selected user or group to assign instance-level permissions to the report template.

8. Click **Save**.

Viewing the Report Templates List

You use the Report Templates page to display the list of templates defined in CentraSite.

➤ To view the list of report templates

1. In CentraSite Control, go to **Reports > Report Templates**.

A list of the defined templates is displayed in the Reports Templates page.

- To filter the list to see just a subset of the available templates, type a partial string in the **Search** field.

CentraSite applies the filter to the **Name** column. The **Search** field is a type-ahead field, so as soon as you enter any characters, the display is updated to show only those templates whose name contains the specified characters. The wildcard character % is supported.

If you type...	CentraSite displays
b	Names that contain b
bar	Names that contain bar
%	All names

The Report Templates page provides the following information about each template:

Column	Description
Name	Name of the report template.
Description	The description for the report template.
Organization	The organization to which the report template belongs.

You can adjust the view to show or hide the individual column by using the **Select Columns** icon that is located in the upper-right corner of the Report Templates page.

The shortcut menu of a particular report template displays one or more actions that you can perform on that template.

Action	Description
Details	Displays the details page of the report template.
Delete	Deletes the template.
Export	Exports the template from the registry to an archive file on the file system.
Download Template File	Downloads an existing BIRT report design (.rptdesign) file.
Upload Template File	Uploads the modified BIRT report design (.rptdesign) file.
Add to Favorites	Adds a shortcut to the report template you want to use routinely or otherwise keep close at hand.
Show Template File	Displays the BIRT report design (.rptdesign) file.

Modifying Report Template Details

You can view or modify details of a report template in the details page.

The following general guidelines apply when modifying the details of a report template:

- Some attributes within a template are designed to be read-only and cannot be modified even if they appear in a template on which you have modify permission.
- Modifications that you make to a report template will take effect as soon as you save the template.

➤ To modify the details of a report template

1. In CentraSite Control, go to **Reports > Report Templates**.

A list of defined templates is displayed in the Report Templates page.

2. Right-click a template for which you want to modify the attributes, and click **Details**.

The Report Template Details page is displayed.

3. To modify the generic attributes of the report template that are displayed in the area labeled **Basic Information**, change values of the attributes in the respective data fields as required.
4. To update the BIRT report design (.rptdesign) file, on the **Actions** menu, select **Download Template File**. Make the necessary changes and upload the template file using the **Upload Template File** command on the **Actions** menu.
5. To update the object types that are associated with the report template, click the **Applicable to Object Types** tab, add or remove object types, and then click **OK**.
6. To update the permissions, click the **Permissions** tab, assign or remove permissions, and then click **OK**.
7. To update the custom defined properties for this template, click the **Object-Specific Properties** tab, change values of the properties as required, and then click **OK**.
8. Click **Save**.

Deleting Report Templates

You cannot delete predefined templates (not even if you belong to a role with the Manage Report Templates permission).

➤ To delete report templates

1. In CentraSite Control, go to **Reports > Report Templates**.

A list of the defined templates is displayed in the Reports Templates page.

2. Right-click a report template you want to delete, and click **Delete**.

You can also select multiple report templates, on the **Actions** menu, and click **Delete**.

3. Click **OK** in the confirmation dialog box.

Managing Reports and Report Templates through Command Line Interface

This section describes operations you can perform to manage reports and report templates through Command Line Interface.

Viewing the Reports List

Pre-requisites:

To view the list of existing reports through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `list Reports` for this purpose.

> To display the list of existing reports

- Run the command `list Reports`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list Reports [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list Reports -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
```

The response to this command could be:

```
Executing the command : list Reports
=====
id:                uddi:b595e05b-8da6-11df-b765-893ca7cd7608
name:              API Usage Report
description:       Short Description of the API Usage Report
classifications:  [shared]
=====
...
Successfully executed the command : list Reports
```

Adding Custom Report

Pre-requisites:

To add a new report through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `add Report` for this purpose.

> To add a new report

- Run the command `add Report`.

The syntax is of the format `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd add Report [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -file <CONFIG-FILE> -template <TEMPLATE-FILE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
CONFIG-FILE	Name of the configuration file which contains the report properties.
TEMPLATE-FILE	Name of the BIRT report design (<code>.rptdesign</code>) file.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd add Report -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-file c:\temp\APIUsageReportConfig.xml -template c:\temp\APIUsageReport.rptdesign
```

The response to this command could be:

```
Executing the command : add Report
Report API Usage Report was successfully added
Successfully executed the command : add Report
```

Report Configuration File

You can configure the details for new report in an XML configuration file. The configuration file *ReportConfig.xml* must look like the one below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties version="1.0">
  <entry key="com.centrasite.scheduledreport.Name"></entry>
  <entry key="com.centrasite.scheduledreport.Description"></entry>
  <entry key="com.centrasite.scheduledreport.Classifications"></entry>
</properties>
```

The contents of the configuration file are listed below:

Tag Name	Description
Name	(Mandatory). Name of the report.
Description	(Optional). The description for the report.
Classifications	(Optional). The classification of the report, which determines whether the report is shared or nonshared. The default is nonshared. If you want to share the report with API Portal, then you mark it as a shared report. If you do not have the classification marked as shared, then the report is automatically nonshared. For multiple classifications, use a comma to separate the values. Currently, CentraSite only supports the shared classification.

Here is a sample configuration file. The sample illustrates how you can add a new report with name API Usage Report, with description Short Description of the API Usage Report, and with the classification shared.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties version="1.0">
  <entry key="com.centrasite.report.Name">API Usage Report</entry>
  <entry key="com.centrasite.report.Description">Short Description of the
API Usage Report</entry>
  <entry key="com.centrasite.report.Classifications">shared</entry>
</properties>
```

Modifying Report Details

Pre-requisites:

To modify the details of an existing report through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `update Report` for this purpose.

The report you want to modify will be specified by the parameter `REPORT`. Any change of the report's property value must be provided using an updated configuration file. All property values provided by the configuration file will be overwritten. To retain an existing property value, you must delete the entire property entry from the configuration file.

➤ To modify the details of a report

- Run the command `update Report`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd update Report [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -report <REPORT> [-file <CONFIG-FILE>] [-template <TEMPLATE-FILE>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
REPORT	Name or key of the report you want to update. If the report name contains white spaces, enclose the name with "".
CONFIG-FILE	(Optional). Name of the configuration file which contains the report properties.
TEMPLATE-FILE	(Optional). Name of the BIRT report design (.rptdesign) file to update.

Examples (all in one line):

Providing report configuration file and BIRT report design file:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd update Report -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-report "API Usage Report" -file c:\temp\APIUsageReportConfig.xml -template
c:\temp\APIUsageReport1.rptdesign
```

Providing report configuration file:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd update Report -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-report "API Usage Report" -file c:\temp\APIUsageReportConfig.xml
```

Providing BIRT report design file:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd update Report -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-report "API Usage Report" -template c:\temp\APIUsageReport1.rptdesign
```

The response to this command could be:

```
Executing the command : update Report
Successfully executed the command : update Report
```

A sample update configuration file is shown.

The sample illustrates how you can overwrite the description of an existing report. All other property values will remain unchanged.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties version="1.0">
  <entry key="com.centrasite.report.Description">Detailed Description of the
API Usage Report </entry>
</properties>
```

Deleting Report

Pre-requisites:

To delete an existing report through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `delete Report` for this purpose.

➤ To delete an existing report

- Run the command `delete Report`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd delete Report [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -report <REPORT>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .

Parameter	Description
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, Administrator.
PASSWORD	The password for the registered CentraSite user identified by the parameter USER-ID.
REPORT	Name or UDDI key of the report you want to delete. If the report name contains white spaces, enclose the name with "".

Examples (all in one line):

Providing report name:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd delete Report -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-report "API Usage Report"
```

Providing report key:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd delete Report -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-report uddi:137738d6-c66b-11e4-8896-da7def414f92
```

The response to this command could be:

```
Executing the command : delete Report
Successfully executed the command : delete Report
```

Sharing Report with API Portal

Pre-requisites:

To share a report through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

In certain scenarios, API Portal users might require your customized reports for scheduling and monitoring the API usage.

CentraSite provides a command tool named `share Report` for this purpose.

➤ To sharing a report with API Portal

- Run the command `share Report`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd share Report [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -sharedReport <REPORT-NAME>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
SHARED-REPORT	Name of the report to share. If the report name contains white spaces, enclose the name with <code>"</code> .

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd share Report -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-sharedReport "API Usage Report"
```

The response to this command could be:

```
Executing the command : share Report
Successfully executed the command : share Report
```

Downloading Report Template

Pre-requisites:

To download the BIRT report design (that is used for constructing a report template) through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `download Report Template` for this purpose.

➤ To download the BIRT report design

- Run the command `download Report Template`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd download Report Template [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -report <REPORT> -template <TEMPLATE-FILE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .

Parameter	Description
PASSWORD	The password for the registered CentraSite user identified by the parameter USER-ID.
REPORT	Name or key of the report you want to download. If the report name contains white spaces, enclose the name with "".
TEMPLATE-FILE	Name of the output file to download the BIRT report design.

Examples (all in one line):

Providing report name:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd download Report Template
-url http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-report "API Usage Report" -template c:\temp\APIUsageReport1.rptdesign
```

Providing report key:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd download Report Template
-url http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-report uddi:137738d6-c66b-11e4-8896-da7def414f92 -template
c:\temp\APIUsageReport1.rptdesign
```

The response to this command could be:

```
Executing the command : download Report Template
Report Template successfully written to c:\temp\APIUsageReport2.rptdesign
Successfully executed the command : download Report Template
```

Viewing the Scheduled Reports List

Pre-requisites:

To view the list of existing scheduled reports through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `list Scheduled Reports` for this purpose.

➤ To display the list of existing reports

- Run the command `list Scheduled Reports`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list Scheduled Reports [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list Scheduled Reports -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
```

The response to this command could be:

```
Executing the command : list Scheduled Reports
=====
id:                uddi:b595e05b-8da6-11df-b765-893ca7cd7608
name:              Schedule Daily
description:       Executes the report and sends the result every day
report:           API Invocations (daily)
email:            xxx@domain.com
locale:           -
occurrence:       daily
frequency:        1
dayofmonth:       0
days:            []
ordinal:          -
=====
...
Successfully executed the command : list Scheduled Reports
```

Adding Custom Scheduled Report

Pre-requisites:

To add a new scheduled report through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `add Scheduled Report` for this purpose.

> To add a new scheduled report

- Run the command `add Scheduled Report`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd add Scheduled Report [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -file <CONFIG-FILE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
CONFIG-FILE	Name of the configuration file which contains the report scheduled properties.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd add Scheduled Report -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-file c:\temp\ScheduledReportConfig.xml
```

The response to this command could be:

```
Executing the command : add Scheduled Report
Successfully executed the command : add Scheduled Report
```

Scheduled Report Configuration File

You can configure the details for new scheduled report in an XML configuration file. The configuration file *ScheduledReportConfig.xml* must look like the one below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties version="1.0">
  <entry key="com.centrasite.scheduledreport.Name"></entry>
  <entry key="com.centrasite.scheduledreport.Description"></entry>
  <entry key="com.centrasite.scheduledreport.Report"></entry>
  <entry key="com.centrasite.scheduledreport.EmailAddress"></entry>
  <entry key="com.centrasite.scheduledreport.Locale"></entry>
  <entry key="com.centrasite.scheduledreport.Occurrence"></entry>
  <entry key="com.centrasite.scheduledreport.Frequency"></entry>
  <entry key="com.centrasite.scheduledreport.DayOfMonth"></entry>
  <entry key="com.centrasite.scheduledreport.Days"></entry>
  <entry key="com.centrasite.scheduledreport.Ordinal"></entry>
</properties>
```

The contents of the configuration file are listed below:

Tag Name	Description
Name	(Mandatory). The name of the scheduled report.
Description	(Optional). Description for the scheduled report.
Report	(Mandatory). The name of the report to execute.

Tag Name	Description
EmailAddress	(Mandatory). The email address of the consumer who should be informed of the result of the report execution.
Locale	(Optional). The locale of the report to execute. If the locale is not set, the default locale is used. Note that the current version of CentraSite only supports the default locale.
Occurrence	(Mandatory). The occurrence options with which you want to schedule a report to execute automatically. Possible values - daily, weekly, monthly.
Frequency	(Optional). The frequency with which you want a report to execute automatically on a regular schedule. Example: If you specify Occurrence=daily, Frequency=10, then your report will execute every 10 days.
DayOfMonth	(Optional). The exact day of the month when you want the scheduled report to execute automatically. Note: You can use this property, when you have the occurrence option set to monthly. Example: If you specify Occurrence=monthly, Frequency=2, DayOfMonth=5, then your report will execute on day 5 of every 2 months.
Days	(Optional). The exact days of the week when you want the scheduled report to execute automatically. Possible values - monday, tuesday, wednesday, thursday, friday, saturday, sunday Note: Specify the values as a comma-separated list. Example: If you specify Occurrence=weekly, Frequency=2, Days=[monday, friday] then your report will execute every 2 weeks on the specified days, that is, Monday and Friday.

Tag Name	Description
Ordinal	<p>(Optional). The ordinal with which you want the scheduled report to execute automatically. Possible values - first, second, third, forth, last</p> <p>Example:</p> <p>If you specify Occurrence=monthly, Frequency=3, Days=[monday], Ordinal=first, then your report will execute the first Monday of every 3 months.</p>

The following table summarizes the effects of scheduled report configuration for seven exemplary cases:

If You Want the Schedule to Execute This Often	Occurrence	Frequency	Day of the Month	Weekdays	Ordinal
Every day	Daily	1	n/a	n/a	n/a
Every specific number of days (for example, every 7 days)	Daily	7	n/a	n/a	n/a
Every week on a certain day (for example, every week on Wednesday)	Weekly	1	n/a	Wednesday	n/a
Every specific number of weeks on a certain day (for example, every 2 weeks on Friday)	Weekly	2	n/a	Friday	n/a
Every month on a specific date, (for example, the 10th of every month)	Monthly	1	10	n/a	n/a
Every month on a certain day (for example, the third Monday of every month)	Monthly	1	n/a	Monday	third
Every specific number of months	Monthly	3	n/a	Friday	last

If You Want the Schedule to Execute This Often	Occurrence	Frequency	Day of the Month	Weekdays	Ordinal
--	------------	-----------	------------------	----------	---------

on a certain day (for example, the last Friday of every 3rd month)

Here is a sample configuration file. The sample illustrates how you can add a new scheduled report with name Schedule Daily, with description Executes the report and sends the result every day, the report to execute is API Invocations (daily), the email address to send the output pdf file is xxx@domain.com and the schedule is every day.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties version="1.0">
  <entry key="com.centrasite.scheduledreport.Name">Schedule Daily</entry>
  <entry key="com.centrasite.scheduledreport.Description">
    Executes the report and sends the result every day</entry>
  <entry key="com.centrasite.scheduledreport.Report">
    API Invocations (daily)</entry>
  <entry key="com.centrasite.scheduledreport.EmailAddress">
    xxx@domain.com</entry>
  <entry key="com.centrasite.scheduledreport.Locale"></entry>
  <entry key="com.centrasite.scheduledreport.Occurrence">daily</entry>
  <entry key="com.centrasite.scheduledreport.Frequency">1</entry>
  <entry key="com.centrasite.scheduledreport.DayOfMonth"></entry>
  <entry key="com.centrasite.scheduledreport.Days"></entry>
  <entry key="com.centrasite.scheduledreport.Ordinal"></entry>
</properties>
```

Modifying Scheduled Report Details

Pre-requisites:

To modify the details of an existing report scheduled through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `update Scheduled Report` for this purpose.

The scheduled report you want to modify will be specified by the parameter `SCHEDULED-REPORT`. Any change of the report's property value must be provided using an updated configuration file. All property values provided by the configuration file will be overwritten. To retain an existing property value, you must delete the entire property entry from the configuration file.

➤ To modify the details of a scheduled report

- Run the command `update Scheduled Report`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd update Scheduled Report [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -scheduledreport <SCHEDULED-REPORT> -file <CONFIG-FILE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, Administrator.
PASSWORD	The password for the registered CentraSite user identified by the parameter USER-ID.
SCHEDULED-REPORT	Name or key of the scheduled report to update. If the scheduled report name contains white spaces, enclose the name with "".
CONFIG-FILE	Name of the configuration file which contains the scheduled report properties.

Examples (all in one line):

Providing scheduled report name:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd update Scheduled Report -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-scheduledreport "Schedule Daily" -file c:\temp\ScheduledReportUpdatedConfig.xml
```

Providing scheduled report key:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd update Scheduled Report -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-scheduledreport uddi:b595e05b-8da6-11df-b765-893ca7cd7608 -file
c:\temp\ScheduledReportUpdatedConfig.xml
```

The response to this command could be:

```
Executing the command : update Scheduled Report
Successfully executed the command : update Scheduled Report
```

A sample update configuration file is shown.

The sample illustrates how you can change the schedule of an existing scheduled report to execute every 2 weeks on the specified weekdays: `monday` and `friday`. All other property values will remain unchanged.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties version="1.0">
  <entry key="com.centrasite.scheduledreport.occurrence">weekly</entry>
```

```
<entry key="com.centrasite.scheduledreport.Frequency">2</entry>
<entry key="com.centrasite.scheduledreport.Days">monday,friday</entry>
</properties>
```

Triggering Scheduled Report

Pre-requisites:

To trigger an existing scheduled report through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `trigger Scheduled report` for this purpose.

> To trigger an existing scheduled report

- Run the command `trigger Scheduled report`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd trigger Scheduled Report [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -scheduledreport <SCHEDULED-REPORT>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
SCHEDULED-REPORT	Name or key of the scheduled report to trigger. If the report name contains white spaces, enclose the name with "".

Examples (all in one line):

Providing report name:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd trigger Scheduled Report
-url http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-scheduledreport "Schedule Daily"
```

Providing report key:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd trigger Scheduled Report
-url http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-scheduledreport uddi:b595e05b-8da6-11df-b765-893ca7cd7608
```

The response to this command could be:

```
Executing the command : trigger Scheduled Report
Successfully executed the command : trigger Scheduled Report
```

Deleting Scheduled Report

Pre-requisites:

To delete an existing scheduled report through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool `delete Scheduled Report` for this purpose.

> To trigger an existing scheduled report

- Run the command `delete Scheduled Report`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd delete Scheduled Report [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -scheduledreport <SCHEDULED-REPORT>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
SCHEDULED-REPORT	Name or key of the scheduled report you want to delete. If the report name contains white spaces, enclose the name with <code>"</code> .

Examples (all in one line):

Providing report name:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd delete Scheduled Report -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-scheduledreport "Schedule Daily"
```

Providing report key:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd delete Scheduled Report -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-scheduledreport uddi:b595e05b-8da6-11df-b765-893ca7cd7608
```

The response to this command could be:

```
Executing the command : delete Scheduled Report
Successfully executed the command : delete Scheduled Report
```

Viewing the Asset Types Associated to Report

Pre-requisites:

To list the asset types that are associated to a report through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `list Report Associations` for this purpose.

> To list asset types associated to a report

- Run the command `list Report Associations`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list Report Associations [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -associatedReport <ASSOCIATED-REPORT>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
ASSOCIATED-REPORT	Name or key of the report you want to fetch the list of associated asset types. If the report name contains white spaces, enclose the name with <code>"</code> .

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list Report Associations
-url http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-associatedReport "API Usage Report"
```

The response to this command could be:

```
Executing the command : list Report Associations
Successfully executed the command : list Report Associations
```

Associating Asset Type to Report

Pre-requisites:

To associate an asset type to a report through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `add Report Association` for this purpose.

➤ To list asset types associated to a report

- Run the command `add Report Association`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd add Report Association [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -associatedReport <ASSOCIATED-REPORT> -associatedType <ASSOCIATED-TYPE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
ASSOCIATED-REPORT	Name or ID (UUID) of the report you want to associate to asset type(s). If the report name contains white spaces, enclose the name with <code>"</code> .
ASSOCIATED-TYPE	Name of a particular asset type you want to associate to the report identified by the parameter <code>ASSOCIATED-REPORT</code> , or simply <code>"Asset Type"</code> to associate all of the predefined asset types to the report. If the asset type name contains white spaces, enclose the name with <code>"</code> . For multiple asset types, use a comma to separate the values.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd add Report Association -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-associatedReport "API Usage Report" -associatedType "REST Service"
```

The response to this command could be:

```
Executing the command : add Report Association
Successfully executed the command : add Report Association
```

Revoking Association Between Asset Type and Report

Pre-requisites:

To remove association between an asset type and report through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `delete Report Association` for this purpose.

➤ To list asset types associated to a report

- Run the command `delete Report Association`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd delete Report Association [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -associatedReport <ASSOCIATED-REPORT> -associatedType <ASSOCIATED-TYPE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
ASSOCIATED-REPORT	Name or ID (UUID) of the report you want to disassociate from the asset type(s). If the report name contains white spaces, enclose the name with <code>"</code> .
ASSOCIATED-TYPE	Name of a particular asset type you want to disassociate from the report identified by the parameter <code>ASSOCIATED-REPORT</code> , or simply <code>"Asset Type"</code> to disassociate all of the predefined asset types from the report. If the asset type name contains white spaces, enclose the name with <code>"</code> . For multiple asset types, use a comma to separate the values.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd delete Report Association
-url http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-associatedReport "API Usage Report" -associatedType "REST Service"
```

The response to this command could be:

```
Executing the command : delete Report Association
```

Successfully executed the command : delete Report Association

12 Portlet Management

■ Introduction to Portlets	910
■ Types of Portlets	910
■ Tailor Your Portlets	913
■ Adding a Portlet to Your Welcome Page	913
■ Viewing Your Portlets	922
■ Configuring a Portlet	922
■ Collapsing or Expanding Portlets	924
■ Rearranging Portlets	924
■ Removing Portlets	925
■ Built-in Design/Change-Time Portlets	926
■ Built-in Run-Time Portlets	937

Introduction to Portlets

The standard Welcome page gives you quick links to the pages of CentraSite Business UI that you will probably use frequently during your day-to-day work with CentraSite. It also provides links to external web sites that provide useful information related to CentraSite.

The Welcome page consists of a menu bar at the top and a set of default portlets below. Each portlet contains a header and content. The header includes a title, some selectable markers (example, to set user configuration of an individual portlet, expand or collapse a portlet, and so on) and a close button. Under the header, you can have a list of entries, either representing the result set of a search query, any external HTML page or a graphical image.

The result set of a search query represents a particular type of information, such as recent searches, recently created assets or changes to assets, most popular assets and so on.

An administrator defines the default set of portlets to display in the Welcome page. You can personalize the Welcome page to suit your requirements and preferences; you can add or remove portlets, rearrange portlets anywhere you want by simply dragging them, and customize the settings of individual portlets.

Using portlets, you can:

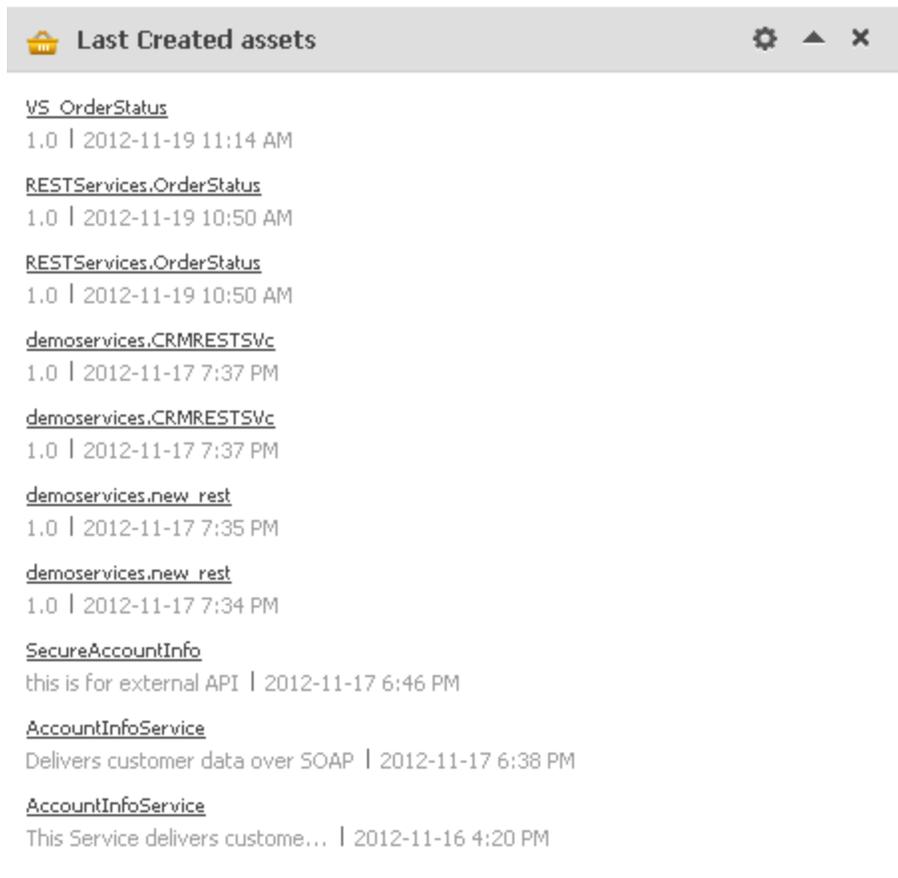
- Display the list of assets that you own in a single portlet.
- Display the list of recently modified assets in a single portlet.
- Display the list of active users in CentraSite registry in a single portlet.
- Create a quick link to the list of saved searches.
- Create a quick link to the list of recent saved searches.

Types of Portlets

CentraSite Business UI includes several types of portlets that you can add to your Welcome page. Each portlet has features that make the portlet suitable for particular types of content. The following sections describes each type of portlet.

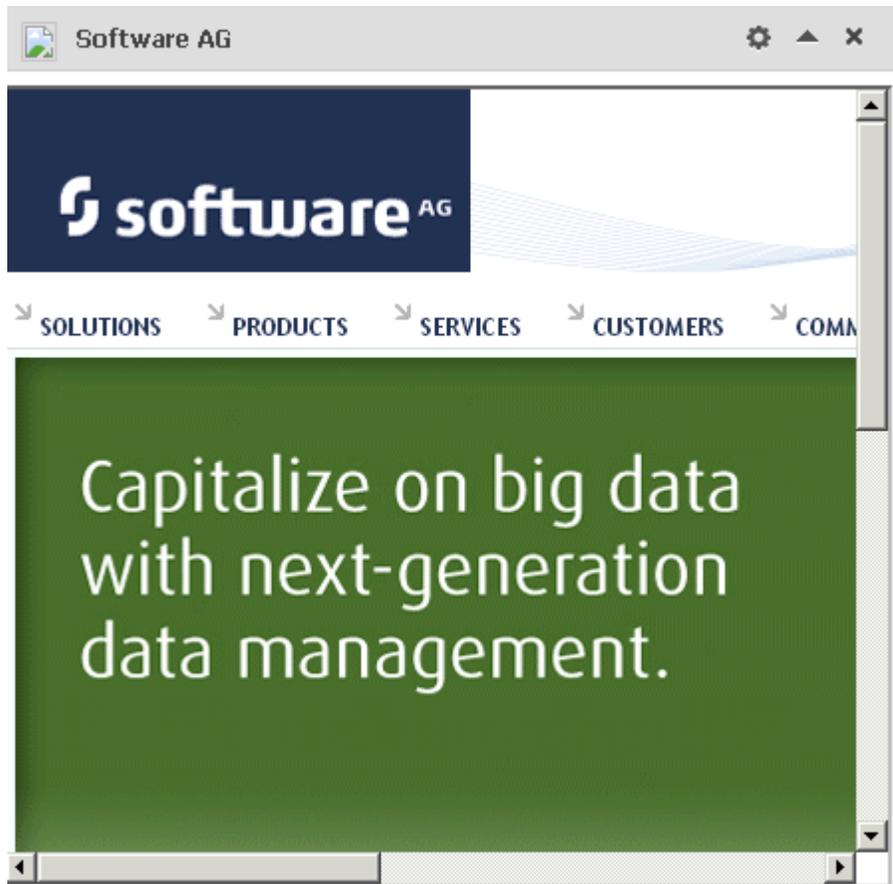
Text Portlet

A **Text** portlet enables you to view and work with content that you find by browsing or using the search tool at design time or runtime. You can create any number of text portlets and add them to your Welcome page. Here is a sample Text portlet:



IFrame Portlet

An **Inline Frame (IFrame)** portlet accesses a specified URL and displays the returned information within a rectangular region that includes scroll bars and borders. You can create any number of IFrame portlets and add them to your Welcome page. When you create an IFrame portlet, you usually supply a URL that points to a complete HTML page, as shown in the following example:

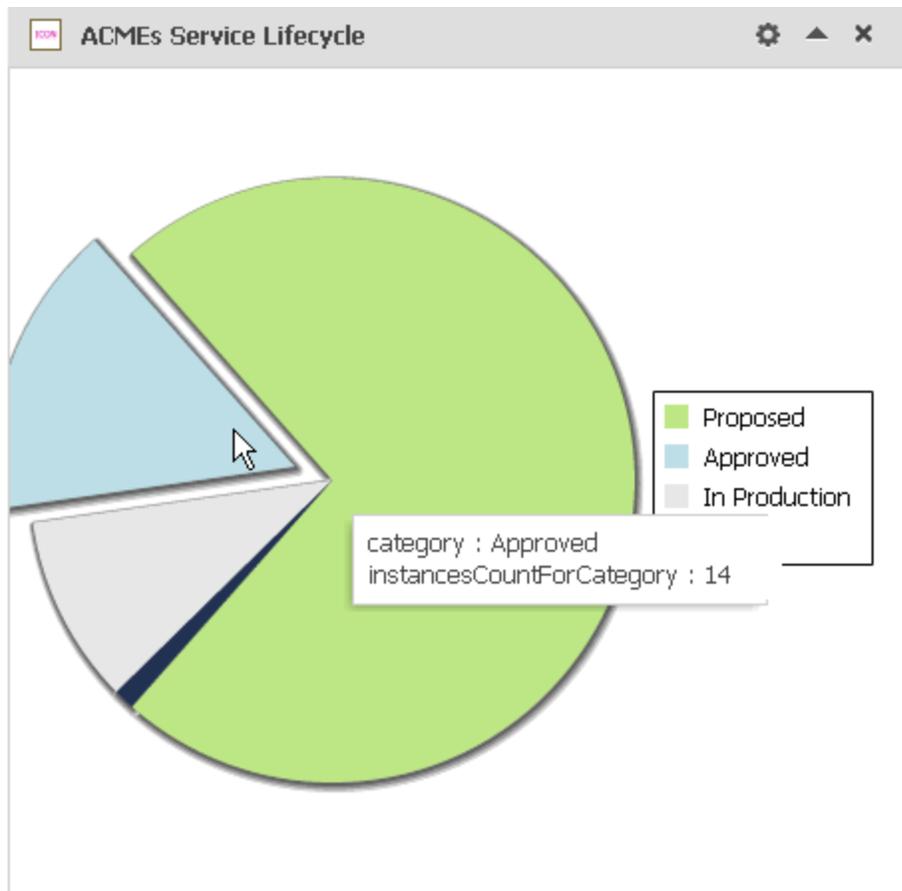


Within this IFRAME, portlets can display many types of content, including HTML, formatted text, images, or elements of an HTML form.

The default height of the frame is 400 pixels. (If necessary, the frame is displayed with scroll bars so that all of the portlet contents can be viewed within the frame.)

Graphical Portlet

A **Graphical** portlet displays a line chart, bar chart, or pie chart of data that is derived from a reporting search. You can create any number of graphical portlets and add them to your Welcome page. Here is a sample Graphical portlet:



Tailor Your Portlets

You can customize the portlets in your Welcome page in these ways:

- Add portlets — You can add an existing portlet or new portlet to your Welcome page using the **Configure** link at the top left corner of the Welcome page.
- Remove portlets — You can remove a portlet from your Welcome page by using the **Close** icon at the upper-right corner of each portlet.
- Change layout — You can drag and drop a portlet in to the desired position.
- Change portlet behavior — The portlet has a gear-shaped icon in the bar at the top, which you can click to display a **Settings** drop-down. This helps you redefine the behavior of a portlet.
- Show or hide portlet contents — You can switch the portlet to show or hide its contents using the expand or collapse icon in the bar at the top right corner.

Adding a Portlet to Your Welcome Page

You can add predefined and user-defined custom portlets to your Welcome page.

Use one of the following ways to add a portlet to your Welcome page:

- Add an existing portlet in CentraSite.
- Add a new custom portlet from scratch.

Important:

Alternatively, you can add the custom portlets to your Welcome page as GWT extension points.

Adding an Existing Portlet

You can add an existing portlet to your Welcome page using the **Configure Your Welcome Page** dialog box in CentraSite Business UI.

> To add an existing portlet to your Welcome page

1. In CentraSite Business UI, click the **Welcome** link that is located in the upper-right corner of the header area.

The Welcome page displays a list of portlets that are configured for your view.

2. Click the **Configure** link (below the label **Welcome to CentraSite Business UI**).

The **Configure your Welcome Page** dialog box opens to display the list of portlets that are available to you.

3. In the **Configure your Welcome Page** dialog box, select the check box of a single portlet, or select the check boxes of multiple portlets to add to your Welcome page.
4. Click **OK**.

The selected portlets are displayed in your personalized Welcome page.

Creating a Text Portlet

You can create a custom text portlet and add the newly created text portlet to your Welcome page using the **Create Portlet** dialog box in CentraSite Business UI.

> To create a Text portlet

1. In CentraSite Business UI, click the **Welcome** link that is located in the upper-right corner of the header area.

The Welcome page displays a list of portlets that are configured for your view.

2. Click the **configure** link (below the label **Welcome to CentraSite Business UI**).

The **Configure your Welcome Page** dialog box displays the list of portlets that are available to you.

3. In the **Configure Your Welcome Page** dialog box, click the **create a portlet** link.
4. In the **Create Portlet** dialog box, provide the required information for each of the displayed data fields:

Field	Description
Name	<p>Name of the portlet. A portlet name can contain any character (including spaces).</p> <p>A portlet name does not need to be unique within the Welcome page. However, to reduce ambiguity, you should avoid giving multiple portlets of the same type the same name. As a best practice, we recommend that you adopt appropriate naming conventions to ensure that portlets are distinctly named within the Welcome page.</p>
Description	(Optional). The description for the portlet.
Type	The portlet display type, Text .
Data Feed	The saved search query for the portlet.
Number of Entries	(Optional). The number of entries to display in the portlet. If the number of entries is set to 0, then all the available entries are displayed.
Attributes	The display attributes for the portlet. The attributes are dynamically displayed depending on the selected search query.

5. Click the chevron to expand the **Advanced Settings** panel. Provide the required information for each of the displayed data fields:

Field	Description
Actions	<p>(Optional). A set of actions that are available for the configuration of the Text portlet.</p> <p>The default actions include:</p> <ul style="list-style-type: none"> ■ Refresh ■ Configure
Icon URL	<p>(Optional). The path to an image file that is used to represent this portlet in the Welcome page.</p> <p>Prerequisite: The image file must be in PNG format. To ensure proper alignment when it is displayed in the user interface, the image must be 16 x 16 pixels in size.</p>

Field	Description
	<p>The image must reside in the folder <code><CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\images\system</code>.</p> <p>The path for the image should be specified, for example, as <code>images/system/icon.png</code></p>
Refresh Interval	<p>(Optional). The time interval (in seconds) after which a the portlet content is refreshed.</p> <p>If a value is not specified (or if the value 0 is specified), refresh will not happen.</p>
Suppress Zero Values	<p>The Suppress Zero Values option button determines how rows containing zero values are handled in the portlet. When this option button is selected, any row that contains a value equal to zero is hidden in the portlet.</p> <p>By default, the Suppress Zero Values property is set to Yes.</p>

6. Click **OK**.

The portlet that you just created is added to the CentraSite Registry Repository, and you are redirected to the **Configure Your Welcome Page** dialog box.

By default, the newly created portlet is disabled and is not displayed in the Welcome page.

7. In the **Configure Your Welcome Page** dialog box, select the newly created portlet to add to your Welcome page.

8. Click **OK**.

The newly created portlet is displayed in your personalized Welcome page.

Creating an iFrame Portlet

You can create a custom iFrame portlet and add the newly created iFrame portlet to your Welcome page using the **Create Portlet** dialog box in CentraSite Business UI.

> To create an iFrame portlet

1. In CentraSite Business UI, click the **Welcome** link in the upper-right corner of the header area.

The Welcome page displays a list of portlets that are configured for your view.

2. Click the **configure** link (below the label **Welcome to CentraSite Business UI**).

The **Configure your Welcome Page** dialog box displays the list of portlets that are available to you.

3. In the **Configure Your Welcome Page** dialog box, click the **create a portlet** link.
4. In the **Create Portlet** dialog box, provide the required information for each of the displayed data fields:

Field	Description
Name	Name of the portlet. A portlet name can contain any character (including spaces). A portlet name does not need to be unique within the Welcome page. However, to reduce ambiguity, you should avoid giving multiple portlets of the same type the same name. As a best practice, we recommend that you adopt appropriate naming conventions to ensure that portlets are distinctly named within the Welcome page.
Description	(Optional). The description for the portlet.
Type	The portlet display type, IFrame .
URL	An arbitrary URL that points to any external HTML page.

5. Click the chevron to expand the **Advanced Settings** panel. Provide the required information for each of the displayed data fields:

Field	Specify...
Actions	(Optional). A set of actions that are available for the configuration of the IFrame portlet. The default actions include: <ul style="list-style-type: none"> ■ Refresh ■ Configure
Icon URL	(Optional). The path to an image file that is used to represent this portlet in the Welcome page. Prerequisite: The image file must be in PNG format. To ensure proper alignment when it is displayed in the user interface, the image must be 16 x 16 pixels in size. The image must reside in the folder <CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\images\system.

Field	Specify...
	The path for the image should be specified, for example, as <code>images/system/icon.png</code>
Refresh Interval	(Optional). The time interval (in seconds) after which a the portlet content is refreshed.
	If a value is not specified (or if the value 0 is specified), refresh will not happen.

6. Click **OK**.

The portlet that you just created is added to the CentraSite Registry Repository, and you are redirected to the **Configure Your Welcome Page** dialog box.

By default, the newly created portlet is disabled and is not displayed in the Welcome page.

7. In the **Configure Your Welcome Page** dialog box, select the newly created portlet to add to your Welcome page.
8. Click **OK**.

The newly created portlet is displayed in your personalized Welcome page.

Creating a Graphical Portlet

You can create a custom graphical portlet and add the newly created graphical portlet to your Welcome page using the **Create Portlet** dialog box in CentraSite Business UI.

> To create a Graphical portlet

1. In CentraSite Business UI, click the **Welcome** link in the upper-right corner of the header area.
The Welcome page displays a list of portlets that are configured for your view.
2. Click the **configure** link (below the label **Welcome to CentraSite Business UI**).
The **Configure your Welcome Page** dialog box displays the list of portlets that are available to you.
3. In the **Configure Your Welcome Page** dialog box, click the **create a new portlet** link.
4. In the **Create Portlet** dialog box, provide the required information for each of the displayed data fields:

Field	Description
Name	<p>Name of the portlet. A portlet name can contain any character (including spaces).</p> <p>A portlet name does not need to be unique within the Welcome page. However, to reduce ambiguity, you should avoid giving multiple portlets of the same type the same name. As a best practice, we recommend that you adopt appropriate naming conventions to ensure that portlets are distinctly named within the Welcome page.</p>
Description	(Optional). The description for the portlet.
Type	The portlet display type, Graphical .
Data Feed	The saved search query for the portlet.
Chart Type	<p>The chart type. The supported types include:</p> <ul style="list-style-type: none"> ■ Bar Chart ■ Line Chart ■ Pie Chart <p>Bar Chart is the default.</p>
X-axis Label and Feed	<p>(For Bar and Line charts). The label that displays for the X-axis. A label can contain any character (including spaces).</p> <p>Select the attribute that you want to show in the X-axis, relative to the specified search query.</p>
Y-axis Label and Feed	<p>(For Bar and Line charts).The label that displays for the Y-axis. A label can contain any character (including spaces).</p> <p>Select the attribute that you want to show in the Y-axis, relative to the specified search query.</p>
Vertical Bar	<p>(For Bar and Line charts).The Vertical Bar option button specifies whether the display of the data is in a horizontal bar or a vertical bar format.</p> <p>The allowed values are Yes or No. Default Yes.</p>
X-axis Value Orientation	<p>(For Bar and Line charts). The X-axis Value Orientation option button specifies whether the display of the X-axis data is in a horizontal or a vertical direction.</p> <p>The allowed values are Vertical or Horizontal with the default being Vertical.</p>

Field	Description
Attribute Label	(For Pie charts). An attribute that displays in the pie chart, relative to the specified search query.
Attribute Value	(For Pie charts). A value for the attribute that dynamically displays in the pie chart, relative to the specified attribute label.

5. Click the chevron to expand the **Advanced Settings** panel. Provide the required information for each of the displayed data fields:

Field	Description
Actions	<p>(Optional). A set of actions that are available for the configuration of the Graphical portlet.</p> <p>The default actions include:</p> <ul style="list-style-type: none">■ Refresh■ Configure
Icon URL	<p>(Optional). The path to an image file that is used to represent this portlet in the Welcome page.</p> <p>Prerequisite: The image file must be in PNG format. To ensure proper alignment when it is displayed in the user interface, the image must be 16 x 16 pixels in size.</p> <p>The image must reside in the folder <CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\images\system.</p> <p>The path for the image should be specified, for example, as images/system/icon.png</p>
Refresh Interval	<p>(Optional). The time interval (in seconds) after which a the portlet content is refreshed.</p> <p>If a value is not specified (or if the value 0 is specified), refresh will not happen.</p>
Show Plot Values	<p>The Show Plot Values option button determines whether to show or hide the data value of a bar, line, or pie plot.</p> <p>The default is set to Yes.</p>
Plot Value Position	<p>The Plot Value Position option button specifies the position of the data value in a bar, line, or pie plot.</p> <p>Possible values are Start, End, Outside.</p>

Field	Description
	The default value for a bar or line chart is <code>outside</code> , and for a pie chart is <code>Start</code> .
Show Plot in Different Color	<p>The Show Plot in Different Color option button determines whether to show each bar, line, or pie plot in a different color.</p> <p>The default is set to <code>No</code>.</p> <p>When you set this option to <code>Yes</code>, specify the colors for each bar, line or pie plot in the <code>centrasite.xml</code> customization file. You should specify the colors using the HEX color code format. The HEX format is a hash (#) followed by 6 numbers or letters. The position of the numbers/letters correlate to the RGB value. For example, <code>#0000ff</code> translates into <i>blue</i>.</p>
Show Legend	<p>The Show Legend option button determines whether to show or hide legends in the chart.</p> <p>The legend appears by default — unless you specify to hide it in the graphical chart.</p>
Show Tooltip	<p>The Show Tooltip option button determines whether to show or hide tooltips in the chart.</p> <p>The tooltip appears by default — unless you specify to hide it in the graphical chart.</p>
Suppress Zero Values	<p>The Suppress Zero Values option button determines how rows or columns containing zero values are handled in the portlet. When this option button is selected, any row or column that contains a value equal to zero is hidden in the portlet.</p> <p>By default, the Suppress Zero Values property is set to Yes.</p>

6. Click **OK**.

The portlet that you just created is added to the CentraSite Registry Repository, and you are redirected to the **Configure Your Welcome Page** dialog box.

By default, the newly created portlet is disabled and is not displayed in the Welcome page.

7. In the **Configure Your Welcome Page** dialog box, select the newly created portlet to add to your Welcome page.

8. Click **OK**.

The newly created portlet is displayed in your personalized Welcome page.

Viewing Your Portlets

You can view the list of portlets that are available to you in the Welcome page.

Important:

A portlet is not displayed in the Welcome page unless you have the portlet added to the Welcome page using the **Configure Your Welcome Page** dialog box.

➤ **To view the portlets in your Welcome page**

1. In CentraSite Business UI, click the **Welcome** link that is located in the upper-right corner of the header area.

The Welcome page displays a list of portlets that are configured for your view.

2. Locate the portlet you want to view.

Each portlet contains multiple entries that match the data feed defined for the portlet.

A list of entries and attributes for each of these entries are displayed based on the portlet's configuration.

3. To view the tooltip text for an attribute in the portlet, hover over the attribute whose details you want to view. The tooltip text shows is the value of the attribute's **Name** field, as defined for an asset in its type definition.
4. To see all the portlets that are available to you, click the **configure** link (below the label **Welcome to CentraSite Business UI**).

Configuring a Portlet

You can modify the details of an existing portlet by using the Welcome page of CentraSite Business UI.

You can configure each portlet in the Welcome page to suit your preferences, modify the configuration parameters, and refresh the contents of the portlet.

The default configurable options for a portlet include the following:

- **Configure**
- **Refresh**

In addition to the default options, you can define your own custom configurable options for each portlet.

Personalize Your Portlets

You can personalize the functional settings of each portlet using the **Advanced Settings** option in CentraSite Business UI.

➤ To personalize the functional settings of a portlet

1. In CentraSite Business UI, click the **Welcome** link that is located in the upper-right corner of the header area.

The Welcome page displays a list of portlets that are configured for your view.

2. Locate the portlet you want to configure.
3. Click the **Settings** icon (in the upper-right corner of the header area).
4. Click **Configure**.

This displays the configuration details of the portlet.

5. Modify the basic configuration settings (such as, Name, Description, Attributes) of the portlet, as required.
6. Modify the extended settings of the portlet. Do the following:
 - a. Click the chevron to expand the **Advanced Settings** panel.
 - b. Modify the fields, as required.
7. After you have made the required changes, click **OK**.

Refresh Intervals for Portlets

You can configure the refresh intervals for each portlet using the **Advanced Settings** option in CentraSite Business UI.

The content in your portlet may need to be refreshed periodically if you are examining real-time data. You can reset the refresh intervals for individual portlets in your Welcome page.

➤ To reset the refresh interval for a portlet

1. In CentraSite Business UI, click the **Welcome** link that is located in the upper-right corner of the header area.

The Welcome page displays a list of portlets that are configured for your view.

2. Locate the portlet you want to configure.
3. Click the **Settings** icon (in the upper-right corner of the header area).
4. Click **Configure**.

This displays the configuration details of the portlet.

5. Click the chevron to expand the **Advanced Settings** panel.
6. In the field labeled **Refresh Interval**, type a new interval, as required.
7. After you have made the required changes, click **OK**.

Collapsing or Expanding Portlets

You can collapse or expand a portlet using the Welcome page in CentraSite Business UI.

> To collapse or expand a portlet

1. In CentraSite Business UI, click the **Welcome** link that is located in the upper-right corner of the header area.

The Welcome page displays a list of portlets that are configured for your view.

2. Locate the portlet you want to collapse or expand.
3. Click the chevron to collapse or expand the list of available data.

Rearranging Portlets

You can rearrange the portlets displayed in the Welcome page of CentraSite Business UI.

> To rearrange portlets in the Welcome page

1. In CentraSite Business UI, click the **Welcome** link that is located in the upper-right corner of the header area.

The Welcome page displays a list of portlets that are configured for your view.

2. Locate the portlet you want to rearrange.
3. To move a portlet, grab the portlet by its header bar, and drag it to the desired location in your Welcome page.

Removing Portlets

You can remove the individual portlets that are displayed in the Welcome page of CentraSite Business UI.

➤ **To remove portlets**

1. In CentraSite Business UI, click the **Welcome** link in the upper-right corner of the header area.

The Welcome page displays a list of portlets that are configured for your view.

2. Click the **configure** link (below the label **Welcome to CentraSite Business UI**).

You can alternatively click the **Close** icon (in the upper-right corner of the portlet's header area).

The **Configure your Welcome Page** dialog box opens to display the list of portlets that are available to you.

3. Clear the check box of a single portlet, or clear the check boxes of multiples portlets you want to remove from the Welcome page.

4. Click **OK**.

Each of the selected portlets are temporarily removed from the personalized Welcome page.

5. To permanently remove (delete) a portlet from the CentraSite Registry Repository, do the following:

- a. In CentraSite Business UI, click your user name that is located in the upper-right corner of the header area.

This opens the User Preferences page.

- b. Locate the **My Portlets** panel.

A list of portlets that are available to you is displayed.

- c. In the displayed list of portlets, hover over the portlet you want to delete.

This displays icons for one or more actions that you can perform on the portlet.

- d. Click **Delete**.

- e. When you are prompted to confirm the delete operation, click **Yes**.

The selected portlet is permanently removed from the personalized Welcome page.

Built-in Design/Change-Time Portlets

This section describes the set of design/change-time report searches that are installed as portlets with CentraSite.

Portlet	Description
Asset Instance Count Per Category for Taxonomy	Number of asset instances currently defined for the specified category (taxonomy).
Get Undelivered Access Tokens	List of all access tokens that were not delivered to API Portal during the retry attempts.
Inbox Notifications	List of all activity notifications received in your Inbox.
Instance Count Per State for Lifecycle Model	Number of currently defined asset instances for each lifecycle state.
Instances Per Type	Number of currently defined asset instances for each asset type.
Last Created Assets	List of all assets that were recently added to the CentraSite registry.
Last Updated Assets	List of all assets that were recently modified in the CentraSite registry.
Linked Instance Count Per Category For Taxonomy	Number of currently defined asset instances for each category (taxonomy).
My All List	Displays all lists that are available to you.
My All Saved Searches	List of all saved searches that were defined by you.
My API Keys	List of all the API keys that were requested by you.
My Approval Requests	List of all requests that you submitted and, if any, requests that were submitted on your behalf by another user.
My Favorites	List of all assets that you marked as favorites.
My List	Displays the lists that you created.
My Pending Approval Requests	List of all requests for which you are an authorized approver.
My Pending Consumer Registration Requests	Deprecated. List of all pending consumer registration requests for which you are an authorized approver.

Portlet	Description
My Saved Searches	List of the X number of saved searches that were defined by you.
Recent Lifecycle Changes	List of all assets whose lifecycle state was recently modified in the CentraSite registry.
Top X Assets by Consumers	List of the top X assets based on number of consumers.
Top X Assets by Incoming Association	List of the top X assets based on number of incoming associations.
Top X Assets by Watchers	List of the top X assets based on number of watchers.
Top X Assets Consumed Per Organization	Number of assets consumed by each organization.
Top X Assets Provided Per Organization	Number of assets defined by each organization.
Top X Assets with Watcher and Consumer Count	List of the top X assets based on the total number of watchers and consumers.
Top X Most Versioned Services	List of the top X services that were versioned for the maximum number of times.

Asset Instance Count Per Category for Taxonomy

Number of asset instances that are classified by a taxonomy or category.

The input parameters are:

Parameter	Description
Taxonomy	<i>Required</i> . <i>String</i> . Specifies the name of a taxonomy or category to filter the asset instances.

The result attributes are:

- **Category** – Name of the taxonomy.
- **Instance Count for Category** – Number of asset instances classified using the taxonomy or category.

Get Undelivered Access Tokens

List of all the access tokens that were not delivered to API Portal during the retry attempts.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer</i> . Specifies the number of access tokens to display in the portlet. Default is 5.

Inbox Notifications

List of all the recent activity notifications in your Inbox.

No input parameters.

Instance Count Per State for Lifecycle Model

Number of asset instances that are assigned to a lifecycle model.

The input parameters are:

Parameter	Description
Lifecycle Model	<i>Required . String</i> . Specifies the name of a lifecycle model to filter the asset instances.

The result attributes are:

- **Lifecycle Model** – Name of the lifecycle model.
- **Instance Count for Lifecycle Model** – Number of asset instances assigned to the lifecycle model.

Instances Per Type

Number of asset instances that are defined for an asset type.

No input parameters.

The result attributes are:

- **Asset Type** – Name of the asset type definition.
- **Instances Count for Asset Type** – Number of asset instances that are defined for the asset type.

Last Created Assets

List of all assets that were recently added to the CentraSite registry.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer .</i> Specifies the number of assets to display in the portlet. Default is 5.
Asset Type	<i>Required . String .</i> Specifies the name of an asset type to filter the asset instances.

The result attributes are:

- **Asset Name** – The fully qualified name of the asset. Click on an asset name to view more information about that asset.
- **Asset Description** – Descriptive information about the asset.
- **Asset Version** – User-assigned version identifier for the asset.
- **Asset Created Date** – Date when the asset was created.

Last Updated Assets

List of all assets that were recently modified in the CentraSite registry.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer .</i> Specifies the number of assets to display in the portlet. Default is 5.
Asset Type	<i>Required . String .</i> Specifies the name of an asset type to filter the asset instances.

The result attributes are:

- **Asset Name** – The fully qualified name of the asset. Click on an asset name to view more information about that asset.
- **Asset Description** – Descriptive information about the asset.
- **Asset Version** – User-assigned version identifier for the asset.
- **Asset Updated Date** – Date when the asset was last modified.

Linked Instance Count Per Category For Taxonomy

Number of asset instances that are classified by a taxonomy or category.

The input parameters are:

Parameter	Description
Taxonomy	<i>Required . String</i> . Specifies the name of a taxonomy or category to filter the asset instances.

The result attributes are:

- **Category** – Name of the taxonomy.
- **Instance Count for Category** – Number of asset instances that are classified using the taxonomy.

My All List

Displays all the lists you have in CentraSite.

No input parameters.

The result attributes are:

- **List Name** – Name of the list.
- **Last Modified Date** – Date when the list was last modified.

My All Saved Searches

List of all the saved searches you have in CentraSite.

No input parameters.

The result attributes are:

- **Search Name** – Name of the saved search. Click on a saved search name to view more information on the search criteria and results.
- **Level** – Saved search is either user-specific, organization-specific, or global-specific.
- **Last Modified Date** – Date when the saved search was last modified.

My API Keys

List of all the API Keys that are available to you.

No input parameters.

My Approval Requests

List of all the requests for which you are an authorized approver (that is, the list includes any request whose approver group included you as a member).

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer .</i> Specifies the number of approval requests to display in the portlet. Default is 5.

The result attributes are:

- **Asset Name** – The fully qualified name of the asset for which an approval request has been triggered. Click on an asset name to view more information about that asset.
- **Asset Description** – Descriptive information about the asset.
- **Approval Flow Name** – Name of the approval workflow.
- **Approval Flow Creation Date** – Date when the approval workflow was created.
- **Approval Flow Status** – Status of the approval workflow (for example, In Progress, Approved, or Rejected).

My Favorites

List of all the assets you have marked as a favorite.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer .</i> Specifies the number of favorite assets to display in the portlet. Default is 5.

The result attributes are:

- **Asset Name** – The fully qualified name of the asset. Click on an asset name to view more information about that asset.
- **Asset Description** – Descriptive information about the asset.
- **Asset Version** –User-assigned version identifier for the asset.

My List

Displays the first X number of lists you have in CentraSite.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer .</i> Specifies the number of lists to display in the portlet. Default is 5.

The result attributes are:

- **List Name** – Name of the list.
- **Last Modified Date** – Date when the list was last modified.

My Pending Approval Requests

List of all the requests for which you are an authorized approver (that is, the list includes any request whose approver group included you as a member).

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer .</i> Specifies the number of approval requests to display in the portlet. Default is 5.

The result attributes are:

- **Pending Asset Name** – The fully qualified name of the asset for which an approval request has been triggered. Click on an asset name to view more information about that asset.
- **Pending Asset Description** – Descriptive information about the asset.
- **Pending Asset Version** – User-assigned version identifier for the asset.
- **Requestor User Name** – User who initiated the approval workflow.
- **Approval Flow Name** – Name of the approval workflow.
- **Approval Flow Description** – Descriptive information about the approval workflow.

My Pending Consumer Registration Requests

Deprecated. List of all the pending consumer registration requests for which you are an authorized approver.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer .</i> Specifies the number of consumer registration requests to display in the portlet. Default is 5.

The result attributes are:

- **Consumer Request Id** – Unique identifier of the registration request to consume the asset.
- **Requested Asset Name** – The fully qualified name of the asset for which a consumer registration request has been triggered. Click on an asset name to view more information about that asset.
- **Requested Asset Version** – User-assigned version identifier for the asset.
- **Requested Asset Description** – Descriptive information about the asset.

My Saved Searches

Displays the first X number of saved searches you have in CentraSite.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer .</i> Specifies the number of saved searches to display in the portlet. Default is 5.

The result attributes are:

- **Search Name** – Name of the saved search.
- **Level** – Saved search is either user-specific, organization-specific, or global-specific.
- **Last Modified Date** – Date when the saved search was last modified.

Recent Lifecycle Changes

List of all the assets whose lifecycle state was modified after the specified number of days.

The input parameters are:

Parameter	Description
Number of Days Past	<i>Required . Integer .</i> Specifies the number of days in the past to list the assets for which the lifecycle model was recently modified.

The result attributes are:

- **Asset Name** – The fully qualified name of the asset. Click on an asset name to view more information about that asset.
- **Asset Description** – Descriptive information about the asset.
- **Asset Version** – User-assigned version identifier for the asset.

Top X Assets by Consumers

List of the top X assets based on the number of consumers.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer</i> . Specifies the number of most popular assets (based on the consumers count) to display in the portlet. Default is 5.

The result attributes are:

- **Asset Name** – The fully qualified name of the asset. Click on an asset name to view more information about that asset.
- **Asset Description** – Descriptive information about the asset.
- **Asset Version** – User-assigned version identifier for the asset.
- **Consumers Count for Asset** – Number of consumers for the asset.

Top X Assets by Incoming Association

List of the top X assets based on the number of incoming associations.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer</i> . Specifies the number of most popular assets (based on the total count of incoming associations) to display in the portlet. Default is 5.

The result attributes are:

- **Asset Name** – The fully qualified name of the asset. Click on an asset name to view more information about that asset.
- **Asset Version** – User-assigned version identifier for the asset.

- **Incoming Associations Count** – Number of incoming associations for the asset.

Top X Assets by Watchers

List of the top X assets based on the number of watchers.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer .</i> Specifies the number of most popular assets based on the watchers count) to display in the portlet. Default is 5.

The result attributes are:

- **Asset Name** – The fully qualified name of the asset. Click on an asset name to view more information about that asset.
- **Asset Version** – User-assigned version identifier for the asset.
- **Watchers Count for Asset** – Number of watchers for the asset.

Top X Assets Consumed Per Organization

List of the top X assets consumed in each organization.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer .</i> Specifies the number of most popular assets (based on the consumption in each organization) to display in the portlet. Default is 5.

The result attributes are:

- **Organization Name** – Name of the organization.
- **Consumed Assets Count for Organization** – Number of assets consumed in this organization.

Top X Assets Provided Per Organization

List of the top X assets created in each organization.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer .</i> Specifies the number of most popular assets (based on the usage in each organization) to display in the portlet. Default is 5.

The result attributes are:

- **Organization Name** – Name of the organization.
- **Provided Assets Count for Organization** – Number of assets created in this organization.

Top X Assets with Watcher and Consumer Count

List of the top X assets based on the total number of watchers and consumers.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer .</i> Specifies the number of most popular assets (based on the total count of watchers and consumers) to display in the portlet. Default is 5.

The result attributes are:

- **Asset Name** – The fully qualified name of the asset. Click on an asset name to view more information about that asset.
- **Asset Version** – User-assigned version identifier for the asset.
- **Watchers and Consumers Count for Asset** – The total number of watchers and consumers for the asset.

Top X Most Versioned Services

List of the top X services that were versioned for the maximum number of times.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required . Integer .</i> Specifies the number of most versioned assets to display in the portlet. Default is 5.

The result attributes are:

- **Asset Name** – The fully qualified name of the asset. Click on an asset name to view more information about that asset.
- **Asset Description** – Descriptive information about the asset.
- **Asset Version** – User-assigned version identifier for the asset.

Built-in Run-Time Portlets

This section describes the set of run-time report searches that are installed as portlets with CentraSite.

Portlet	Description
Service Performance Metrics	Displays the run-time metrics of the selected service over a specific period of time.
Service Performance Metrics Over Time	Displays the run-time metrics of the selected service over an extended period of time.
Top X Consumers Based on Runtime Invocations	Lists the top X consumers based on the maximum run-time invocations on services.
Top X Monitoring Events Per Service	Lists the top X services based on the maximum run-time events on services for the given number of days.
Top X Services Based on Invocations	Lists the top X services based on the maximum invocations for the given number of days.
Top X Services Based on Payload Size	Lists the top X services based on the maximum payload size.
Top X Services Based on Runtime Errors	Lists the top X services based on the maximum run-time errors.
Top X Services Based on Runtime Policy Violations	Lists the top X services based on the maximum run-time policy violations.

Service Performance Metrics

The Service Performance Metrics portlet in the Welcome page displays the run-time metrics for a service over a specified period of time.

The input parameters are:

Parameter	Description
Service Key	<i>Required.</i> Specifies the Universally Unique Identifier (UUID) of the service that you want to include in the view.

Parameter	Description
Start Time / End Time	<i>Required.</i> Specifies the starting and ending date and time during which you want to examine metrics.

The result attributes are:

- **Service Name** – The fully qualified name of the service.
- **Service Key** – The Universally Unique Identifier (UUID) that is assigned to the service and uniquely identifies it within the registry.
- **Service Description** – The comment or descriptive information about the service.
- **Service Version** – The user-assigned version identifier for the service.
- **Total Request Count** – The total number of requests for the service running for the current interval.
- **Total Success Count** – The number of successful service invocations for the service for the current interval.
- **Total Fault Count** – The number of failed invocations for the service for the current interval.
- **Minimum Response Time** – The minimum amount of time (in milliseconds) it took to complete an invocation in the current interval.
- **Maximum Response Time** – The maximum amount of time (in milliseconds) it took to complete an invocation in the current interval.
- **Average Response Time** – The average amount of time it took the service to complete each invocation in the current interval.

Service Performance Metrics Over Time

The Service Performance Metrics portlet in the Welcome page displays the run-time metrics for a service over an extended period of time.

The input parameters are:

Parameter	Description
Service Key	<i>Required.</i> Specifies the Universally Unique Identifier (UUID) of the service that you want to include in the view.
Time Interval	<i>Required.</i> Specifies the time interval (as expressed in Days, Hours, Minutes, and Seconds) between consecutive repeats of examining metrics.
Duration (Past X Days/Hours)	<i>Required.</i> Specifies a duration (as expressed in Days, Hours, Minutes, and Seconds) during which you want to examine metrics.

The result attributes are:

- **Service Name** – The fully qualified name of the service.
- **Service Key** – The Universally Unique Identifier (UUID) that is assigned to the service and uniquely identifies it within the registry.
- **Service Description** – The comment or descriptive information about the service.
- **Service Version** – The user-assigned version identifier for the service.
- **Total Request Count** – The total number of requests for the service running for the current interval.
- **Total Success Count** – The number of successful service invocations for the service for the current interval.
- **Total Fault Count** – The number of failed invocations for the service for the current interval.
- **Minimum Response Time** – The minimum amount of time (in milliseconds) it took to complete an invocation in the current interval.
- **Maximum Response Time** – The maximum amount of time (in milliseconds) it took to complete an invocation in the current interval.
- **Average Response Time** – The average amount of time it took the service to complete each invocation in the current interval.

Top X Consumers Based on Runtime Invocations

The Top X Consumers Based on Runtime Invocations portlet in the Welcome page lists the top X consumers based on number of run-time invocations of services.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required. Integer.</i> Specifies the number of most popular consumers (which is based on the total number of invocations) that you want to include in the view. By default, this portlet displays up to five consumers.

The result attributes are:

- **Consumer Application Name** – The fully qualified name of the consumer application asset.
- **Consumer Application Description** – The comment or descriptive information about the consumer application asset.
- **Invocation Count for Consumer Application** – The number of invocations made by the consumer application asset on a service.

Top X Monitoring Events Per Service

The Top X Monitoring Events per Service portlet in the Welcome page lists the top X run-time events for a selected service for the given number of days.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required. Integer.</i> Specifies the number of most popular run-time events for the service that you want to include in the view.
Number of Days Past	<i>Required. Integer.</i> Specifies the number of days in the past to filter run-time events for the service.

The result attributes are:

- **Service Name** – The fully qualified name of the service.
- **Service Key** – The Universally Unique Identifier (UUID) that is assigned to the service and uniquely identifies it within the registry.
- **Service Description** – The comment or descriptive information about the service.
- **Service Version** – The user-assigned version identifier for the service.
- **Monitoring Event Count for Service** – The number of monitoring events made on the service.

Top X Services Based on Invocations

The Top X Services Based on Runtime Invocations portlet in the Welcome page lists the top X services based on run-time invocations for the given number of days.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required. Integer.</i> Specifies the number of most popular services (which is based on the maximum number of invocations) that you want to include in the view.
Number of Days Past	<i>Required. Integer.</i> Specifies the number of days in the past to filter services that had the maximum number of invocations.

The result attributes are:

- **Service Name** – The fully qualified name of the service.
- **Service Key** – The Universally Unique Identifier (UUID) that is assigned to the service and uniquely identifies it within the registry.

- **Service Description** – The comment or descriptive information about the service.
- **Service Version** – The user-assigned version identifier for the service.
- **Invocation Count for Service** – The number of invocations made on the service.

Top X Services Based on Payload Size

The Top X Services Based on Payload Size portlet in the Welcome page lists the top X services with maximum payload size.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required. Integer.</i> Specifies the number of services that you want to include in the view.
Maximum Payload Size	<i>Required. Integer.</i> Specifies the maximum payload size (in bytes).

The result attributes are:

- **Service Name** – The fully qualified name of the service.
- **Service Key** – The Universally Unique Identifier (UUID) that is assigned to the service and uniquely identifies it within the registry.
- **Service Description** – The comment or descriptive information about the service.
- **Service Version** – The user-assigned version identifier for the service.
- **Maximum Payload Size for Service** – The maximum payload size for the service.

Top X Services Based on Runtime Errors

The Top X Services Based on Runtime Errors portlet in the Welcome page lists the top X services with maximum run-time errors.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required. Integer.</i> Specifies the number of services that you want to include in the view.

The result attributes are:

- **Service Name** – The fully qualified name of the service.
- **Service Key** – The Universally Unique Identifier (UUID) that is assigned to the service and uniquely identifies it within the registry.

- **Service Description** – The comment or descriptive information about the service.
- **Service Version** – The user-assigned version identifier for the service.
- **Error Count for Service** – The number of runtime errors marked on the service.

Top X Services Based on Runtime Policy Violations

The Top X Services Based on Runtime Policy Violations portlet in the Welcome page lists the top X services with maximum run-time policy violations.

The input parameters are:

Parameter	Description
Number of Entries	<i>Required. Integer.</i> Specifies the number of services that you want to include in the view.

The result attributes are:

- **Service Name** – The fully qualified name of the service.
- **Service Key** – The Universally Unique Identifier (UUID) that is assigned to the service and uniquely identifies it within the registry.
- **Service Description** – The comment or descriptive information about the service.
- **Service Version** – The user-assigned version identifier for the service.
- **Policy Violations Count for Service** – The number of runtime policy violations marked on the service.

13 Runtime Governance

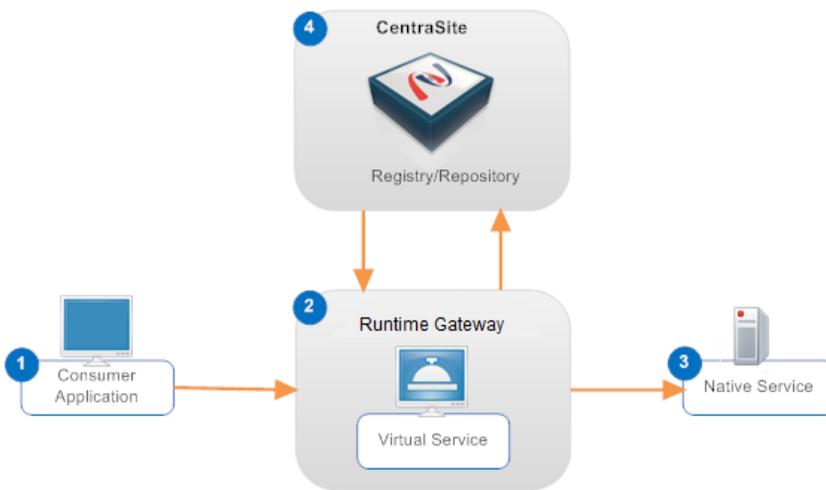
■ Introduction to Runtime Governance	944
■ Virtual Service Asset Management	959
■ Run-Time Policy Management	1176
■ Gateway Management	1364
■ Consumer Management	1397
■ Access Token Management	1430
■ Run-Time Alias Management	1463
■ Endpoint Management	1472
■ Runtime Events and Key Performance Indicator (KPI) Metrics	1479
■ Monitoring Logs	1511

Introduction to Runtime Governance

This section introduces the most important aspects of CentraSite's service oriented architecture (SOA) runtime governance.

The Components of Runtime Governance

When you use CentraSite to govern Web services at run-time, the basic components in the run-time environment include the following - native services, virtual services, runtime policies, runtime gateways, consumer applications, CentraSite registry repository, and the runtime events and metrics.



Native Service

A *Native Service* is a plain Web service that processes the requests submitted by consumers. When a native Web service produces a response, CentraSite returns the response to a virtual service, and then the virtual service returns it to the consumer.

You use CentraSite to define Web services of the type - SOAP Service, REST Service, and OData Service.

For more information on defining and managing Web service assets, see [“Managing Assets through CentraSite Business UI” on page 379](#).

Virtual Service

A *Virtual Service*, also called as Virtual Alias, Virtual API, is a service that runs on a runtime gateway and acts as the consumer-facing proxy for a Web service that runs elsewhere on the network. It provides a layer of abstraction between the service consumer and the service provider, and promotes loose coupling by providing location, protocol, and format independence between the consuming

application and the provider service. You can choose to expose any kind of a native Web service as virtual service, which performs a role similar to the native service.

For example, virtual services enable you to:

- Move native services to other physical addresses or switch providers without affecting existing consumer applications.
- Bridge differences (for example, transport differences, message structure differences) between the capabilities of a consuming application and the requirements of a native service.
- Block portions of a service interface from certain consuming applications (that is, expose selected portions of the native service to certain consumers).
- Provide access to different versions of a service through a single endpoint.

You use CentraSite to define proxy services of the type - Virtual SOAP Service, Virtual REST Service, and Virtual OData Service, and to deploy them on specified runtime gateways.

When you define a virtual service in CentraSite, you also define the following criteria, which enables users to securely access the virtual service:

- Policy enforcement rules
- API keys and OAuth2 client credentials

When you choose to expose a Web service as virtual service, you need to:

1. Create the Virtual Service asset
2. Configure governance rules for the virtual service
3. Create consumer applications to consume the virtual service at run-time
4. Deploy virtual service and consumer applications to a runtime gateway

After you deploy a virtual service, you use CentraSite as a dashboard to view the performance metrics and other runtime data relating to its usage.

For more information on defining and deploying virtual service assets, see [“Virtual Service Asset Management” on page 959](#).

Runtime Policy

A *Runtime Policy*, also called as governance rules, defines a set of policy actions that are to be carried out by a runtime gateway when a consumer requests access to a particular Web service through the gateway. The actions in a runtime policy perform activities such as identifying and authenticating consumers, validating digital signatures, logging run-time events, and capturing performance measurements.

You use CentraSite to define runtime policies with the required sequence of actions and configuration parameters, associate them with virtual services, and deploy them on specified gateways in the runtime environment. You also use CentraSite to monitor quality-of-service and other performance metrics for the services to which you have attached runtime policies.

For more information on defining global policies, see [“Run-Time Policy Management” on page 1176](#).

For more information on defining policies for a particular virtual service, see [“Virtual Service Asset Management” on page 959](#).

Runtime Gateway

A *Runtime Gateway*, also called as Policy-Enforcement Point (PEP), hosts virtual services, which are proxy services that receive requests from consumer applications on behalf of a particular Web service.

A runtime gateway is a registry object that represents a particular instance of a policy enforcement point (for example, an instance of API Gateway or Mediator). The gateway object specifies the address of the deployment endpoint, which is the endpoint that CentraSite uses to interact with gateway to deploy the virtual services.

The gateway handles mediation measures between consumer and provider such as protocol bridging, message transformation, and message routing. Besides serving as an intermediary between consumer applications and Web services, the gateway also collects performance statistics and event information about the traffic flowing between consumer applications and the Web services and reports this data to CentraSite.

CentraSite supports the following runtime gateways:

- webMethods API Gateway
- webMethods Mediator

To use an instance of CentraSite with an instance of webMethods API Gateway or webMethods Mediator, you must define a Gateway asset that identifies the specific instance of API Gateway or Mediator you want to use.

If you use multiple gateways with an instance of CentraSite, you must create a Gateway asset instance for each gateway. To make the gateways easier to distinguish when they are viewed in CentraSite, consider adopting a naming convention for gateways that clearly identifies to which environment the gateway belongs (for example, development, test, production). You can deploy any given virtual services to one or more gateways.

Note:

In addition to using webMethods API Gateway or webMethods Mediator, you can use webMethods Insight. Insight Server is an additional monitoring tool from Software AG that you can use with CentraSite. Insight Server enables you to see what is happening in real-time with service transactions as they flow across any system. It provides visibility and control at the transaction level to heterogeneous SOA environments. CentraSite provides support for Insight as a gateway type out-of-the-box. For more information about Insight's uses and capabilities, see the Insight user documentation.

Instead of (or in addition to) using the above runtime gateways for mediation and/or policy enforcement, you can use other third-party products with CentraSite. Support for third-party policy-enforcement and runtime governance tools is available through integrations that are

provided by members of the CentraSite Community. These tools are made available through the Software AG TECHcommunity Website .

For more information on defining and registering runtime gateways with CentraSite, see [“Gateway Management” on page 1364](#).

Consumer Application

A *Consumer Application* is represented in CentraSite by an Application asset. Application assets are used by the runtime gateway to determine from which consumer application a request for an asset originated. An application asset defines the precise characteristics (for example, a list of user names in HTTP headers, a range of IP addresses, and so on) by which the gateway can identify and authenticate messages from a specific consumer application and enable consumer access to a protected Web service at run-time. Consumer applications submit requests for operations that are provided by Web services that reside on various systems in the network. As shown in the figure above, a consumer application submits its request to a virtual service on the gateway and not directly to the Web service itself.

The ability of runtime gateway to relate a message to a specific consumer application enables the gateway to:

- Indicate the consumer application to which a logged transaction event belongs.
- Monitor a virtual service for violations of a service-level agreement (SLA) for a specified consumer application.
- Control access to a virtual service at run-time (that is, allow only authorized consumer applications to invoke a virtual service).

For more information on defining consumer applications and configuring identifiers for the consumer authentication, see [“Consumer Management” on page 1397](#).

CentraSite Registry Repository

The *CentraSite Registry Repository* (CRR) serves several key roles in the run-time environment. Besides serving as the system of record for the artifacts in the SOA environment (such as Virtual Services and their runtime governance policies), CentraSite provides the tools you use to define these artifacts and deploy them to Mediator. Additionally, CentraSite receives the performance metrics and event data collected by Mediator and provides tools for viewing this data.

Runtime Events and Metrics

A *Runtime Event* is an event that occurs while virtual services are actively deployed on the runtime gateway.

Examples of runtime events include:

- Successful or unsuccessful requests/responses.
- Policy violation events, which are generated upon violation of virtual service's runtime policy.

- Service monitoring events, which are generated by the service-monitoring actions in the runtime policy.

Runtime Metrics, also called as KPI metrics, are used to monitor the runtime execution of virtual services. Metrics include the maximum response time, average response time, fault count, availability of virtual services, and more. If you include runtime monitoring actions in your runtime policies, the actions will monitor the KPI metrics for virtual services, and can send alerts to various destinations when user-specified performance conditions for a virtual service are violated.

The runtime events and KPI metrics are collected by the gateway and published to CentraSite via SNMP.

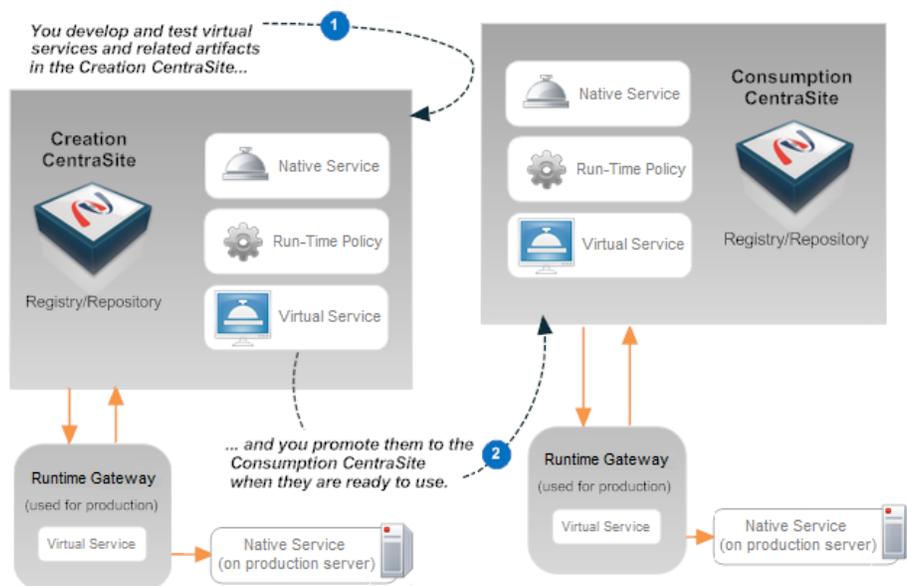
The gateway publishes the following types of runtime events - Lifecycle, Error, Policy Violation, Transaction, and Monitoring.

For the Monitoring event type, gateway publishes the following types of KPI metrics - Availability, Average Response Time, Fault Count, Maximum Response Time, Minimum Response Time, Successful Request Count, and Total Request Count.

For more information on configuring events and metrics logging, see [“Runtime Events and Key Performance Indicator \(KPI\) Metrics”](#) on page 1479.

Runtime Governance Deployment Architecture

When you use CentraSite for run-time governance, Software AG recommends that you use a two-stage deployment as shown in the following diagram. With this deployment strategy, *each instance of CentraSite has a set of runtime gateways*. This configuration creates an air gap between the development and production environments, which completely separates the components that support your production applications and services from those that support development and testing.



Creation Run-time environment

The creation environment supports the development and testing of run-time policies and virtual services. It is used by the following types of users:

- *Developers, analysts, or other authorized CentraSite users* publish the native services that are developed and added to the SOA environment.
- *Policy administrators* develop and test the run-time policies that are to be applied to the native services when they are virtualized.
- *Asset administrators* create and test the virtual services that are used to mediate access to the native services.

After a virtual service has been created and tested on the creation CentraSite, the virtual service, the run-time policies associated with it, and the native service that it represents are promoted to the Consumption CentraSite.

Consumption Run-time environment

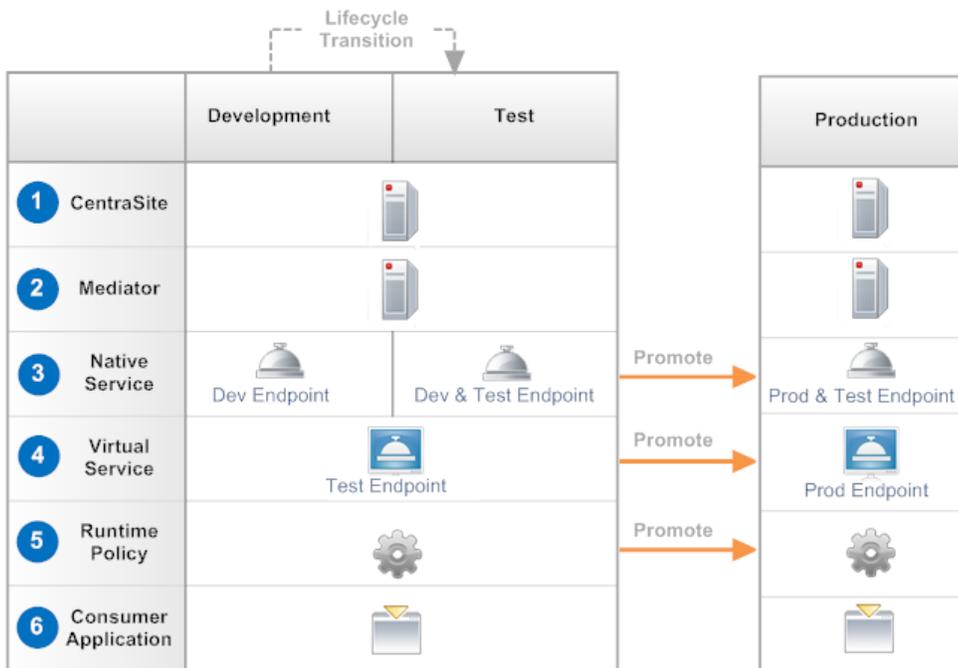
The Consumption CentraSite supports the production environment. Typically, Consumption CentraSite is managed by the Operations or IT organization and only users in this organization are permitted to publish assets to it.

The Consumption CentraSite is used by the following types of users:

- *Developers and analysts* access the Consumption CentraSite to discover services that are available for reuse. When access to a native service is mediated by a virtual service, users who browse the catalog for re-usable services see the virtual service, not the native service itself.
- *Designated administrators* from the IT or Operations organization import the virtual services, run-time policies, and native services that have been developed and tested in the creation CentraSite. These administrators also:
 - Adjust the permissions settings to ensure that these objects can be viewed by the appropriate groups of users (for example, they typically hide native services from developers and analysts who browse the catalog for reusable services and expose only the virtual services to those users).
 - Deploy virtual services to the Mediator.
- *Owners of the consumer applications* that invoke virtual services access the consumption CentraSite to view the performance metrics and other events relating to the operation of virtual services running in the Mediator.

Operations between the Creation and Consumption Environments

The following diagram illustrates how the creation and consumption instances of CentraSite interrelate:



Description

- CentraSite:** The creation CentraSite supports the development and testing of services and virtual services and the consumption CentraSite supports services and virtual services that are in production. Typically, both registries have the same basic organizational structure, although they might each have certain utility organizations that are unique to their role as a creation or consumption server. (The two instances of CentraSite are not required to have the same organizational structure. They can have different structures if that approach better suits your needs.)
- Mediator:** Each instance of CentraSite has its own Mediator (or Mediators). The Mediator in the creation environment provides a test bed that developers use during the development of virtual services and run-time policies. The Mediator in the consumption CentraSite is used exclusively for hosting virtual services that are in production.
- Native Services:** A native service begins its lifecycle on the creation CentraSite. When the service is ready for production, you promote it to the consumption CentraSite. On the consumption CentraSite, the native service is typically hidden from users who browse the catalog looking for services to reuse and is visible only to certain administrators (as a best practice).

The catalog entry for a native service includes the endpoint(s) where the service is deployed. As indicated by the figure above, these endpoints evolve as the service moves through development, test, and production.
- Virtual Services:** Like a native service, a virtual service begins its lifecycle on the creation CentraSite. You cannot create a virtual service until the native service it

Description

represents is registered in the creation CentraSite and it has been deployed on an endpoint in the network.

Typically, one creates a virtual service during the late stages of the development phase or when the native service enters the test phase (in other words, after the service's interface is completely implemented and an instance of the service is deployed and running at a known point in the development environment).

After the virtual service has been tested and it is considered ready for production use, it is promoted to the consumption CentraSite and deployed to a production Mediator.

- 5 **Run-time Policy:** A run-time policy defines a sequence of standard policy-enforcement actions that are to be executed when a virtual service is invoked.

Administrators define and test run-time policies on the creation CentraSite. When the policies are considered ready for production use, they are promoted to the consumption CentraSite.

Note:

Before you deploy a virtual service to the Mediator in the consumption environment, it is important to ensure that all the run-time policies that apply to the virtual service have been promoted to the consumption CentraSite.

- 6 **Consumer Application:** A consumer application identifies an application that invokes virtual services. Consumer applications are defined directly on the consumption CentraSite. They are not promoted from the creation CentraSite. (Administrators on the creation CentraSite who define run-time policies typically define dummy consumers for testing purposes.)

Note:

A consumer application is represented by an Application asset in the registry.

Enabling CentraSite Run-Time Aspects

You cannot configure the run-time policies for an API from CentraSite by default; and the run-time policies must be configured from the gateway where the APIs are published. However, you can enable the CentraSite run-time policies by modifying the `centrasite.xml` file.

The `RuntimeComponentSetting` setting in the `centrasite.xml` file is used to determine if the run-time aspects configured from CentraSite must be enabled or not. The value of this setting is *false* by default to use CentraSite as a pure design-time application. You can set its value to *true* to use CentraSite as a design-time as well as a run-time application.

Note:

When you migrate 10.4 from a lower version, the run-time aspects are automatically enabled. You can disable them from the `centrasite.xml`. For steps to disable, see [“Disabling CentraSite Run-Time Aspects” on page 952](#).

➤ To enable the run-time policies configured from CentraSite

1. Open the customization file, `centrasite.xml`, in a rich text editor.

You can find the **centrasite.xml** file located in the directory `CentraSiteInstall_Directory\cast\cswebapps\BusinessUI\custom\conf`.

2. Locate the `RuntimeComponentSetting` setting in the file and set its value to `true`.
3. Save and close the file.

The run-time policies can now be configured from CentraSite.

Disabling CentraSite Run-Time Aspects

The CentraSite run-time aspects are disabled, by default. However, you can enable or disable them whenever required. They are also enabled automatically when you migrate to 10.4 from a lower version. When enabled, you can disable CentraSite the run-time aspects using the `RuntimeComponentSetting` setting located in the `centrasite.xml` file.

➤ To disable the run-time policies configured from CentraSite

1. Open the customization file, `centrasite.xml`, in a rich text editor.

You can find the **centrasite.xml** file located in the directory `CentraSiteInstall_Directory\cast\cswebapps\BusinessUI\custom\conf`.

2. Locate the `RuntimeComponentSetting` setting in the file and set its value to `false`.
3. Save and close the file.

The run-time policies can now be configured from CentraSite.

Runtime Governance with Mediator

This section provides the important information you need for designing and configuring your CentraSite's runtime governance environment to use webMethods Mediator as your gateway.

Mediator Deployment Model

The deployment process is carried out by a sequence of interactions that occur between CentraSite and Mediator:

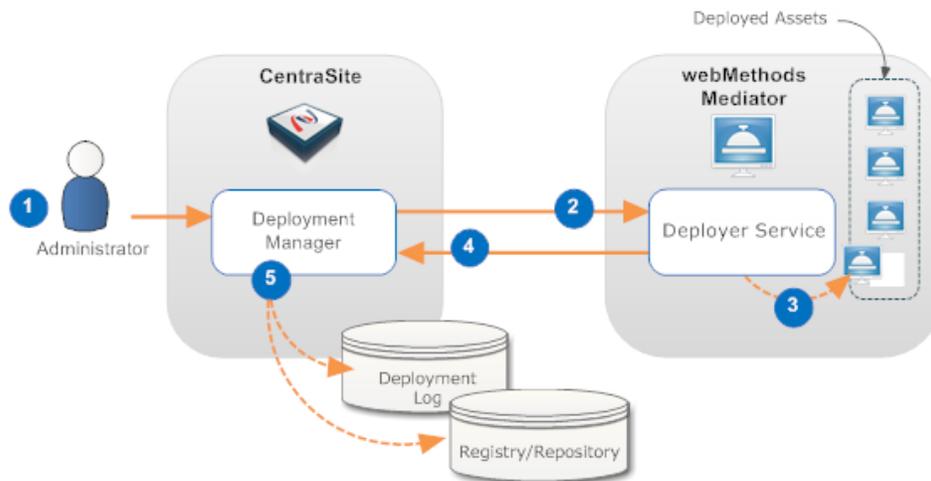
1. CentraSite pushes the virtual service that is ready for deployment to Mediator gateway.
2. Mediator deploys the virtual service that was received from CentraSite along with its effective runtime policy.

The deployment process is initiated from CentraSite and is carried out by the Deployer service on Mediator.

3. Mediator notifies CentraSite when the deployment process is complete.

Note:

The diagram demonstrates the deployment process of virtual services to Mediator.



Step Description

- 1 An administrator initiates the deployment by selecting the assets that are to be deployed and specifies to which Mediator they are to be deployed.
- 2 The Deployment Manager on CentraSite prepares the asset for deployment (the specific preparation steps depend on the type of asset being deployed) and invokes the deployer service on the Mediator. The prepared asset is submitted as input to this service.
- 3 The deployer service deploys the asset in Mediator.
- 4 If the deployment is successful, the deployer service returns a success message and data that is pertinent to the deployed asset. If the deployment is unsuccessful, the deployer service returns a failure message.
- 5 The Deployment Manager on CentraSite logs information about the deployment in the Deployment log. If the deployer service returned specific data about the asset, the asset's metadata is updated as needed in CentraSite.

General Guidelines for Effective Deployment of Virtual Services

- CentraSite automatically validates the service's run-time policy (or policies) to ensure that:
 - Any action (for example, Log Invocation) that appears in a policy multiple times is allowed to appear multiple times.
 - All action dependencies in a policy are properly met.

CentraSite informs you of any violation and you have to correct the violations before deploying the service.

- **You must make modifications to deployed assets in CentraSite.**

If you have to modify a virtual service that is already deployed, you must modify it in CentraSite and then redeploy it to Mediator to synchronize the changes.

- **You cannot make changes to a runtime policy while it is active.**

To make changes to a policy after it has been switched to the active state, you must do one of the following:

- Switch the policy to the Suspended state (to deactivate it), update the policy, and then switch it back to the Productive state (to reactivate it).
- Create a new version of the policy, make your changes to the new version of the policy, and then switch the new version to the Productive state. Switching the new version of the policy to the Productive state will automatically Retire (and deactivate) the old version.

If you have to update a runtime policy that is already deployed with virtual services that are in production, always use the second method described above (that is, create a new version of the policy). If you use the first method, which requires you to suspend the existing policy, the production services will be running without the policy while you are making revisions to it.

- **When you deploy a virtual service, CentraSite generates a VSD.**

When you deploy a virtual service to a Mediator, CentraSite generates an XML document called a virtual service definition (VSD). The VSD defines the virtual service for Mediator, and contains all the run-time policies and resources required to deploy the virtual service to Mediator.

- **You should not manually edit the endpoint information for virtual services.**

CentraSite automatically updates the service's CentraSite endpoint to its Mediator endpoint. You can view the Mediator endpoint on the virtual service's detail page in CentraSite. Because the endpoint information for virtual services is generated and updated by CentraSite, unlike when managing Native Services, you should not manually add endpoints to a virtual service. Instead, allow CentraSite to generate and manage the endpoints for the virtual services that you deploy.

However, you can deploy multiple virtual services for a single Native Service to make the service available over multiple transports and security mechanisms.

- **If deployment fails, the status is set to Failed and the failure is logged.**

If Mediator encounters a problem deploying or redeploying a virtual service, it sets the service's Deployment Status to Failed and sends a message to CentraSite describing the problem. This failure is also logged to Mediator. In this case, the CentraSite Administrator or Mediator Administrator has to take corrective action and redeploy the service manually from CentraSite.

If the reason for the failure is that the Mediator instance is unavailable, and then you restart the Mediator instance, it loads all information about any previously deployed assets.

Conditions that Must be Satisfied for Effective Deployment of Virtual Services

To deploy a virtual service to a Mediator, the following conditions must be met:

- Ensure that you have the Mediator Publisher role. Only users with this permission can deploy a virtual service. CentraSite will not enable the deployment controls for any other users.
- Ensure that the run-time policies for the virtual service are active. This is indicated in the **Policies** profile in the Virtual Service's detail page. If a policy is inactive, you must activate it.
- Ensure that the Mediator gateway to which the virtual service will be deployed has already been created.
- Ensure that the Mediator gateway's specified deployment URL is active and the user credentials of Integration Server are valid. To check this, go to the Mediator gateway's detail page and click **Publish**. If the connection is not active and valid, activate the deployment endpoint and modify the user credentials as required.
- If the virtual service is under the control of an active lifecycle model (LCM), make sure that:
 - The virtual service is in a Deployable lifecycle state. If you are not certain of what the Deployable lifecycle state is, consult your CentraSite Administrator.
 - The virtual service has a design-time policy that includes the Change Deployment Status action and it is set to Yes. This action specifies whether the service is eligible for deployment.

If these conditions are not satisfied, all or part of the deployment user interface controls will be disabled when you view the virtual service.

Deployment using CentraSite Business UI

In CentraSite Business UI, Mediator supports the following deployment methods:

- The **Web Services Stack (WSSTACK)** Mediation:
 - Axis framework based and therefore provides limited capabilities
 - Supports both SOAP and REST APIs
 - Default configuration
- The **Axis Free** Mediation:
 - In addition to the capabilities of WSSStack Mediation, Axis Free provides support for HTTP patch method.
 - Supports both REST and OData APIs

The deployment method for Virtual REST Service assets is defined in CentraSite using the configuration parameter, `RestServiceStack`, in the **centrasite.xml** customization file. By default, this parameter is set to `wsstack`.

To change the default method to Axis Free Mediation, you need to set the `RestServiceStack` parameter to `Axis-free`. When the deployment method is set to `Axis-free`, any new Virtual REST Service that is published to Mediator goes to the Axis Free Mediation. When the same service is republished to Mediator, CentraSite considers the mediation stack that the service has already been published to.

If a value is not specified for the `RestServiceStack` parameter, then the Virtual REST Service goes to WSSStack Mediation.

Note:

Republishing a Virtual REST Service does not modify its mediation stack. To change the mediation stack for an already deployed service, it needs to be unpublished first, and then republished.

Note:

The republished Virtual REST Services do not show any different behavior because of switching between the mediation stack. However, if CentraSite is publishing a Virtual REST Service to Mediator 9.10 or earlier versions, it publishes and republishes the Virtual REST Services to the WSSStack Mediation stack.

Deployment using CentraSite Control

In CentraSite Control, Mediator supports only Web Services Stack (WSSTACK) Mediation for deployment.

The **WSSTACK** Mediation:

- Axis framework based and therefore provides limited capabilities
- Supports both SOAP and REST web service APIs
- Default configuration

The deployment method for virtual services is defined by the configuration parameter, `restServiceStack`, in the customization file, **centrasite.xml**. By default, this parameter is set to `wsstack`. Any new Virtual Service asset that is published to Mediator goes to WSSStack Mediation.

Runtime Governance with API Gateway

This section provides the important information you need for designing and configuring your CentraSite's runtime governance environment to use webMethods API Gateway.

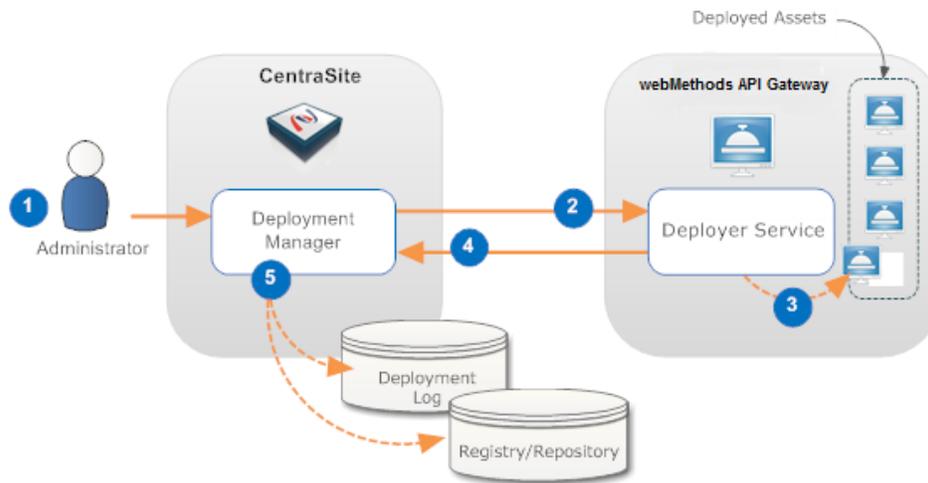
API Gateway Deployment Model

The deployment process is carried out by a sequence of interactions that occur between CentraSite and API Gateway:

1. CentraSite pushes the virtual service that is ready for deployment to API Gateway.
2. API Gateway deploys the virtual service that was received from CentraSite along with its effective runtime policy.

The deployment process is initiated from CentraSite and is carried out by the Deployer service on API Gateway.

The diagram demonstrates the deployment process of virtual services to API Gateway.



Step Description

- 1 An administrator initiates the deployment by selecting the assets that are to be deployed and specifies to which API Gateway they are to be deployed.
- 2 The Deployment Manager on CentraSite prepares the asset for deployment (the specific preparation steps depend on the type of asset being deployed) and invokes the deployer service on the API Gateway. The prepared asset is submitted as input to this service.
- 3 The deployer service deploys the asset in API Gateway.
- 4 If the deployment is successful, the deployer service returns a success message and data that is pertinent to the deployed asset. If the deployment is unsuccessful, the deployer service returns a failure message.
- 5 The Deployment Manager on CentraSite logs information about the deployment in the Deployment log. If the deployer service returned specific data about the asset, the asset's metadata is updated as needed in CentraSite.

Note:

When a virtual service is published to API Gateway, you cannot modify the runtime policy configuration for the virtual service in CentraSite. You can modify the runtime policy configuration through API Gateway only. However, to modify the runtime policy configuration of a virtual service through CentraSite, you must unpublish the virtual service from API Gateway.

Conditions that Must be Satisfied for Effective Deployment of Virtual Services

To deploy a virtual service to a API Gateway, the following conditions must be met:

- Ensure that you have the API Gateway Publisher role. Only users with this permission can deploy a virtual service. CentraSite will not enable the **Publish** icon for any other users.
- Ensure that the API Gateway asset to which the virtual service will be deployed has already been created.
- Ensure that the API Gateway's specified deployment URL is active and the user credentials of Integration Server are valid. To check this, go to the API Gateway's detail page and click the **Publish** button. If the connection is not active and valid, activate the deployment endpoint and modify the user credentials as required.
- If the virtual service is under the control of an active lifecycle model (LCM), make sure that:
 - The virtual service is in a Deployable lifecycle state. If you are not certain of what the Deployable lifecycle state is, consult your CentraSite Administrator.
 - The virtual service has a design-time policy that includes the Change Deployment Status action and it is set to Yes. This action specifies whether the service is eligible for deployment.

If these conditions are not satisfied, all or part of the deployment user interface controls will be disabled when you view the virtual service.

Enabling API Editing in API Gateway

When CentraSite run-time aspects are enabled, you cannot edit the details of an API from API Gateway if the API is published from CentraSite. However, you can enable the editing of APIs in API Gateway using the `APIGatewayDeploymentSettings` setting located in the `centrasite.xml` file.

➤ To enable API editing in API Gateway

1. Open the customization file, `centrasite.xml`, in a rich text editor.

You can find the **centrasite.xml** file located in the directory `CentraSiteInstall_Directory\cast\cswebapps\BusinessUI\custom\conf`.

2. Locate the `APIGatewayDeploymentSettings` setting in the file and set its value of `APIMetadata readonly` to `false`.

```
<APIGatewayDeploymentSettings>
  <APIMetadata readonly="false" />
</APIGatewayDeploymentSettings>
```

3. Save the customization file.
4. Restart Software AG Runtime for the change to take effect.

You can now edit the details of an API from API Gateway.

Virtual Service Asset Management

A Virtual Service runs on runtime gateway, for example webMethods Mediator, and acts as the consumer-facing proxy for a service that runs elsewhere on the network. You can create a Virtual Service asset for the SOAP, REST, and OData Web services. A Virtual Service provides a layer of abstraction between the service consumer and the service provider, and promotes loose coupling by providing location, protocol, and format independence between the consuming application and the provider service.

For example, Virtual Services enable you to:

- Move native services to other physical addresses or switch providers without affecting existing consumer applications.
- Bridge differences (for example, transport differences, message structure differences) between the capabilities of a consuming application and the requirements of a native service.
- Block portions of a service interface from certain consuming applications (that is, expose selected portions of the native service to certain consumers).
- Provide access to different versions of a service through a single endpoint.

You use CentraSite to define Virtual Services and to deploy them on specified Mediators. After you deploy a Virtual Service, you use CentraSite as a dashboard from which to view performance metrics and other run-time data relating to the usage of a Virtual Service.

Services You Should Virtualize

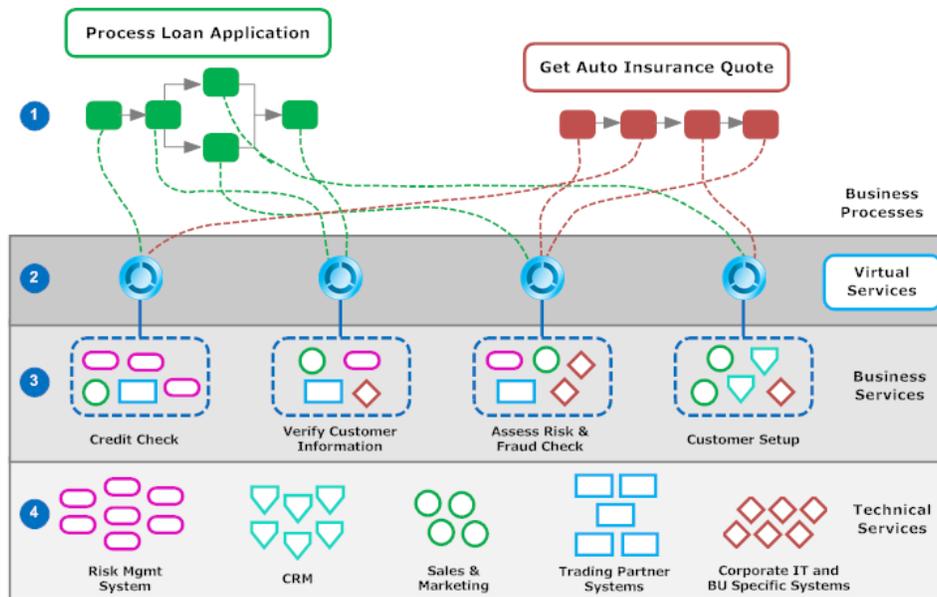
Although it is possible to virtualize any native service that is registered in CentraSite, you will generally virtualize only certain types of services. The use of virtual services creates an additional hop in the execution path and also consumes resources from an execution perspective. It is generally not practical (or beneficial) to virtualize every native service in your environment. With respect to virtualization, you want to strike a balance between the need to provide an SOA infrastructure that is flexible and extensible with the need to maintain a manageable infrastructure that is not overly complex.

Broadly speaking, there are three types of services you should consider virtualizing:

- Business services (which you virtualize at the point of consumption)
- Shared services
- Services that are provided and consumed in different domains of control (for example, cloud computing)

Virtualizing Business Services at the Point of Consumption

As shown in the following diagram, one approach to virtualization is to think of your services in terms of *business services* and *technical services* and to virtualize those services that are business services:



Description

- 1 *Business processes* are end-user applications that perform high-level tasks within your enterprise (for example, fulfilling an order or generating a quote). Business processes provide business functionality by orchestrating operations provided by different business services (depicted in layer 3).
- 2 *Virtual services* run in the layer between the business processes and business services. Each virtual service functions as a proxy for a particular business service.
- 3 *Business services* are coarse-grain services that perform business-related tasks, such as performing a credit check, setting up a new customer account or checking the status of an order. Business services generally perform their work by invoking the operations of many different technical services (depicted in layer 4).

Because business services represent the point of consumption by end-user applications and processes, they are good candidates for virtualization. Additionally, there are generally far fewer business services than technical services (typically, 10% to 15% of services in an SOA environment are business services).

- 4 *Technical services* are fine-grained services that perform low-level tasks and utility functions such as updating the employee database, retrieving a customer record or executing a query against the order database. Often, a technical service provides access to the functionality of a specific back-end system such as a CRM system, an order-entry system or a financial system.

Note:

There are cases when you might want to virtualize a technical service. However, these situations are rare. Generally speaking, you should avoid virtualizing technical services unless they are used by consumers in multiple functional domains.

Virtualizing Shared Services

A shared service is a service that is used by multiple functional domains within an enterprise. For example, consumers in the CRM area, the Sales and Marketing area and the Risk Management area might each need access to customer data. Instead of giving these systems direct access to the data service for the customer database, you virtualize the service and give these consumers access to the virtual service (or virtual services). Virtualizing the service gives you greater control over the interface that is exposed to these consumers, enables you to accommodate differences among the consumers by applying different run-time policies and/or processing steps to them and also gives you the flexibility to make modifications to the native service without impacting existing consumers.

Virtualizing Services that are in Different Domains of Control (for example, Cloud Computing)

Any service that is provided by an entity outside of the enterprise or is consumed by an entity outside of the organization should be virtualized. For example, if you have an outside service that provides sales forecasts for your industry, virtualizing this service would enable you to:

- Monitor the performance and availability of the service, including compliance with service-level agreements (SLAs).
- Shield consumers from changes in service providers.
- Track dependencies between the external service and the applications within the enterprise that consume the service.
- Resolve protocol and format inconsistencies between the outside service and the consuming applications within your enterprise.

Similarly, you should virtualize any service that your organization offers to applications that execute outside the enterprise (for example, an inventory control service that you extend to suppliers and/or distributors).

The Basic Elements of a Virtual Service

A virtual service is a Web service that runs on webMethods Mediator. You use CentraSite to create, edit, publish, and manage virtual services. Virtual services have the following major elements:

- **Basic service metadata and WSDL.** When you create a virtual service, the metadata from the native service is copied to the virtual service. The WSDL from a native SOAP-based Web service is also copied to the virtual service. After the virtual service is generated, you can edit its metadata and/or the WSDL as necessary.

Note:

When the metadata of a native service is modified, for example, using an updated WSDL, the changes are not propagated to its derived virtual services. To have the same metadata changes in one or more of its derived virtual services, you must attach the updated WSDL to each of its required virtual services.

- **A set of processing steps.** Every virtual service includes a set of processing steps that you configure before deploying the virtual service. The processing steps specify how the virtual

service will handle the requests it receives from consuming applications. Processing steps are discussed in more detail later in this.

- **One or more targets.** The **Deployment** profile for a virtual service in CentraSite Control specifies the targets (webMethods Mediators) on which the virtual service is deployed.
- **Run-time policies associated with the virtual service.** The **Policies** profile for a virtual service identifies the run-time policies that apply to the virtual service. These run-time policies are the ones that CentraSite will include when it deploys the virtual service to the Mediators specified on the **Deployment** profile.
- **Performance and Event profiles.** The **Performance** and **Events** profiles enable you to examine the run-time data associated with a virtual service. You use these profiles to view performance metrics for a specified time period and to view events that have been logged for the virtual service (for example, SLA violations, service failures, logged request/response messages and so forth).

Processing Steps of a Virtual Service in CentraSite Control

The processing steps associated with a virtual service determine how the virtual service handles the requests it receives from consumer applications. All virtual services have four processing steps:

- Entry Protocol step
- Request Processing step
- Routing step
- Response Processing step

You configure these steps to specify how Mediator is to act upon the requests it receives for this virtual service.

Entry Protocol Step

The Entry Protocol step specifies the protocol (JMS, HTTP, or HTTPS) in which the virtual service accepts requests. This step allows you to bridge protocols between the consuming application and the native service. For example, let's say that you have a native service that is exposed over JMS and a consuming application that submits SOAP requests over HTTP. In this situation, you can configure the virtual service's Entry Protocol step to accept HTTP requests and configure its Routing Step to route the request to the native service using JMS.

Besides using the Entry Protocol step to resolve protocol differences between the consumer and the native service, you might use this step to intentionally expose a virtual service over a particular protocol. For example, if you have a native service that is exposed over HTTP, you might expose the virtual service over JMS simply to gain the asynchronous-messaging and guaranteed-delivery benefits that one gains by using JMS as the message transport.

Request Processing Step

The Request Processing step specifies how the request message is to be transformed or pre-processed before it is submitted to the native service. You can configure this step to perform message transformations using a specified XSLT file or by calling the webMethods IS service (that is, a

webMethods Integration Server service running on the same Integration Server as webMethods Mediator).

You can use this processing step to accommodate differences between the message content that a consuming application is capable of submitting and the message content that a native service expects. For example, if the consuming application submits an order record using a slightly different structure than the structure expected by the native service, you can use the Request Processing step to transform the record submitted by the consuming application to the structure required by the native service.

Routing Step

The Routing step specifies the endpoint to which requests are to be routed and the protocol (HTTP or JMS) by which they are to be submitted to the native service.

If the native service is exposed over JMS, you use the routing step to specify the queue to which the Mediator is to submit the request and the destination to which the native service is to return the response.

If the native service is exposed over HTTP or HTTPS, you can configure this step to route all requests to a specified endpoint (*straight through* routing), route requests to different endpoints based on the content of the request (*content-based* routing), route requests to different endpoints based on factors such as the time of day or the requestor's IP address (*context-based* routing) or distribute requests across multiple endpoints (*load-balancing* routing).

Note:

When you configure the Routing step, you can either manually type the endpoint of the native service or you can select the endpoint from a list of known endpoints in the registry. As a best practice, you should always select the endpoint rather than typing it manually. The act of selecting an endpoint establishes a relationship between the virtual service and the native service that is hosted at the selected endpoint. This relationship is rendered when you examine the virtual service or the native service using the Impact Analysis feature.

Using the Routing Step to Direct Requests across Multiple Endpoints

If you have a native service that is hosted at two or more endpoints, you can use the load balancing option in the Routing Step to distribute requests among the endpoints or you can use the content-based or context-based options to route different types of messages to different endpoints.

Using the content-based routing option, you can route messages to different endpoints based on specific values that appear in the request message. You might use this capability, for example, to determine which operation the consuming application has requested and route requests for complex operations to an endpoint on a fast machine.

Using the context-based routing option, you can route messages based on criteria such as the time of day and/or the identity of the consuming application. For example, you might use this capability to route requests from certain high-priority consumers to endpoints on a fast machine.

Note:

With either option, you must provide a default endpoint to which the virtual service can route requests that do not satisfy any of the specified criteria.

Response Processing Step

The Response Processing step is similar to the Request Processing step. This step specifies how the response message from the native service is to be transformed or processed before it is returned to the consuming application. Like the Request Processing step, you can configure the Response Processing step to perform message transformations using a specified XSLT file or by calling the webMethods IS service. You can also use this step to return a customized error message to the consuming application when a SOAP fault occurs. (CentraSite provides a set of context variables that you can use to incorporate specific details about the transaction into the error message. You might use these variables to include information such as the time and date of the error, the consumer identifier, and/or the requester user ID.)

Configuring CentraSite for Virtual Services

Before you can create and deploy Virtual Services on your instance of CentraSite, there are two important configuration steps that you must perform.

- *You must define a gateway object for each instance of webMethods Mediator that CentraSite uses.*
- *You must define a lifecycle model for services and virtual services.*

Defining Gateways

Before you can deploy virtual services, you must define gateways to represent the Mediators that are attached to your instance of CentraSite. For example, if you will be deploying virtual services to two different Mediators, you must create two gateway objects, one for each Mediator instance.

Note: CentraSite will not enable the **Deploy** button on the **Deployment** profile for a virtual service until at least one gateway has been defined on your instance of CentraSite.

Understanding the Lifecycle for Services and Virtual Services

A virtual service is a specialized form of a Service asset type. Because virtual services are actually service objects, a lifecycle model that applies to services applies to virtual services as well. Yet a virtual service and a native service have distinctly different lifecycles. To accommodate this difference, you must create a lifecycle model that defines two separate lifecycle paths.

The following diagram shows a simple lifecycle model that supports both types of services. Note that this lifecycle has a path for native services and a path for virtual services. Policies are used to switch native services and virtual services to the appropriate path.

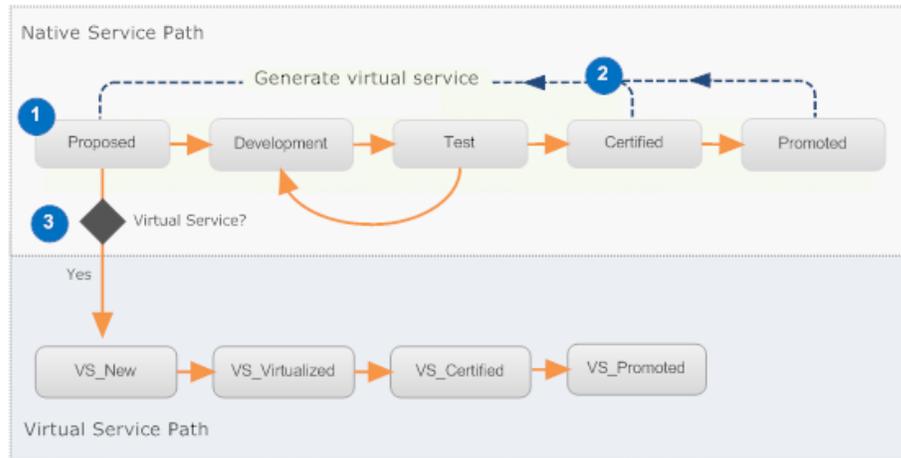
Note:

To distinguish virtual services from native services (that is, regular SOAP-based Web services, REST services or OData services), CentraSite adds the **CentraSite VirtualTypes: Virtual services** classifier to a virtual service. This classifier enables you to create design/change-time policies that target virtual services specifically.

In the following example, the **Proposed** state is the lifecycle model's initial state. When a native service is created, it enters the Proposed state and from there, it follows the lifecycle path for native services. After a native service is tested and it is ready to be promoted for production, a virtual service is generated for it. The virtual service initially enters the Proposed state when it is created.

However, a design/change-time policy immediately switches the virtual service to the lifecycle path for virtual services.

Simple Lifecycle of a Virtual Service on the Creation CentraSite



Description

- 1 The **Proposed** state is the initial state for this lifecycle model. When a native service is created, it enters the Proposed state and follows the lifecycle path for native services.
- 2 A virtual service is generated from the native service when the service is ready to go to production. Generally, this step is performed after the native service has been tested and is considered ready for production or after it has been promoted to the production environment.
- 3 The virtual service enters the lifecycle in the Proposed state, but a policy immediately switches it to the **VS_New** state, which is the beginning of the lifecycle path for virtual services. This lifecycle path includes states that enable or disable the deployment of the virtual service.

Note:

To prevent users from manually switching a native service to the lifecycle path for a virtual service, you can apply a policy to the **VS_New** state to verify that only service assets classified as **CentraSite VirtualTypes: Virtual services** enter this path.

Creating the Lifecycle Model for Services and Virtual Services

To create a lifecycle model that supports both native services and virtual services, you must perform the general steps described below.

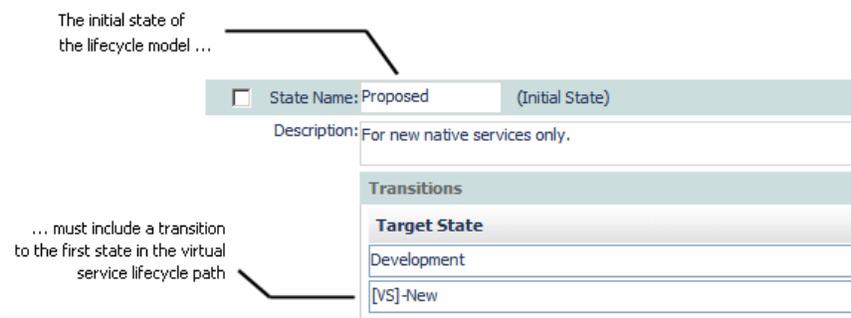
1. Create a lifecycle model for the Service asset type and in this model define the sequence of states and transitions that make up the lifecycle path for a native service.

- In the same lifecycle model (and following the sequence of states that you defined in the previous step), define the sequence of states and transitions that make up the lifecycle for a virtual service.

Note:

The lifecycle path for a virtual service must include at least one state that represents the point where the virtual service has been completely configured and is ready to be deployed. Before creating the lifecycle path for a virtual service, ensure that your lifecycle path includes the appropriate deployment-related states and policies.

- Define a transition from the initial state of the lifecycle model to the first state in the lifecycle path for virtual services. This will be the only transition that should connect the two lifecycle paths. In the following example, this is accomplished by allowing a transition from the **Proposed** state, which is the initial state for the entire model, to the **VS_New** state, which is the first state in the lifecycle path for a virtual service.



- Apply a policy to the lifecycle model's initial state (PostStateChange) that switches the state of a virtual service to the first state in the lifecycle path for virtual services. Use the Classification filter to scope the policy so that it executes only for virtual services.

In the preceding example, Simple Lifecycle of a Virtual Service on the Creation CentraSite, this policy executes on the PostStateChange for the **Proposed** state and switches virtual services to the **VS_New** state.

- Optionally, create a policy that executes on the **VS_New** state (PreStateChange) and verifies that the service includes the CentraSite **VirtualTypes: Virtual services** classifier. Doing this will prevent someone from inadvertently switching a native service to the virtual service lifecycle path.

Defining a Lifecycle Path that Enables Deployment of a Virtual Service

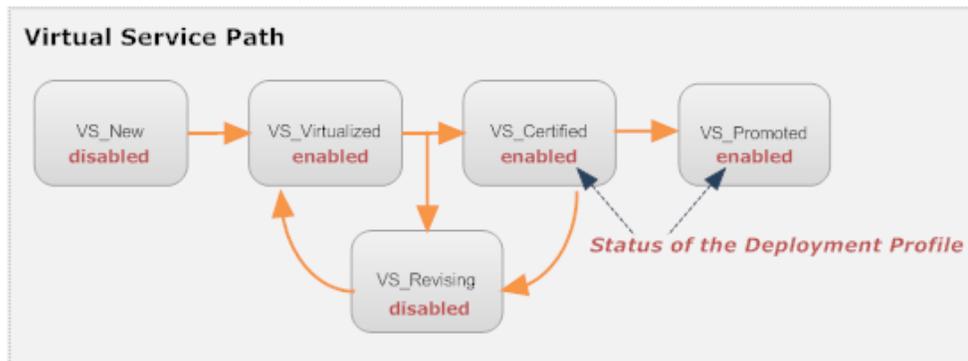
The virtual service's **Deployment** profile contains the controls that you use to deploy, undeploy and redeploy the virtual service. When the **Deployment** profile is disabled, you cannot perform these operations on the virtual service.

By default, the **Deployment** profile is disabled when you create a virtual service. This is to prevent anyone from deploying the virtual service until after its processing steps have been properly configured. To enable the **Deployment** profile, you must switch the virtual service to a state that triggers the execution of a policy that enables the **Deployment** profile.

When you define the lifecycle path for a virtual service, you must determine during which states the **Deployment** profile will be enabled and during which states it will be disabled. Then, you must create policies to enable or disable the **Deployment** profile as appropriate when the virtual service enters these states.

For example, if you wanted the **Deployment** profile to behave as shown in the lifecycle path below, you would apply a policy that enables the **Deployment** profile when the virtual service switches to the **VS_Virtualized**, **VS_Certified** or **VS_Promoted** state, and you would apply a policy to disable the **Deployment** profile when the virtual service switches to the **VS_Revising** or **VS_New** state.

Virtual Service Lifecycle Path with Deployment Status



Creating a Policy that Enables the Deployment Profile

To enable the **Deployment** profile for a virtual service, create a policy that contains the following action and apply this policy (on a PostStateChange) to all states during which you want the controls on the **Deployment** profile to be enabled.

```
Change Deployment Status (enable)
```

If you want to prevent users from modifying the processing steps for a virtual service after the **Deployment** profile is enabled, include the Processing Step Status action in the policy to disable the **Processing Step** profile as shown below.

```
Processing Step Status (disable)
Change Deployment Status (enable)
```

Creating a Policy that Disables the Deployment Profile

To disable the **Deployment** profile, create a policy that contains the following action and apply this policy (on a PostStateChange) to all states during which you want the controls on the **Deployment** profile to be disabled.

```
Change Deployment Status (disable)
```

If your lifecycle includes policies that automatically disable the **Processing Step** profile when the **Deployment** profile is enabled, this policy should include the Processing Step Status action to re-enable the **Processing Step** profile as shown below.

Change Deployment Status (disable)
Processing Step Status (enable)

Creating Virtual Services

To create a virtual service in CentraSite, you must select the native service for which you want to create the virtual service and run the Virtualize command. During the virtualization process, CentraSite copies the metadata (including the WSDL) from the native service to the virtual service. In other words, the virtual service is basically cloned from the native service.

Because the virtual service is cloned from the native service, it has its own copy of the service metadata. If you make a change to the metadata in the native service after you generate the virtual service, you will need to explicitly update the virtual service if you want that change reflected in the virtual service, too.

Note:

If the native service includes file attributes that refer to documents in the supporting document library, the virtual service will reference the same documents. CentraSite does not create separate copies of the supporting documents for the virtual service. The virtual service simply refers to the same supporting documents as the native service.

When Should You Create a Virtual Service?

Generally speaking, you do not want to generate the virtual service unless the following conditions are satisfied:

- The interface for the native service is completely implemented and that interface is reflected in the WSDL that is registered for the service in CentraSite.
- An instance of the native service is deployed and running at a known point in network.
- The metadata for the native service is valid and up-to-date. If the metadata for the native service has not been completely specified or is out-of-date, you should update it before you generate the virtual service so that you do not carry inaccurate/incomplete data into the virtual service.

Important:

Take care when assigning names to your virtual services. The name given to a virtual service when it is created, cannot be changed afterwards.

Who Should Create a Virtual Service?

If a user has View permission on a native service and Create Assets permission within their own organization, he or she can create a virtual service. However, the user will not be permitted to configure the processing steps for the virtual service unless he or she also has the Manage Runtime Policies permission for their organization. Only users with Manage Runtime Policies permission can configure these steps.

Consider identifying a small group of users who will be responsible for configuring the processing steps for a virtual service. Give this group a role that includes the Manage Run-time Policies permission. Because these users might configure virtual services that other users have created, they will also need Modify permission on the virtual services. To ensure that these users can edit

the virtual services that they need to configure, consider creating a design/change-time policy that automatically gives this group Modify permission on a virtual service when it is created.

Virtual Service Ownership

One issue to consider when creating virtual services is the issue of ownership. When you create a virtual service, CentraSite automatically adds the virtual service to *your organization* (even if the native service itself belongs to another organization). You cannot explicitly specify the organization to which you want the virtual service added.

The issue of ownership is important with respect to virtual services, because it determines which run-time policies are applied to the virtual service when it is deployed. If the native service belongs to another organization, the existing run-time policies for your organization might or might not be appropriate for it.

When you define the general process that your site will follow to create and deploy virtual services (that is, when you determine who will create a virtual service, who will configure a virtual service, and who will deploy a virtual service), keep in mind that CentraSite always adds a virtual service to the organization of the user who creates it. Make sure that whatever process you adopt for creating virtual services places a virtual service in the appropriate organization.

Deploying a Virtual Service

There are several ways you can deploy a virtual service to a Mediator instance. All methods except the first one allow you to deploy multiple virtual services in a single step.

- From the virtual service's detail page.
- From the **Operations > Deployment** page.
- Running a script file from a command line.
- Running a batch file.

To deploy a virtual service, the following conditions must be satisfied:

- Ensure that you have the Mediator Publisher Role or the publish permission on the gateway. Only users with this permission can deploy a virtual service. CentraSite will not enable the deployment controls for any other users.
- Ensure that the run-time policies for the Virtual Service are active. This is indicated in the **Policies** profile on the Virtual Service's detail page. If a policy is inactive, you must activate it.
- Ensure that the Virtual Service has a design-time policy that includes the Change Deployment Status action and it is set to Yes. This action specifies whether the service is eligible for deployment.
- Ensure that the Virtual Service has at least one gateway associated with it, and the gateway must already have been created.
- Ensure that the gateway's specified deployment URL is active and the user credentials of Integration Server are valid. To check this, go to the gateway details page and click the **Check**

Connection button. If the connection is not active and valid, activate the deployment endpoint and modify the user credentials as required.

- Ensure that the Virtual Service is in a deployable lifecycle state. If you are not certain in which lifecycle states a Virtual Service is eligible for deployment, consult your CentraSite administrator.

If these conditions are not satisfied, all or part of the deployment user interface controls will be disabled when you view the virtual service.

Note:

Only users that are completely familiar with your site's mediation environment should be given permission to deploy virtual services. Generally, this would include a small number of administrators who have operational responsibility for the Mediators on which virtual services are deployed.

The Deployment Process

The deployment process is carried out by a sequence of interactions that occur between CentraSite and the Mediator:

1. CentraSite pushes the Virtual Service that is ready for deployment to the webMethods Mediator gateway.
2. Instantly, the Mediator deploys the Virtual Service that was received from CentraSite (along with its effective run-time policy), and notifies CentraSite when the deployment process is complete.

Undeploying a Virtual Service

After you deploy a virtual service to a Mediator, the virtual service remains deployed and active on that Mediator until you manually undeploy. You can deploy a virtual service using the same deployment mechanisms mentioned above.

Redeploying a Virtual Service

A virtual service that is already deployed on a Mediator can be manually redeployed. You can redeploy a virtual service using the same deployment mechanisms mentioned above. If you make changes to a virtual service's processing steps, for example, you must manually redeploy the virtual service to put those changes into effect.

You cannot make changes to a run-time policy while it is active. To make changes to a policy after it has been switched to the active state you must do one of the following:

- Switch the policy to the Suspended state (to deactivate it), update the policy and then switch it back to the Productive state (to reactivate it).
- Create a new version of the policy, make your changes to the new version of the policy and then switch the new version to the Productive state. Switching the new version of the policy to the Productive state will automatically retire (and deactivate) the old version.

If you need to update a run-time policy that is already deployed with virtual services that are in production, always use the second method described above (that is, create a new version of the

policy). If you use the first method, which requires you to suspend the existing policy, your production services will be running without the policy while you are making your revisions to it.

Revising a Virtual Service

Web services are bound to change and evolve over time. The loose coupling principles of service-oriented architecture (SOA) imply that service providers can release a new version of a shared service without waiting for consumers to adapt, and that service consumers should test and certify on a new shared service version before switching. Consequently, you might need to have multiple versions of a shared service running concurrently and simultaneously accessible by different service consumers. Some service consumers might need to continue using an old version of a service until migration of the consumer code occurs. Therefore, Web services versioning is an important subject that should be considered carefully in all enterprise SOA approaches.

Current standards for Web services have no explicit support for versioning. However, there are two alternatives for handling access to multiple versions of Web services:

- Require the consumer applications to change their code to specify which versions to access.

This option is rarely implemented due to its prohibitively complex and time-consuming nature.

- Use a mediation layer (for example, Mediator) to decouple the consumer from the provider, and thus allow the mediation layer to route requests to the desired version of a given service.

Mediator provides versioning solutions that you can implement, called versioning patterns. To implement versioning patterns, you configure virtual services in CentraSite so that consumers can access the desired version of a given service. You can use the versioning patterns to handle access to both Minor and Major versions of services.

Mediator cannot run multiple versions of the same virtual service simultaneously. Mediator only retains the last deployed version of a virtual service. However, suppose you have multiple versions of a native Web service. By using a versioning pattern, a single virtual service can provide access to the various native service versions based on an intelligent routing scheme that routes requests from a particular consumer to the correct native service version. A second option would be to provide multiple virtual services that correspond to multiple native service versions. For example, suppose you have two versions of the native service `GetOrder`. You have the following options in Mediator:

- Provide a single virtual service that intelligently routes each consumer to the appropriate `GetOrder` version (either version 1 or version 2).
- Provide one virtual service that routes consumers to version 1, and one virtual service that routes consumers to version 2.

Minor Versions vs. Major Versions

Minor and Major versions of Web services are described as follows:

- **Minor version:** A Minor version is a version that is compatible with all consumers of the existing virtual service. That is, the changes in a Minor version do not break the existing applications that use the service. Examples of changes for a Minor version include:
 - Bug fixes

- Performance improvements
- The addition of a supporting document
- The addition of operations (as long as it does not break the existing applications)
- A change in the Description attribute
- **Major version:** A Major version is a version that is *incompatible* with consumers of the existing virtual service. That is, the changes in a Major version “break” the existing applications that use the service. Examples of a Major version include:
 - Modifications to the namespace assignments
 - Modifications to message descriptions
 - Modifications to interface definitions and/or operation signatures in the service WSDL
 - Changes to the implementation of the service that do not explicitly affect the WSDL, but nevertheless affect the way in which an existing consumer application interacts with the service

For example, a service that returns an expanded set of result codes or generates a different form of customer ID might break an existing consumer application even if the interface defined in the service WSDL did not change.

Note:

Be aware that sometimes versioning one asset will necessitate the versioning of another. For example, if an XML schema changes and that schema is imported by a Web service, you will need to generate a new version of the XML schema and a new version of the Web service that references it.

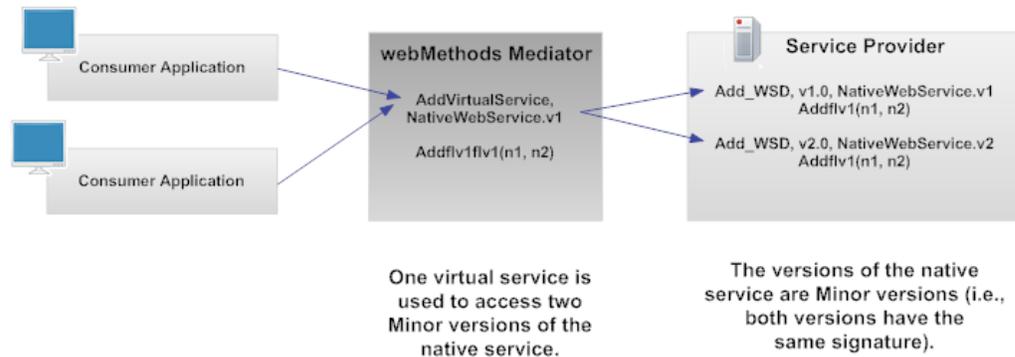
The Layer of Indirection Pattern

This pattern allows multiple Minor versions of a native service to coexist in the registry without requiring consumers to change the code in their consumer applications, and helps to ensure a graceful migration to the new Minor version.

To implement this pattern, you configure a single virtual service to route each request to the version that is appropriate for each consumer (or group of consumers). That is, you configure the virtual service's Routing step to use either the content-based routing option or the context-based routing option.

- **Content-based routing option:** Using the content-based routing option, you can route request messages to different endpoints based on specific values that appear in the request message. For example, if a new Minor version contains an additional operation, you can write a rule that routes all requests that reference the newly-added operation to the new Minor version.
- **Context-based routing option:** Using the context-based routing option, you can route request messages to different endpoints based on the identity of the consuming application. For example, if you want to allow only certain consumers to access a new Minor version, you write a rule that routes only their requests to the new Minor version.

The Layer of Indirection Pattern



The Adapter Pattern

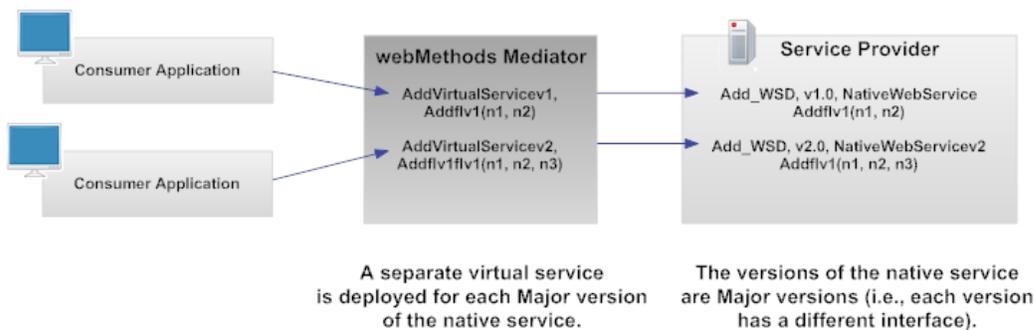
This pattern allows multiple *Major* versions of a native service to coexist in the registry without requiring consumers to change the code in their consumer applications, and helps to ensure a graceful migration to the new Major version.

This is called the adapter pattern because Mediator will act as an adapter, adapting the client requests before they are submitted to the native services.

Unlike the Layer of Indirection pattern, which has *one* virtual service that can access each Minor version of the native service, this pattern has *a separate virtual service for each Major version of the native service*.

To implement this pattern, you configure the virtual service's Request Processing step so that it transforms the endpoint specified in a request to the endpoint of the desired version. The Request Processing step specifies how the request message is to be transformed before it is submitted to the native service. You can configure this step to perform message transformations using a specified XSLT file or by passing the message to a webMethods IS service (that is, an Integration Server service running on the same Integration Server as webMethods Mediator).

The Adapter Pattern



Combination of the Layer of Indirection Pattern and the Adapter Patterns

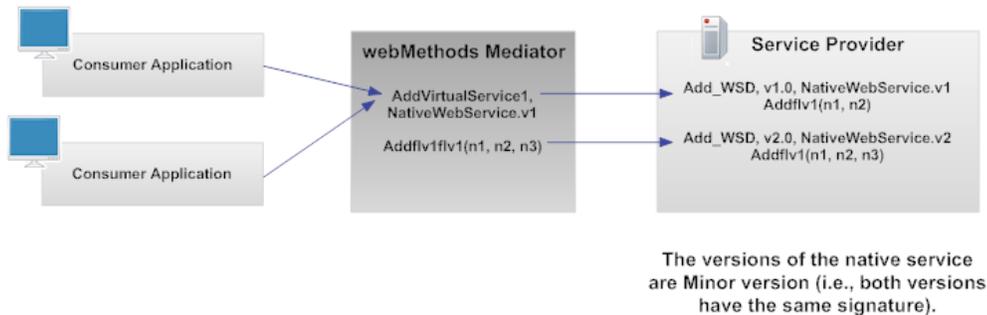
This pattern has one virtual service that can access:

- Multiple Major versions of a native service (that is, utilizing the Adapter pattern).

Thus you configure the virtual service's Request Processing step so that it transforms the endpoint specified in a request to the endpoint of the desired version.

- Thus you configure the virtual service with either the content-based routing option or the context-based routing option in order to route requests to the Minor version that is appropriate for each consumer (or group of consumers).

Combination of the Layer of Indirection Pattern and the Adapter Pattern



Managing Virtual Service Assets through CentraSite Business UI

This section describes operations you can perform to manage virtual service assets, such as, Virtual (SOAP) Services, Virtual REST Services, and Virtual OData Services through CentraSite Business UI.

Virtual Service Management

This section describes operations you can perform to manage Virtual (SOAP) Services through CentraSite Business UI.

Adding Virtual Service to Your Asset Catalog

To create and manage Virtual (SOAP) Service asset in CentraSite Business UI, you must have the following permissions:

- CentraSite Administrator
- Organization Administrator
- Asset Provider
- API Runtime Provider (required to configure run-time actions for the Virtual Services)
- Mediator Publisher (required to publish Virtual Services to Mediator gateways)
- API Portal Publisher (required to publish Virtual Services to API Portal gateways)

- Instance-level Modify permission for a gateway (required to publish Virtual Services to that particular gateway)

If you have the CentraSite Administrator role, you can create and manage Virtual Services within any organization.

If you have the Organization Administrator role or API Portal Administrator role for a specific organization, you have the ability to create and manage Virtual Services within that specific organization.

The following general guidelines apply when adding a Virtual Service asset in CentraSite Business UI:

- Ensure that the interface for the Native Service is completely implemented and that the interface is reflected in the WSDL or schema file that is registered for the Web Service in the CentraSite repository.
- An instance of the Web Service is deployed and running at a known point in network.
- The metadata for the Native Service is valid and up-to-date. If the metadata for the Native Service has not been completely specified or is out-of-date, you should update it before you generate the Virtual Service so that you do not carry inaccurate or incomplete data into the Virtual Service.

In CentraSite Business UI, you can add a **Virtual Service** asset to the catalog in the following ways:

- You can *create a Virtual Service from an existing Web Service, also called as a (Native) Service in CentraSite*, meaning that you create the virtual copy (proxy) of the existing Web Service using an already imported input file.
- You can *create a Virtual Service using an importer*, which is a utility that generates the Virtual Service from an imported archive file.
- You can *create a Virtual Service from scratch*, meaning that you create the Virtual Service (and set its attributes) manually.
- You can *create a Virtual Service using a command line tool*, which is a utility that generates the Virtual Service from an input WSDL file.

Adding Virtual Service using an Existing Native Service

You can create a virtual service asset from an existing native service asset. However, if you want to configure the run-time policies for the newly created virtual service asset using CentraSite, the `RuntimeComponentSetting` in the `centrasite.xml` file must be enabled. For information on enabling run-time aspects from CentraSite, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

When run-time aspects are enabled in CentraSite, then the creation of a virtual service from a native service would involve the following three steps. Else, only the first of the following three can be performed from CentraSite. In such scenario, the run-time policies can be configured from gateways to which the APIs are published.

1. Creating a virtual copy (proxy) of the existing Native Web Service asset. For procedures, see [“Create Virtual Service” on page 976](#).

2. Configuring the run-time policy actions for the Virtual Web Service asset. For procedures, see [“Assign Policy Actions for Virtual Service” on page 978](#).
3. Implementing the virtualization of the existing Native Web Service asset, and publishing the Virtual Web Service asset to one or more gateways. For procedures, see [“Virtualize and Publish Virtual Service to Gateways” on page 978](#).

Create Virtual Service

You use panel 1 of the Virtualize Your API page to specify the proxy and invocation aliases, and endpoints for the new Virtual Service.

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Web Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **Service**.
 - c. Click **OK**.

5. Click the Web Service you want to virtualize.

This opens the Web Service details page. Also, the actions bar displays a set of actions that are available for working with the Web Service.

6. On the actions bar of the Web Service details page, click **Virtualize**.
7. In the **Virtualize <Service_Name> (Step 1 of 3)** wizard, provide the required information for each of the displayed data fields.

Field	Description
Create New Virtual Alias	Name of the Virtual Service asset (also, termed as Virtual Alias).

Field	Description
	<p>The name of a Virtual Service asset must be NCName-conformant, meaning that:</p> <ul style="list-style-type: none"> ■ The name must begin with a letter or the underscore character (_). ■ The remainder of the name can contain any combination of letters, digits, or the following characters: . - _ (that is, period, dash, or underscore). It can also contain combining characters and extender characters (for example, diacriticals). ■ The name cannot contain any spaces. ■ Furthermore, if the Virtual Service name contains any non-conformant character, upon publishing the Virtual Service to any gateway, the non-conformant character is simply replaced with the underscore character (_) in Mediator. However, in CentraSite the Virtual Service name defined by you is displayed. <p>For more information about the NCName type, see http://www.w3.org/TR/xmlschema-2/#NCName</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: The name of a Virtual Service asset must be unique within an organization. If, for example, a Virtual Service with the same name already exists within the CentraSite registry, a warning message will be issued.</p> </div> <p>By default, CentraSite populates the Create a New Virtual Alias field with the display name that was specified for the Native Service.</p>
Endpoint prefix for invocation alias	<p>(Optional). Prefix for the Virtual Service.</p> <p>This field accepts URL paths.</p> <p>Example:</p> <ul style="list-style-type: none"> ■ /testmethod/myprefix/ ■ test@1234
Endpoints of <API_name> to Virtualize	<p>The Native Service endpoint you want to use for invoking the Virtual Service.</p> <p>The Endpoints list displays a list of the Endpoint URLs available for the Native Service.</p>

8. Do one of the following:

- Click **Next** and proceed to panel 2 to proceed with configuring run-time policy actions. This button is enabled only if the run-time aspects from CentraSite is enabled. For procedure to enable CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).
- Click **Virtualize** to create virtual (proxy) copy of REST Service (without publishing the newly created Virtual Service to the selected gateways).

A Virtual REST Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the Virtual REST Service that you just created is displayed.

Note:

The **Virtualize** button is displayed in panel 1, only if the CentraSite run-time aspects are not enabled.

Assign Policy Actions for Virtual Service

You use panel 2 of the Virtualize Your API page to configure policy actions for the Virtual Service.

1. Navigate to **(Step 2 of 3)** of the **Virtualize <Service_Name>** wizard.
2. Drag and drop the policy actions you want CentraSite to execute at runtime.
3. Specify the parameters for each policy action as described in [“Configure Policy Action Parameters” on page 1046](#), and then click **Next**.

Virtualize and Publish Virtual Service to Gateways

You use panel 3 of the Virtualize Your API page to configure the gateways for publishing the Virtual Service. The publishing operation allows the API Providers to expose the Virtual Service in the selected gateways. Clients can then access and examine the usage of the exposed Virtual Service.

1. Navigate to **(Step 3 of 3)** of the **Virtualize <Service_Name>** wizard.
2. In the **Gateway** list, select the gateways to publish the Virtual Service.

The available gateways are:

- API Portal
- Mediator
- Insight Server

3. In the **Sandbox** list, select the category by which the gateways are classified.

4. Select the check box of a single gateway, or select the check boxes of multiple gateways to publish the Virtual Service.
5. (Applicable for an API Portal gateway only). Configure the sandbox categories for publishing the Virtual Service to the API Portal gateways. Follow these steps:
 - a. In the field labeled **Sandbox**, click the **Configure** icon.
 - b. In the **API Portal Settings** dialog box, select the sandbox category to which you want to publish the Virtual Service.
6. (Applicable for a Mediator or Insight Server gateway only). Select the **Expose to Consumers** option button to allow clients to simply access and examine the usage of the exposed Virtual Service.
7. Do one of the following:
 - Click **Virtualize** to create virtual (proxy) copy of Web Service (without publishing the newly created Virtual Service to the selected gateways).
 - Click **Publish** to create virtual (proxy) copy of Web Service, and simultaneously publish the newly created Virtual Service to the selected gateways.

A Virtual Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the Virtual Service that you just created is displayed.
8. Configure the extended attributes of the Virtual Service as described later in this topic.

Adding Virtual Service using an Archive

Pre-requisites:

To add a Virtual Service asset to an organization's asset catalog, you must belong to a role that has the Create Assets or Manage Assets permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

Before you import a Virtual Service asset to the catalog, you must have the archive file (.zip file). This file must reside on the file system of the computer where your browser is running.

You can import a Virtual Service using the archive file (.zip file) to which the Virtual Service was previously exported. You can import Virtual Services into the same CentraSite registry from which they were originally exported or to a different CentraSite registry.

➤ To add a Virtual Service asset using importer

1. In the CentraSite Business UI activity bar, click **Create Asset**.

This opens the **Create New Asset** wizard. The bottom panel of the **Create New Asset** wizard displays the option to import an archive file.

2. To import an archive file, click **Choose** to navigate to the folder where an exported archive file of the Virtual Service asset resides, and choose the file.

When you choose a file to import, the fields in the area labeled **Basic Information** cannot be edited.

3. Click **Next**.

The **Create New Asset** wizard displays a list of the referenced objects for the Virtual Service to import. The check box next to each object indicates whether the referenced object should be imported. By default, all objects displayed are included in the import set.

4. To exclude any referenced object from the import set, clear the corresponding check box.
5. Click the **Advanced Settings** chevron to expand the additional options that are available for the Virtual Service asset. Provide values for each of the displayed options:

Option	Description
Change Owner	<p>The imported Virtual Service can be assigned to the same owner as in the source CentraSite registry, or you can assign a new owner.</p> <p>The Change Owner field is type-ahead field. As you type characters in this field, the dialog box lists the user names that match the characters you specify.</p>
Change Organization	<p>When you import a Virtual Service, you can import it into the same organization in the target CentraSite registry as in the source CentraSite registry from which they were exported, or you can assign a new organization.</p> <p>The Change Organization field is type-ahead field. As you type characters in this field, the dialog box will list the organization names that match the characters you specify.</p>
Retain lifecycle state	<p>This option determines whether the lifecycle state of the imported Virtual Service is preserved. Enable the option to retain the lifecycle state of the Virtual Service which is imported.</p>
Overwrite existing entities	<p>This option specifies that an existing Virtual Service with the same uuid in the target CentraSite registry will be overwritten, even if the Virtual Service in the archive is older than the one in the target CentraSite registry. Enable the option to overwrite the existing Virtual Service.</p>
Import groups that	<p>This option determines that when you import a user, whether you want to import the groups that the user belongs to. This applies only to user-defined</p>

Option	Description
the user belongs to	groups. System-defined groups are not imported. Enable the option to import the groups.
Ignore API keys and OAuth2 tokens	This option determines whether the API keys and OAuth2 tokens of the imported assets are to be imported into the target CentraSite registry. Enable the option to ignore the API keys and OAuth2 tokens during the import process.

- Click **Import** to import the Virtual Service.

After the import operation completes, the import wizard informs you if the import was successful or if there were any errors and warnings.

- Click **Download Import Log** to view the import logs.

The import log lists the status of all the objects stating whether they were successfully imported or if there were errors and warnings.

- Click **OK** to terminate the import wizard.

A Virtual Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the Virtual Service asset that you just created is displayed.

- Configure the extended attributes of the Virtual Service asset as described later in this topic.

Tip:

If you had previously imported an WSDL file that has an associated schema file and you now re-import just the schema file with modifications, your browser may not display the updated contents of the schema file. This can happen if the browser cache is not being updated automatically. To rectify the problem, you can change your browser settings so that pages are always updated on every visit.

Adding Virtual Service from Scratch

Pre-requisites:

To add a Virtual Service asset to an organization's asset catalog, you must belong to a role that has the Create Assets or Manage Assets permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

Before you add a Web Service asset to the catalog, you must have the WSDL specification file that you want to import. This file can reside on the file system of the computer where your browser is

running or it can reside anywhere on the network, as long as its location is addressable through a URL.

➤ To add a Virtual Service from scratch

1. In the CentraSite Business UI activity bar, click **Create Asset**.

This opens the **Create Asset** wizard.

2. In the area labeled **Basic Information**, provide the required information for each of the displayed data fields:

Field	Description
Name	<p>(Optional). Name of the Virtual Service asset.</p> <p>The name of a Virtual Service asset must be NCName-conformant, meaning that:</p> <ul style="list-style-type: none"> ■ The name must begin with a letter or the underscore character (_). ■ The remainder of the name can contain any combination of letters, digits, or the following characters: . - _ (that is, period, dash, or underscore). It can also contain combining characters and extender characters (for example, diacriticals). ■ The name cannot contain any spaces. ■ Furthermore, if the Virtual Service name contains any non-conformant character, upon publishing the Virtual Service to any gateway, the non-conformant character is simply replaced with the underscore character (_) in Mediator. However, in CentraSite the Virtual Service name defined by you is displayed. <p>For more information about the NCName type, see http://www.w3.org/TR/xmlschema-2/#NCName</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: The name of a Virtual Service asset must be unique within an organization. If, for example, a Virtual Service with the same name already exists within the CentraSite registry, a warning message will be issued.</p> </div> <p>By default, CentraSite populates the Create a New Virtual Alias field with the display name that was specified for the Native Service.</p>
Type	The asset type, Virtual Service .
Organization	The organization to which you want to add the Virtual Service. (The Organization list only displays organizations for which you have the Manage Assets permission or at least the Create Assets permission.)

Field	Description
Version	<p>(Optional). The version identifier for the Virtual Service. You can enter any string in this field, that is, the version identifier does not need to be numeric. You can also leave the field blank. You can later create new versions of the Virtual Service. The default is 1.0.</p> <p>Examples:</p> <pre>0.0a 1.0.0 (beta) Pre-release 001 V1-2007.04.30</pre>

Description (Optional). The description for the Virtual Service.

Note:

This is the description information that users will see when they view this Virtual Service asset in the CentraSite user interfaces. Therefore, we recommend that you specify a meaningful description for each Virtual Service.

Import a File The input WSDL file for the Virtual Service. You may want to read the input WSDL file from a URL-addressable location on the network (the **URL** option) or from your local file system (the **File** option).

Option	Description
URL	If the WSDL file you are importing resides on the network, you can specify its URL.
File	If the specification resides in your local file system, specify the file name. You can use the Browse button to navigate to the required folder.

- Click the **Advanced Settings** chevron to expand the additional options that are available for the Virtual Service asset. Provide the required information for each of the displayed fields:

Field	Description
Credentials	If you have specified a URL and the site you want to access through the URL requires user authentication, type a username and password for authentication at the URL site.
Resolution	<p>Select a resolution strategy, which will allow you to specify how an already existing imported and included file is handled. For each of the imported and included files you have one of these options:</p> <ul style="list-style-type: none"> ■ Create new version: Creates a new version of the file with the new content (if, for example, you want to modify a WSDL file but want to retain its previous version).

Field	Description
-------	-------------

- **Always overwrite:** Overwrites the importing file with new content.

4. Click **Next**.

You cannot navigate to the next screen unless all the required attributes have been set.

5. In the **Preview** panel, review the basic information for the Virtual Service before you add to the CentraSite registry.6. Click **Save**.

A Virtual Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the Virtual Service asset that you just created is displayed.

7. Configure the extended attributes of the Virtual Service asset as described later in this topic.

Viewing Virtual Service List

You use the Search Results page to display the list of Virtual Service assets.

➤ To view the list of Virtual Service assets

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.3. To search for the assets of type, Virtual Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

A list of scopes that are available to you is displayed. By default, CentraSite contains the following predefined scopes:

Scope	Description
Everything	Displays a list of the currently available assets, organizations, and users in CentraSite registry.
Assets	Displays a list of the currently available assets in CentraSite registry.

Scope	Description
	This is the default scope.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **Virtual Service**.
 - c. Click **OK**.

A list of defined Virtual Service assets is displayed in the Search Results page.

5. To filter the list to see just a subset of the available Virtual Service assets, type a partial string in the **Keyword** text box. Add the specified keyword to the Search Recipe, by clicking the plus symbol (+) next to the text box, or press Enter.

The Search Results page provides the following information about each Virtual Service asset:

Column	Description
Name	Name of the Virtual Service asset.
Description	The description for the Virtual Service.
Asset Type	The asset type, Virtual Service .
Last Updated Date	The date on which the Virtual Service was last modified.
Owner	The user who owns the Virtual Service.
Organization	The organization which owns the Virtual Service.
Version	The user-assigned version identifier for the Virtual Service.

To specify the sorting preference, select an attribute from the **Sort by** list.

Note:

Use the **View** menu to display the additional attributes.

Viewing and Modifying Virtual Service Details

You use the details page of a Virtual Service asset to examine and modify the WSDL specification.

The asset type **Virtual Service** has a unique set of profiles. However, an administrator can configure this asset type to display a customized set of profiles and attributes.

The following general guidelines apply when modifying the details of a Virtual Service asset in CentraSite Business UI:

- If you are not the owner of the Virtual Service, you cannot examine or modify the details of the Virtual Service, unless you have the View or Modify permission on the Virtual Service (granted though either a role-based permission or an instance-level permission).
- When you view the details page of a Virtual Service, you will only see profiles for which you have the profile-level View permission. You will only be able to modify the attributes of the profiles on which you have the Modify permission.
- To modify the name of a Virtual Service, you must first undeploy the Virtual Service.
- When you hover over an attribute, CentraSite displays the tooltip text that provides a short description of the attribute's purpose.
- Some attributes accept only specific types of information. For example, if the Virtual Service asset type includes a URL type attribute, you must supply a URL when you modify that attribute. Other attribute types that require a specific type of value include the Date and Email attributes.
- Some attributes are designed to be read-only. You cannot modify the read-only attributes even if you have the CentraSite Administrator role.

In this task you examine and modify the basic and type-specific attributes that are associated with a Virtual Service. You can also examine and modify the bindings, operations, WSDL file, associated schema files, and external links to the WSDL and schema files.

➤ **To examine and modify the details of a Virtual Service asset**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.This displays a list of assets in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Virtual Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **Virtual Service**.
 - c. Click **OK**.

A list of defined Virtual Service assets is displayed in the Search Results page.

5. Click the Virtual Service you want to examine and modify the attributes.

This opens the Virtual Service details page. Also, the actions bar displays a set of actions that are available for working with the Virtual Service.

You can hover over the **info** symbol next to an attribute to display the tooltip text, which describes the purpose of the attribute. The tooltip text displays the values of the attribute's Name, and Description fields that are contained in the Virtual Service type definition.

6. To modify the generic attributes that are displayed in the **Basic Information** profile, on the actions bar, click **Edit**. Change values of the attributes in the respective data fields as required.
7. To modify the extended attributes that are displayed in the individual profiles, follow these steps:
 - a. Select the profile that contains the attribute(s) you want to modify.
 - b. Change values of the attributes in the respective data fields as necessary.
 - c. Repeat steps 7.a and 7.b for each profile for which you want to modify the attributes.
8. Click **Save**.

The details page of a Virtual Service asset includes the following additional information:

Identification Profile (for Assets with Key-based Authentication)

Field	Description
API Key String	<i>Read-only. String.</i> The confidential secret key used to securely authenticate the consumer. The API Key String field is visible only to the consumer who requested an API key.
Expiry Date	<i>Read-only. String.</i> An expiration date for the API key.

Identification Profile (for Assets with OAuth-based Authentication)

Field	Description
Client Id	<i>Read-only. String.</i> The unique identifier that is used by the client to fetch access tokens for the virtual API.
Client Secret	<i>Read-only. String.</i> The secret key value that is used with the client identifier, serves as a password to fetch access tokens for the virtual API.
Client Name	<i>Read-only. String.</i> The name of the client (consumer application) that is attempting to get access to the virtual API.

Field	Description
Scope	<i>Read-only. String.</i> The scope value is the name of the virtual API. If the scope value is valid, Mediator obtains the access token. If no scope value is provided, Mediator provides the access token to the scope in which the client is allowed and adds the scope to the response.
Refresh Token	<i>Read-only. String.</i> The unique identifier used by the client to obtain a new access token when the current access token becomes invalid or expires.

API Key Scope Profile

Field	Description
API Service	<i>Read-only. String.</i> The name of the virtual API that is associated with the API key. To view details of the virtual API, click its hyperlinked name.

Deleting Virtual Services

If you are not the owner of a Virtual Service asset, you cannot delete the Virtual Service unless you have Full permission on the Virtual Service (granted though either a role-based permission or instance-level permission).

The following general guidelines apply when deleting a (Virtual) Service asset in CentraSite Control:

- A Virtual Service can only be deleted if it is not the target of an association from another registry object.
- When you delete a Virtual Service, CentraSite removes the catalog entry for the Virtual Service (that is, it removes the instance of the Virtual Service from CentraSite's object database). Also note that:
 - The performance metrics and event information of the Virtual Service are also deleted.

Note:

When you delete the Virtual Service, this information is deleted by the built-in action **Delete RuntimeEvents and RuntimeMetrics of Service**, which is installed with CentraSite.

- When you delete a composite Virtual Service, all of its nonshared components are also deleted.
- Deleting a Virtual Service will *not* remove:
 - Other assets to which the Virtual Service refers (unless the reference is to an asset that is a nonshared component of the Virtual Service you are deleting). For example, if you are deleting a Virtual Service which has a Consumes or Consumed By relationship with other services in the registry, the related services will not be deleted.
 - Supporting documents that are attached to the Virtual Service.

- Earlier versions of the Virtual Service. Only the latest version of the Virtual Service can be deleted; to remove earlier versions, they must be purged.
- You cannot delete a Virtual Service if:
 - The Virtual Service is in a pending state (for example, awaiting approval).
 - Any user in your CentraSite registry is currently modifying the Virtual Service.

➤ **To delete Virtual Service assets**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.This displays a list of assets in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Virtual Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **Virtual Service**.
 - c. Click **OK**.

A list of defined Virtual Service assets is displayed in the Search Results page.

5. Select one or multiple Virtual Services you want to delete.
6. On the actions bar of the Search Results page, click **Delete**.

You can also delete a single Virtual Service assets from the actions bar of its details page.

7. When you are prompted to confirm the delete operation, click **Yes**.

Note:

If you have selected a set of Virtual Services, where one or multiple Virtual Services are in a pending state (for example, awaiting approval), CentraSite ignores the pending list of Virtual Services, and deletes any remaining Virtual Services for which you have the required permission.

Exposing a Virtual Service as Virtual REST Service

CentraSite offers you the possibility to expose Virtual Services also as Virtual REST Services in webMethods Mediator.

Note:

You cannot convert a virtual service as a virtual REST service, if the CentraSite run-time aspects are not enabled.

The REST-enabled support for a Virtual Service is achieved using the **Enable REST Support** policy action that comes pre-shipped with CentraSite.

When you include the **Enable REST Support** action in a Virtual Service's run-time configuration, clients can invoke the REST-enabled Virtual Service in Mediator using both a SOAP request and a REST request.

This policy action is set by default in the **Receive** stage for all Virtual Services. To disable the REST support for a Virtual Service, manually remove the **Enable REST Support** action from the **Receive** stage of the Virtual Service.

REST-Enabled Virtual Service in webMethods API Portal

Note:

You cannot publish a REST-enabled Virtual Service to webMethods API Portal, if the CentraSite run-time aspects are not enabled.

When you publish a REST-enabled Virtual Service to webMethods API Portal, it is exposed only as a Virtual REST Service in API Portal. All the operations of the Virtual Service are exposed as REST Resources. This allows clients to test the exposed Virtual REST Service in API Portal using a REST request.

Currently, API Portal only supports the HTTP GET and POST methods with these kinds of REST Resources.

Limitations:

- If a SOAP operation exposed as a REST resource is invoked using a GET request, the required values for the SOAP operation should be sent as Query parameters. But these Query parameters are not pre-populated with the exposed REST resource. You must manually specify them.
- It is assumed that the default content-type is `application/json`.

Virtual REST Service Management

This section describes operations you can perform to manage virtual REST services through CentraSite Business UI.

Virtual REST Service Compatibility

Beginning with version 9.7, CentraSite supports the enhanced interface for Virtual REST Services (in contrast, earlier versions of CentraSite supported a standardized interface for Virtual REST

Service). Documentation of the prior Virtual REST Service interface is available to CentraSite customers who have a current maintenance contract in Empower Product Support website.

- **If you Migrate Virtual REST Services from a Pre-9.7 Release:** If you have Virtual REST Services that were created prior to CentraSite version 9.7, these Virtual REST Services will continue to hold the version's metadata in the enhanced Virtual REST Service interface implemented by current version of CentraSite.

To invoke the migrated Virtual REST Service, the client would use the following URL syntax:

```
http://localhost:5555/ws/VSName/ResourceName
```

- **If you Migrate Virtual REST Services from a 9.7 Release:** If you have Virtual REST Services that were created in the CentraSite version 9.7 using the CentraSite Business UI, these Virtual REST Services will again continue to hold the version's metadata in the enhanced Virtual REST Service interface implemented by current version of CentraSite. However, you will find the following information in the migrated Virtual REST Service:

- The sample request and response messages that existed under the REST Method display without changes.
- The status codes that existed under the REST Method will now display under the REST Response.

To invoke the migrated Virtual REST Service, the client would use the following URL syntax:

```
http://localhost:5555/ws/VSName/ResourcePath
```

Note:

Beginning with version 9.8, although CentraSite supports the existing REST Sample Requests and Responses, we enforce you use the REST Requests and REST Responses to specify additional details about the REST payload. You may use the Sample Requests and Responses if required.

Adding Virtual REST Service to Your Asset Catalog

To create and manage Virtual REST Service asset in CentraSite Business UI, you must have the following permissions:

- CentraSite Administrator
- Organization Administrator
- Asset Provider
- API Runtime Provider (required to configure run-time actions for the Virtual REST Services)
- Mediator Publisher (required to publish Virtual REST Services to Mediator gateways)
- API Portal Publisher (required to publish Virtual REST Services to API Portal gateways)
- Instance-level Modify permission for a gateway (required to publish Virtual REST Services to that particular gateway)

If you have the CentraSite Administrator role, you can create and manage Virtual REST Services within any organization.

If you have the Organization Administrator role or API Portal Administrator role for a specific organization, you have the ability to create and manage Virtual REST Services within that specific organization.

The following general guidelines apply when adding a Virtual REST Service in CentraSite Business UI:

- Ensure that the interface for the Native REST Service is completely implemented and that the interface is reflected in the RAML or Swagger file that is registered for the service in the CentraSite repository.
- An instance of the service is deployed and running at a known point in network.
- The metadata for the Native REST Service is valid and up-to-date. If the metadata for the Native REST Service has not been completely specified or is out-of-date, you should update it before you generate the Virtual REST Service so that you do not carry inaccurate/incomplete data into the Virtual REST Service.

In CentraSite Business UI, you can add a **Virtual REST Service** asset to the catalog in the following ways:

- You can *create a Virtual REST Service from an existing REST service, also called as a (Native) REST Service in CentraSite*, meaning that you create the virtual copy (proxy) of the existing REST Service using an already imported input file.
- You can *create a Virtual REST Service using an importer*, which is a utility that generates the Virtual REST Service from an imported archive file.
- You can *create a Virtual REST Service from scratch*, meaning that you create the Virtual REST Service (and set its attributes) manually.
- You can *create a Virtual REST Service using a command line tool*, which is a utility that generates the Virtual REST Service from an input RAML or Swagger file.

Adding Virtual REST Service using an Existing Native REST Service

You can create a virtual REST service asset from an existing native REST service asset. However, if you want to configure the run-time policies for the newly created virtual service asset using CentraSite, the `RuntimeComponentSetting` in the `centrasite.xml` file must be enabled. For information on enabling run-time aspects from CentraSite, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

When run-time aspects are enabled in CentraSite, then the creation of a virtual service from a native service would involve the following three steps. Else, only the first of the following three can be performed from CentraSite. In such scenario, the run-time policies can be configured from gateways to which the APIs are published.

Adding a Virtual REST Service asset from an existing Native REST Service asset, involves the following steps:

1. Creating a virtual copy (proxy) of the existing Native REST Service asset. For procedures, see [“Create Virtual REST Service” on page 993](#).
2. Configuring the run-time policy actions for the Virtual REST Service asset. For procedures, see [“Assign Policy Actions for Virtual REST Service” on page 995](#).
3. Implementing the virtualization of the existing Native REST Service asset, and publishing the Virtual REST Service asset to one or more gateways. For procedures, see [“Virtualize and Publish Virtual REST Service to Gateways” on page 995](#).

Create Virtual REST Service

You use panel 1 of the Virtualize Your API page to specify the proxy and invocation aliases, and endpoints for the new Virtual REST Service.

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, REST Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **Service**.
 - c. Click **OK**.
5. Click the REST Service you want to virtualize.

This opens the REST Service details page. Also, the actions bar displays a set of actions that are available for working with the REST Service.

6. On the actions bar of the REST Service details page, click **Virtualize**.
7. In the **Virtualize <Service_Name> (Step 1 of 3)** wizard, provide the required information for each of the displayed data fields.

Field	Description
Create a New Virtual Alias	<p>Name of the Virtual REST Service asset (also, termed as Virtual Alias).</p> <p>The name of a Virtual REST Service asset must be NCName-conformant, meaning that:</p> <ul style="list-style-type: none"> ■ The name must begin with a letter or the underscore character (<code>_</code>). ■ The remainder of the name can contain any combination of letters, digits, or the following characters: <code>.</code> <code>-</code> <code>_</code> (that is, period, dash, or underscore). It can also contain combining characters and extender characters (for example, diacriticals). ■ The name cannot contain any spaces. ■ Furthermore, if the Virtual REST Service name contains any non-conformant character, upon publishing the Virtual REST Service to any gateway, the non-conformant character is simply replaced with the underscore character (<code>_</code>) in Mediator. However, in CentraSite the Virtual REST Service name defined by you is displayed. <p>For more information about the NCName type, see http://www.w3.org/TR/xmlschema-2/#NCName</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: The name of a Virtual REST Service asset must be unique within an organization. If, for example, a Virtual REST Service with the same name already exists within the CentraSite registry, a warning message will be issued.</p> </div> <p>By default, CentraSite populates the Create a New Virtual Alias field with the display name that was specified for the Native REST Service.</p>
Endpoint prefix for invocation alias	<p>(Optional). Prefix for the Virtual REST Service.</p> <p>This field accepts URL paths.</p> <p>Example:</p> <ul style="list-style-type: none"> ■ <code>/testmethod/myprefix/</code> ■ <code>test@1234</code>
Endpoints of <Service_Name> to be virtualized	<p>The Native REST Service endpoint you want to use for invoking the Virtual REST Service.</p>

Field	Description
	The Endpoints list displays a list of the Endpoint URLs available for the Native REST Service.

8. Do one of the following:

- Click **Next** and proceed to panel 2 to proceed with configuring run-time policy actions. This button is enabled only if the run-time aspects from CentraSite is enabled. For procedure to enable CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).
- Click **Virtualize** to create virtual (proxy) copy of REST Service (without publishing the newly created Virtual Service to the selected gateways).

A Virtual REST Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the Virtual REST Service that you just created is displayed.

Note:

The **Virtualize** button is displayed in panel 1, only if the CentraSite run-time aspects are not enabled.

Assign Policy Actions for Virtual REST Service

You use panel 2 of the Virtualize Your API page to configure the policy actions for the Virtual REST Service.

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

1. Navigate to **(Step 2 of 3)** of the **Virtualize <Service_Name>** wizard.
2. Drag and drop the policy actions you want CentraSite to execute at runtime.
3. Specify the parameters for each policy action as described in [“Configure Policy Action Parameters” on page 1046](#), and then click **Next**.

Virtualize and Publish Virtual REST Service to Gateways

You use panel 3 of the Virtualize Your API page to configure the gateways for publishing the Virtual REST Service. The publishing operation allows the API Providers to expose the Virtual REST Service in the selected gateways. Clients can then access and examine the usage of the exposed Virtual REST Service.

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

1. Navigate to **(Step 3 of 3)** of the **Virtualize <Service_Name>** wizard.
2. In the **Gateway** list, select the gateways to publish the Virtual REST Service.

The available gateways are:
 - API Portal
 - Mediator
 - Insight Server
3. In the **Sandbox** list, select the category by which the gateways are classified.
4. Select the check box of a single gateway, or select the check boxes of multiple gateways to publish the Virtual REST Service.
5. (Applicable for an API Portal gateway only). Configure the sandbox categories for publishing the Virtual REST Service to the API Portal gateways. Follow these steps:
 - a. In the field labeled **Sandbox**, click the **Configure** icon.
 - b. In the **API Portal Settings** dialog box, select the sandbox category to which you want to publish the Virtual REST Service.
6. (Applicable for a Mediator or Insight Server gateway only). Select the **Expose to Consumers** option button to allow clients to simply access and examine the usage of the exposed Virtual REST Service.
7. Do one of the following:
 - Click **Virtualize** to create virtual (proxy) copy of REST Service (without publishing the newly created Virtual REST Service to the selected gateways).
 - Click **Publish** to create virtual (proxy) copy of REST Service, and simultaneously publish the newly created Virtual REST Service to the selected gateways.

A Virtual REST Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the Virtual REST Service that you just created is displayed.
8. Configure the extended attributes of the Virtual REST Service as described later in this topic.

Adding Virtual REST Service using an Archive

Pre-requisites:

To add a Virtual REST Service asset to an organization's asset catalog, you must belong to a role that has the Create Assets or Manage Assets permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

Before you import a Virtual REST Service asset to the catalog, you must have the archive file (.zip file). This file must reside on the file system of the computer where your browser is running.

You can import a Virtual REST Service using the archive file (.zip file) to which the Virtual REST Service was previously exported. You can import Virtual REST Services into the same CentraSite registry from which they were originally exported or to a different CentraSite registry.

➤ To add a Virtual REST Service asset using importer

1. In the CentraSite Business UI activity bar, click **Create Asset**.

This opens the **Create New Asset** wizard. The bottom panel of the **Create New Asset** wizard displays the option to import an archive file.

2. To import an archive file, click **Choose** to navigate to the folder where an exported archive file of the Virtual REST Service asset resides, and choose the file.

When you choose a file to import, the fields in the area labeled **Basic Information** cannot be edited.

3. Click **Next**.

The **Create New Asset** wizard displays a list of the referenced objects for the Virtual REST Service to import. The check box next to each object indicates whether the referenced object should be imported. By default, all objects displayed are included in the import set.

4. To exclude any referenced object from the import set, clear the corresponding check box.
5. Click the **Advanced Settings** chevron to expand the additional options that are available for the Virtual REST Service asset. Provide values for each of the displayed options:

Option	Description
Change Owner	The imported Virtual REST Service can be assigned to the same owner as in the source CentraSite registry, or you can assign a new owner. The Change Owner field is type-ahead field. As you type characters in this field, the dialog box lists the user names that match the characters you specify.

Option	Description
Change Organization	<p>When you import a Virtual REST Service, you can import it into the same organization in the target CentraSite registry as in the source CentraSite registry from which they were exported, or you can assign a new organization.</p> <p>The Change Organization field is type-ahead field. As you type characters in this field, the dialog box lists the organization names that match the characters you specify.</p>
Retain lifecycle state	This option determines whether the lifecycle state of the imported Virtual REST Service is preserved. Enable the option to retain the lifecycle state of the Virtual REST Service which is imported.
Overwrite existing entities	This option specifies that an existing Virtual REST Service with the same uuid in the target CentraSite registry will be overwritten, even if the Virtual REST Service in the archive is older than the one in the target CentraSite registry. Enable the option to overwrite the existing Virtual REST Service.
Import groups that the user belongs to	This option determines that when you import a user, whether you want to import the groups that the user belongs to. This applies only to user-defined groups. System-defined groups are not imported. Enable the option to import the groups.
Ignore API keys and OAuth2 tokens	This option determines whether the API keys and OAuth2 tokens of the imported assets are to be imported into the target CentraSite registry. Enable the option to ignore the API keys and OAuth2 tokens during the import process.

- Click **Import** to import the Virtual REST Service.

After the import operation completes, the import wizard informs you if the import was successful or if there were any errors and warnings.

- Click **Download Import Log** to view the import logs.

The import log lists the status of all the objects stating whether they were successfully imported or if there were errors and warnings.

- Click **OK** to terminate the import wizard.

A Virtual REST Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the Virtual REST Service asset that you just created is displayed.

- Configure the extended attributes of the Virtual REST Service asset as described later in this topic.

Tip:

If you had previously imported a RAML/Swagger file that has an associated schema file and you now re-import just the schema file with modifications, your browser might not display the updated contents of the schema file. This can happen if the browser cache is not being updated automatically. To rectify the problem, you can change your browser settings so that pages are always updated on every visit.

Adding Virtual REST Service from Scratch

Pre-requisites:

To add a Virtual REST Service asset to an organization's asset catalog, you must belong to a role that has the Create Assets or Manage Assets permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

Before you add a Virtual REST Service asset to the catalog, you must have the RAML or Swagger specification file that you want to import. This file can reside on the file system of the computer where your browser is running or it can reside anywhere on the network, as long as its location is addressable through a URL.

➤ To add a Virtual REST Service from scratch

1. In the CentraSite Business UI activity bar, click **Create Asset**.

This opens the **Create New Asset** wizard.

2. In the area labeled **Basic Information**, provide the required information for each of the displayed data fields:

In this field...	Do the following...
Name	<p>(Optional). Name of the Virtual REST Service asset.</p> <p>The name of a Virtual REST Service asset must be NCName-conformant, meaning that:</p> <ul style="list-style-type: none"> ■ The name must begin with a letter or the underscore character (_). ■ The remainder of the name can contain any combination of letters, digits, or the following characters: . - _ (that is, period, dash, or underscore). It can also contain combining characters and extender characters (for example, diacriticals). ■ The name cannot contain any spaces. ■ Furthermore, if the Virtual REST Service name contains any non-conformant character, upon publishing the Virtual REST Service to any gateway, the

In this field...**Do the following...**

non-conformant character is simply replaced with the underscore character (_) in Mediator. However, in CentraSite the Virtual REST Service name defined by you is displayed.

For more information about the NCName type, see <http://www.w3.org/TR/xmlschema-2/#NCName>

Note:

The name of a Virtual REST Service asset must be unique within an organization. If, for example, a Virtual REST Service with the same name already exists within the CentraSite registry, a warning message will be issued.

By default, CentraSite populates the **Create a New Virtual Alias** field with the display name that was specified for the Native REST Service.

Type

The asset type, **Virtual REST Service**.

Organization

The organization to which you want to add the Virtual REST Service. (The **Organization** list only displays organizations for which you have the Manage Assets permission or at least the Create Assets permission.)

Version

(Optional). The version identifier for the Virtual REST Service. You can enter any string in this field, that is, the version identifier does not need to be numeric. You can also leave the field blank. You can later create new versions of the Virtual REST Service. The default is 1.0.

Examples:

```
0.0a
1.0.0 (beta)
Pre-release 001
V1-2007.04.30
```

Description (Optional). The description for the Virtual REST Service.

Note:

This is the description information that users will see when they view this Virtual REST Service asset in the CentraSite user interfaces. Therefore, we recommend that you specify a meaningful description for each Virtual REST Service.

Import a File

The input RAML or Swagger file for the Virtual REST Service. You may want to read the input RAML or Swagger file from a URL-addressable location on the network (the **URL** option) or from your local file system (the **File** option).

Option**Description****URL**

If the RAML or Swagger file you are importing resides on the network, you can specify its URL.

In this field...	Do the following...
------------------	---------------------

File	If the specification resides in your local file system, specify the file name. You can use the Browse button to navigate to the required folder.
-------------	---

- Click the **Advanced Settings** chevron to expand the additional options that are available for the Virtual REST Service asset. Provide the required information for each of the displayed fields:

Field	Description
Credentials	If you have specified a URL and the site you want to access through the URL requires user authentication, type a username and password for authentication at the URL site.
Resolution	Select a resolution strategy, which will allow you to specify how an already existing imported and included file is handled. For each of the imported and included files you have one of these options: <ul style="list-style-type: none"> ■ Create new version: Creates a new version of the file with the new content (if, for example, you want to modify a RAML or Swagger file but want to retain its previous version). ■ Always overwrite: Overwrites the importing file with new content. <p>Note: Currently, CentraSite does not support this option for a REST Service with RAML or Swagger specification.</p>

- Click **Next**.

You can not navigate to the next screen unless all the required attributes have been set.

- In the **Preview** panel, review the basic information for the Virtual REST Service before you actually add to the CentraSite registry.

- Click **Save**.

A Virtual REST Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the Virtual REST Service asset that you just created is displayed.

- Configure the extended attributes of the Virtual REST Service asset as described later in this topic.

Viewing Virtual REST Service List

You use the Search Results page to display the list of Virtual REST Service assets.

> To view the list of Virtual REST Service assets

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, Virtual REST Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

A list of scopes that are available to you is displayed. By default, CentraSite contains the following predefined scopes:

Scope	Description
Everything	Displays a list of the currently available assets, organizations, and users in CentraSite registry.
Assets	Displays a list of the currently available assets in CentraSite registry. This is the default scope.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:

- a. Click the chevron next to **Assets** option button.
- b. In the displayed list of asset types, select **Virtual REST Service**.
- c. Click **OK**.

A list of defined Virtual REST Service assets is displayed in the Search Results page.

5. To filter the list to see just a subset of the available Virtual REST Service assets, type a partial string in the **Keyword** text box. Add the specified keyword to the Search Recipe, by clicking the plus symbol (+) next to the text box, or press Enter.

The Search Results page provides the following information about each Virtual REST Service asset:

Column	Description
Name	Name of the Virtual REST Service asset.
Description	The description for the Virtual REST Service.
Asset Type	The asset type, Virtual REST Service .
Last Updated Date	The date on which the Virtual REST Service was last modified.
Owner	The user who owns the Virtual REST Service.
Organization	The organization which owns the Virtual REST Service.
Version	The user-assigned version identifier for the Virtual REST Service.

To specify the sorting preference, select an attribute from the **Sort by** list.

Note:

Use the **View** menu to display the additional attributes.

Viewing and Modifying Virtual REST Service Details

You use the details page of a Virtual REST Service asset to examine and modify the RAML or Swagger specification.

The asset type **Virtual REST Service** has a unique set of profiles. However, an administrator can configure this asset type to display a customized set of profiles and attributes.

The following general guidelines apply when modifying the details of a Virtual REST Service in CentraSite Business UI:

- If you are not the owner of the Virtual REST Service asset, you cannot examine and modify the details of the Virtual REST Service, unless you have the View or Modify permission on the Virtual REST Service (granted through either a role-based permission or an instance-level permission).
- When you view the details page of a Virtual REST Service asset, you will only be able to modify the attributes of the profiles on which you have the Modify permission.
- When you hover over an attribute, CentraSite displays the tooltip text that provides a short description of the attribute's purpose.
- Some attributes accept only specific types of information. For example, if the Virtual REST Service asset type includes a URL type attribute, you must supply a URL when you modify that attribute. Other attribute types that require a specific type of value include the Date and Email attributes.
- Some attributes are designed to be read-only. You cannot modify the read-only attributes even if you have the CentraSite Administrator role.

- You can toggle the **Resources | Methods** menu to display the details of a Virtual REST Service in either the **Resources** view or the **Methods** view. The default display is Method-Centric view.
- When you are examining the Resources and Methods of a Virtual REST Service, you can choose to delete one or more of the top level REST components - REST Resources and the REST Methods.
- When you are modifying the Resources and Methods of a Virtual REST Service, you can choose to delete the other REST components - REST Parameters, HTTP Requests, HTTP Responses, and Sample Requests and Responses.
- When you view the Virtual REST Service details page in an Edit mode, you will only see an editable user interface of the Resource-Centric view. There is no Method-Centric view in the Edit mode.
- In addition to modifying the REST components of a Virtual REST Service, you can choose to delete one or more of the REST components - REST Resources, REST Methods, REST Parameters, HTTP Requests, HTTP Responses, and the Sample Requests and Responses.
- Currently, CentraSite supports only specific properties of the RAML and Swagger specifications. For example, if the Swagger specification includes a swagger version property, you will not be able to define the swagger's version in the Virtual REST Service details page.

Viewing and Modifying Basic Details of Virtual REST Service

In this task you examine and change the various basic and type-specific attributes associated with the Virtual REST Service. In addition, you can examine and change the various components of the Virtual REST Service - Resources, HTTP methods, Parameters, HTTP Requests and Responses, and the Sample Messages in the **Resources** view and the **Methods** view, as applicable. You can also delete the existing REST components - Resource Parameters, Method Parameters, HTTP Requests and Responses, and the Sample Messages.

➤ To examine and modify the basic details of a Virtual REST Service

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, Virtual REST Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the list of asset types, select **Virtual REST Service**.
 - c. Click **OK**.

A list of defined Virtual REST Service assets is displayed in the Search Results page.

5. Click the Virtual REST Service you want to examine and modify the basic attributes.

This opens the Virtual REST Service details page. Also, the actions bar displays a set of actions that are available for working with the Virtual REST Service.

You can hover over the **info** symbol next to an attribute to display the tooltip text, which describes the purpose of the attribute. The tooltip text displays the values of the attribute's Name, and Description fields that are contained in the Virtual REST Service type definition.

6. To modify the generic attributes that are displayed in the **Basic Information** profile, on the actions bar, click **Edit**. Change values of the attributes in the respective data fields as required.
7. Click **Save**.

The details page of a Virtual REST Service asset includes the following additional information:

Identification Profile (for Assets with Key-based Authentication)

Field	Description
API Key String	<i>Read-only. String.</i> The confidential secret key used to securely authenticate the consumer. The API Key String field is visible only to the consumer who requested an API key.
Expiry Date	<i>Read-only. String.</i> An expiration date for the API key.

Identification Profile (for Assets with OAuth-based Authentication)

Field	Description
Client Id	<i>Read-only. String.</i> The unique identifier that is used by the client to fetch access tokens for the virtual API.
Client Secret	<i>Read-only. String.</i> The secret key value that is used with the client identifier, serves as a password to fetch access tokens for the virtual API.
Client Name	<i>Read-only. String.</i> The name of the client (consumer application) that is attempting to get access to the virtual API.

Field	Description
Scope	<i>Read-only. String.</i> The scope value is the name of the virtual API. If the scope value is valid, Mediator obtains the access token. If no scope value is provided, Mediator provides the access token to the scope in which the client is allowed and adds the scope to the response.
Refresh Token	<i>Read-only. String.</i> The unique identifier used by the client to obtain a new access token when the current access token becomes invalid or expires.

API Key Scope Profile

Field	Description
API Service	<i>Read-only. String.</i> The name of the virtual API that is associated with the API key. To view details of the virtual API, click its hyperlinked name.

Viewing and Modifying Extended Details of Virtual REST Service

You use the **Resources and Methods** profile of a Virtual REST Service to view, modify, and delete the extended REST components - Resources and Methods. CentraSite Business UI displays a list of the currently defined Resources, or Methods based on the Resource-Centric view or Method-Centric view that you select.

If you have multiple Resources and Methods defined for a Virtual REST Service, you would modify the Resources and Methods using the corresponding **Edit** icons in the **Resources and Methods** profile.

➤ To examine and modify the details of resources and methods

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, Virtual REST Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:

- a. Click the chevron next to **Assets** option button.

- b. In the displayed list of asset types, select **Virtual REST Service**.
- c. Click **OK**.

A list of defined Virtual REST Service assets is displayed in the Search Results page.

5. Click the Virtual REST Service you want to examine and modify the extended attributes.

This opens the Virtual REST Service details page. Also, the actions bar displays a set of actions that are available for working with the Virtual REST Service.

You can hover over the **info** symbol next to an attribute to display the tooltip text, which describes the purpose of the attribute. The tooltip text displays the values of the attribute's Name, and Description fields that are contained in the Virtual REST Service type definition.

6. To modify the Virtual REST Service details that are displayed in the **Resource and Methods** profile, on the actions bar, click **Edit**.
7. Select the **Resource and Methods** profile. Add or modify the REST information at the Resource level and at the Method level, as required.

- Resource level details include the basic information for a REST Resource, and its Request Parameters.
- Method level details include the basic information for a REST Method, its Request parameters, Content Types, Status Codes, and HTTP Messages that are defined for the REST Method.

8. To modify the details of a REST Resource, follow these steps:

- a. In the list of REST Resources, hover over the resource you want to modify the attributes.
- b. Click the **Edit** icon.

This opens the **Edit Resource** dialog box

- c. In the **Edit Resource** dialog box, provide the required information for each of the displayed data fields:

Field	Description
Name	Name of the REST Resource. Make sure the name you specify in this field is a valid value for NCName.
Resource Path	The Resource URI.
Description	(Optional). The description for the REST Resource.
Upload Schema	(Optional). The XML Schema Definition (XSD) file for the REST Resource.

Field	Description
	<p>Note: If you have a Virtual REST Service that uses XML as content, then you can optionally upload an XML schema document.</p>
Upload Files	(Optional). Input files that provide additional information about the REST Resource.
Parameters	<p>(Optional). Request Parameters for the REST Resource.</p> <p>Modify an Existing Parameter</p> <ol style="list-style-type: none">In the list of Parameters, hover over the Parameter for which you want to modify the attributes. This displays icons for one or more actions that you can perform on the Parameter.Click the Edit icon. This opens the Edit Parameter dialog box.In the Edit Parameter dialog box, provide new values for the fields.Click OK.Repeat for each Parameter that you want to modify in the REST Resource. <p>Delete an Existing Parameter</p> <ol style="list-style-type: none">In the list of Parameters, hover over the Parameter you want to delete. This displays icons for one or more actions that you can perform on the Parameter.Click the Delete icon.Repeat for each Parameter that you want to delete from the REST Resource. <p>Add a New Parameter</p> <ol style="list-style-type: none">Click the Add Parameter link. This opens the Add Parameter dialog box.In the Add Parameter dialog box, provide values for the REST Parameter. To specify multiple parameters, click the Add Parameter link, and provide values for the new parameters.

Field	Description
	<ul style="list-style-type: none"> c. Click OK. d. Repeat for each Parameter that you want to add to the REST Resource.

9. To modify the details of a REST Method, follow these steps:

- a. In the list of Resources, hover over the Resource for which you want to modify the attributes.
- b. Click the **Edit** icon.

This opens the **Edit Resource** dialog box

- c. In the **Edit Method** dialog, provide the required information for each of the displayed data fields:

Field	Description
Name	Name of the REST (HTTP) Method.
Description	(Optional). The description for the REST Method.
HTTP Method	The HTTP operation you want to perform on the REST Resource.
Request Content-Type	The content format for HTTP Request message.
Response Content-Type	The content format for HTTP Response message.
Parameters	(Optional). Request Parameters for the REST Method.

Modify an Existing Parameter

- a. In the list of Parameters, hover over the Parameter for which you want to modify the attributes.

This displays icons for one or more actions that you can perform on the Parameter.

- b. Click the **Edit** icon.

This opens the **Edit Parameter** dialog box.

- c. In the **Edit Parameter** dialog box, provide new values for the fields.
- d. Click **OK**.
- e. Repeat for each Parameter that you want to modify in the REST Method.

Delete an Existing Parameter

Field	Description
	<ul style="list-style-type: none">a. In the list of Parameters, hover over the Parameter you want to delete. <p>This displays icons for one or more actions that you can perform on the Parameter.</p>b. Click the Delete icon.c. Repeat for each Parameter that you want to delete from the REST Method. <p>Add a New Parameter</p> <ul style="list-style-type: none">a. Click the Add Parameter link. <p>This opens the Add Parameter dialog box.</p>b. In the Add Parameter dialog box, provide values for the REST Parameter. <p>To specify multiple parameters, click the Add Parameter link, and provide values for the new parameters.</p>c. Click OK.d. Repeat for each Parameter that you want to add to the REST Method.
Requests	<p>(Optional). HTTP Requests indicating the operation that could be performed with the addressed REST Resource.</p> <p>Modify an Existing Request</p> <ul style="list-style-type: none">a. In the list of HTTP Requests, hover over the Request for which you want to modify the attributes. <p>This displays icons for one or more actions that you can perform on the Request.</p>b. Click the Edit icon. <p>This opens the Edit Request dialog box.</p>c. In the Edit Request dialog box, provide new values for the fields.d. Click OK.e. Repeat for each Request that you want to modify in the REST Method. <p>Delete an Existing Request</p> <ul style="list-style-type: none">a. In the list of HTTP Requests, hover over the Request you want to delete.

Field	Description
	<p>This displays icons for one or more actions that you can perform on the Request.</p> <ol style="list-style-type: none"> <li data-bbox="535 336 876 378">b. Click the Delete icon. <li data-bbox="535 399 1412 483">c. Repeat for each Request that you want to delete from the REST Method. <p>Add a New Request</p> <ol style="list-style-type: none"> <li data-bbox="535 546 958 588">a. Click the Add Request link. <p>This opens the Add Request dialog box.</p> <ol style="list-style-type: none"> <li data-bbox="535 672 1477 714">b. In the Add Request dialog box, provide values for the HTTP Request. <p>To specify multiple requests, click the Add Request link, and provide values for the new requests.</p> <ol style="list-style-type: none"> <li data-bbox="535 819 714 861">c. Click OK. <li data-bbox="535 882 1461 924">d. Repeat for each Request that you want to add to the REST Method.
Responses	<p>(Optional). HTTP Responses indicating the success or failure of a request invocation.</p> <p>Modify an Existing Response</p> <ol style="list-style-type: none"> <li data-bbox="535 1092 1445 1176">a. In the list of HTTP Responses, hover over the Response for which you want to modify the attributes. <p>This displays icons for one or more actions that you can perform on the Response.</p> <ol style="list-style-type: none"> <li data-bbox="535 1281 844 1323">b. Click the Edit icon. <p>This opens the Edit Response dialog box.</p> <ol style="list-style-type: none"> <li data-bbox="535 1407 1477 1449">c. In the Edit Response dialog box, provide new values for the fields. <li data-bbox="535 1470 714 1512">d. Click OK. <li data-bbox="535 1533 1412 1596">e. Repeat for each Response that you want to modify in the REST Method. <p>Delete an Existing Request</p> <ol style="list-style-type: none"> <li data-bbox="535 1680 1477 1764">a. In the list of HTTP Responses, hover over the Response you want to delete. <p>This displays icons for one or more actions that you can perform on the Response.</p>

Field	Description
	<ul style="list-style-type: none">b. Click the Delete icon.c. Repeat for each Response that you want to delete from the REST Method.
	<p>Add a New Response</p> <ul style="list-style-type: none">a. Click the Add Response link. This opens the Add Response dialog box.b. In the Add Response dialog box, provide values for the HTTP Response. To specify multiple requests, click the Add Response link, and provide values for the new requests.c. Click OK.d. Repeat for each Response that you want to add to the REST Method.
Sample Requests and Responses	<p>(Optional). Sample Requests to the Resources of the Virtual REST Service, and the corresponding Sample Responses from the Virtual REST Service.</p> <p>Modify an Existing Sample</p> <ul style="list-style-type: none">a. In the list of Sample Requests and Responses, hover over the Sample for which you want to modify the attributes. This displays icons for one or more actions that you can perform on the Sample.b. Click the Edit icon. This opens the Edit Sample Request and Response dialog box.c. In the Edit Sample Request and Response dialog box, provide new values for the fields.d. Click OK.e. Repeat for each Sample that you want to modify in the REST Method. <p>Delete an Existing Sample</p> <ul style="list-style-type: none">a. In the list of Sample Requests and Responses, hover over the Sample you want to delete. This displays icons for one or more actions that you can perform on the Sample.b. Click the Delete icon.

- | Field | Description |
|-------|--|
| | <p>c. Repeat for each Sample that you want to delete from the REST Method.</p> <p>Add a New Sample</p> <p>a. Click the Add Request and Response link.</p> <p style="padding-left: 20px;">This opens the Add Sample Request and Response dialog box.</p> <p>b. In the Add Sample Request and Response dialog box, provide values for the Sample.</p> <p style="padding-left: 20px;">To specify multiple samples, click the Add Response link, and provide values for the new samples.</p> <p>c. Click OK.</p> <p>d. Repeat for each Sample that you want to add to the REST Method.</p> |
| | <p>10. Click Save.</p> |

Deleting Virtual REST Services

If you are not the owner of a Virtual REST Service asset, you cannot delete the Virtual REST Service unless you have Full permission on the Virtual REST Service (granted through either a role-based permission or instance-level permission).

The following general guidelines apply when deleting a (Virtual) Service asset in CentraSite Control:

- A Virtual REST Service can only be deleted if it is not the target of an association from another registry object.
 - When you delete a Virtual REST Service, CentraSite removes the catalog entry for the Virtual REST Service (that is, it removes the instance of the Virtual REST Service from CentraSite's object database). Also note that:
 - The performance metrics and event information of the Virtual REST Service are also deleted.
- Note:**
When you delete the Virtual REST Service, this information is deleted by the built-in action **Delete RuntimeEvents and RuntimeMetrics of Service**, which is installed with CentraSite.
- When you delete a composite Virtual REST Service, all of its nonshared components are also deleted.
 - Deleting a Virtual REST Service will *not* remove:

- Other assets to which the Virtual REST Service refers (unless the reference is to an asset that is a nonshared component of the Virtual REST Service you are deleting). For example, if you are deleting a Virtual REST Service which has a Consumes or Consumed By relationship with other services in the registry, the related services will not be deleted.
- Supporting documents that are attached to the Virtual REST Service.
- Earlier versions of the Virtual REST Service. Only the latest version of the Virtual REST Service can be deleted; to remove earlier versions, they must be purged.
- You cannot delete a Virtual REST Service if:
 - The Virtual REST Service is in a pending state (for example, awaiting approval).
 - Any user in your CentraSite registry is currently modifying the Virtual REST Service.

➤ **To delete Virtual REST Service assets**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.This displays a list of assets in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Virtual REST Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **Virtual REST Service**.
 - c. Click **OK**.A list of defined Virtual REST Service assets is displayed in the Search Results page.
5. Select one or multiple Virtual REST Services you want to delete.
6. On the actions bar of the Search Results page, click **Delete**.

You can also delete a single Virtual REST Service assets from the actions bar of its details page.
7. When you are prompted to confirm the delete operation, click **Yes**.

Note:

If you have selected a set of Virtual REST Services, where one or multiple Virtual REST Services are in a pending state (for example, awaiting approval), CentraSite ignores the pending list of Virtual REST Services, and deletes any remaining Virtual REST Services for which you have the required permission.

Resource Synchronization in Virtual REST Services

REST Services are bound to change and evolve over time as they move through development, test and production. Modifications or enhancements such as, adding additional resources, HTTP methods or parameters, modifying the existing capabilities of resources, methods or parameters are performed on the particular REST Service. In the case of a REST Service that is virtualized to create a new Virtual REST Service, keep in mind that the newly created Virtual REST Service is an identical copy of the existing Native REST Service and not just a reference REST Service. This means that after a Virtual REST Service is created, any subsequent changes in the Native REST Service are not automatically propagated to the Virtual REST Service. This means that if you add Resource B to the Native REST Service, the newly added Resource B is not added to the already existing Virtual REST Service.

CentraSite provides the ability to deal with evolution of REST Services. Depending on the nature of change in the Native REST Service and your versioning strategy, CentraSite offers different mechanisms:

- **Versioning the REST Services:** CentraSite Business UI allows versioning of Native REST Services and Virtual REST Services. You can make the following kinds of versions on the REST Services:
 - Create a new version of the existing Native REST Service, make the necessary changes to the new version, and then create a Virtual REST Service from the new version. We recommend that you use this kind of versioning when there is a requirement for considerable change in the Native REST Service.
 - Create a new version of the existing Virtual REST Service, make the necessary changes to the new version, and then republish the Virtual REST Service to gateways.
- **Resynchronizing the REST Services:** CentraSite Business UI allows resynchronization of the Virtual REST Services. This means that you copy the resource definition of the Native REST Service (which includes HTTP methods, parameters, or sample requests and responses) to the Virtual REST Service without affecting other configurations such as, the run-time policy actions and the consumption settings.

CentraSite Business UI offers the possibility of synchronizing the REST resource metadata for a Virtual REST Service.

Note that there is no automatic synchronization of resource metadata between the REST Service and the Virtual REST Service. If you want to have the resource metadata in the Virtual REST Service to be the same as the resource metadata in the REST Service, you must manually reconfigure the resource metadata in the Virtual REST Service to synchronize with the resources in the REST Service.

The user must have API Runtime Provider role to manage synchronization for the specified Virtual REST Service.

When planning for the resource synchronization in Virtual REST Service, as a best practice, take the following points into consideration:

- If you want to simply reconfigure the run-time policy action configurations for a particular Virtual REST Service, we recommend that you use the **Virtualize** action in the details page of that Virtual REST Service.
- If you want to reconfigure the alias field labeled **Endpoint prefix for invocation alias** or synchronize resources for a particular Virtual REST Service, we then recommend that you use the **Virtualize** action in the details page of that Native REST Service.
- Be aware that only the additions and deletions at the resource-level are recognized by the **Virtualize** action during reconfiguration of a Virtual REST Service. Any changes that are made at the API-level and method-level are not recognized.

When you allow synchronization to happen on the Virtual REST Service, the updated metadata of the Native REST Service will completely overwrite the existing metadata of the Virtual REST Service.

For example, if you have a Native REST Service with two resources - Resource A and Resource B. Consider these two resources have the following methods:

Resource A	Resource B
GET Method	GET Method
POST Method	PUT Method

Assume you virtualize the Native REST Service to create a Virtual REST Service.

The Virtual REST Service will include the two resources - Resource A and Resource B with the above specified methods.

Consider you update the details page of the Native REST Service to include an additional metadata *DELETE Method* to the existing Resource B. The Native REST Service will now have the two resources with the following methods:

Resource A	Resource B
GET Method	GET Method
POST Method	PUT Method
	<i>POST Method</i>

Consider you update the details page of the Virtual REST Service to include an additional metadata *DELETE Method* to the Resource B. The Virtual REST Service will now have the two resources with the following methods:

Resource A	Resource B
GET Method	GET Method
POST Method	PUT Method
	<i>DELETE Method</i>

When you reconfigure the Virtual REST Service from the details page of the Native REST Service, in the **Recreate resources** section, CentraSite displays both the resources - Resource A and Resource B (with a comment **already exists**) with pre-selected check boxes by default.

Note that the Resource B has a different set of method specification for the Native and Virtual REST Services. However, on reconfiguring the Virtual REST Service, this difference of the method specification is not recognized in the user interface.

Now when you choose to reconfigure the Virtual REST Service with the default selection, upon synchronization, the existing metadata of the Virtual REST Service will be completely overwritten by the updated metadata of the Native REST Service. As a result, the Resource B of the Virtual REST Service will now contain the inherited **POST Method**; but the user-defined **DELETE Method** will no longer exist.

Important:

As a best practice, Software AG recommends that you maintain the Native REST Service as the single point of truth and synchronize all your changes with the Virtual REST Service using the synchronization process.

Resource Synchronization Usage Scenarios

There are significant use cases that require real-time and accurate data synchronization. Consider a REST Service allows users access to a collection of resources through a Virtual REST Service. The goal is to keep the data on the Virtual REST Service synchronized with the data of Native REST Service. Types of use cases that are contemplated within the scope of the synchronization are exemplified by, but not limited to, the following scenarios:

- **Add Resource Usage Scenario:** You choose to reconfigure an existing Virtual REST Service, and the Native REST Service indicates resources that have been introduced since the Virtual REST Service was initially created.
- **Edit Resource Usage Scenario:** You choose to reconfigure an existing Virtual REST Service, and the Native REST Service indicates resources that are also available in the Virtual REST Service.
- **Delete Resource Usage Scenario:** You choose to reconfigure an existing Virtual REST Service, and the Native REST Service indicates resources that have been deleted since the Virtual REST Service was initially created.
- **Combination Usage Scenario:** You choose to reconfigure an existing Virtual REST Service, and the Native REST Service indicates resource metadata that has undergone various changes (modifications, additions, deletions) since the Virtual REST Service was initially created.

The outcome of each operation (edit, add, and delete) on the resource metadata is given based on a very simple instance configuration in each of the usage scenarios.

Key	Description
Native REST Service	An instance of the type "REST Service"
Virtual REST Service	An instance of the type "Virtual REST Service"

Add Resource Usage Scenario

Native REST Service	Virtual REST Service
Resource A	Resource A (already exists)
Resource B	Resource B (already exists)
Resource C(newly added)	Resource C (newly added)

In this scenario, CentraSite displays the Resource C which has been newly added to the Native REST Service with a unselected check box.

- To allow synchronization of the newly added resource (that is, add the new Resource C to the Virtual REST Service), under the **Recreate resources** section, select the check box of the newly added Resource C, and click **Next**.
- To ignore synchronization, under the **Recreate resources** section, unselect all of the check boxes, and click **Next**.

Edit Resource Usage Scenario

Native REST Service	Virtual REST Service
Resource A	Resource A (already exists)
Resource B (updated)	Resource B (already exists)

In this scenario, CentraSite displays the Resource A and Resource B that are available in both the Native REST Service and the Virtual REST Service with pre-selected check boxes.

- To allow synchronization of the updated resource metadata (that is, include the modification to Resource B in the Virtual REST Service), under the **Recreate resources** section, retain the default check box selection, and click **Next**.
- To ignore synchronization, under the **Recreate resources** section, unselect all of the check boxes, and click **Next**.

Delete Resource Usage Scenario

Native REST Service

Resource A

Resource B (deleted)

Virtual REST Service

Resource A (already exists)

Resource B (differences)

In this scenario, CentraSite displays the Resource B which has been deleted from the Native REST Service with a unselected check box.

- To allow synchronization of the deleted resource metadata (that is, delete the Resource B from the Virtual REST Service), under the **Resource differences** section, select the check box of the Resource B you want to delete, and click **Next**.
- To ignore synchronization, keep all of the check boxes unselected in the **Recreate resources** and **Resource differences** sections, and click **Next**.

Combination of Usage Scenarios**Native REST Service**

Resource A (deleted)

Resource B (updated)

Resource C (added)

Virtual REST Service

Resource A (differences)

Resource B (already exists)

Resource C (newly added)

In this scenario, CentraSite displays the updated Resource B with a pre-selected check box, and the deleted Resource A and newly added Resource C with an unselected check box.

The resource synchronization mechanism can be best understood with the following table:

Use case	Recreate resources...			Resource differences...		
	Resource A	Resource B	Resource C	Resource A	Resource B	Resource C
Complete synchronization	N/A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	N/A	N/A
Synchronize modified resource only	N/A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	N/A	N/A
Synchronize added resource only	N/A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	N/A	N/A
Synchronize deleted resource only	N/A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	N/A	N/A
No synchronization	N/A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	N/A	N/A

Virtual OData Service Management

This section describes operations you can perform to manage virtual OData services through CentraSite Business UI.

Adding Virtual OData Service to Your Asset Catalog

To create and manage Virtual OData Service asset in CentraSite Business UI, you must have the following permissions:

- CentraSite Administrator
- Organization Administrator
- Asset Provider
- API Runtime Provider (required to configure run-time actions for the Virtual OData Services)
- Mediator Publisher (required to publish Virtual OData Services to Mediator gateways)
- API Portal Publisher (required to publish Virtual OData Services to API Portal gateways)
- Instance-level Modify permission for a gateway (required to publish Virtual OData Services to that particular gateway)

If you have the CentraSite Administrator role, you can create and manage Virtual OData Services within any organization.

If you have the Organization Administrator role or API Portal Administrator role for a specific organization, you have the ability to create and manage Virtual OData Services within that specific organization.

The following general guidelines apply when adding a Virtual OData Service asset in CentraSite Business UI:

- Ensure that the interface for the Native OData Service is completely implemented and that the interface is reflected in the EDMX file that is registered for the service in the CentraSite repository.
- An instance of the OData Service is deployed and running at a known point in network.
- The metadata for the Native OData Service is valid and up-to-date. If the metadata for the Native OData Service has not been completely specified or is out-of-date, you should update it before you generate the Virtual OData Service so that you do not carry inaccurate or incomplete data into the Virtual OData Service.

In CentraSite Business UI, you can add a **Virtual OData Service** asset to the catalog in the following ways:

- You can *create a Virtual OData Service from an existing OData service, also called as a (Native) OData Service in CentraSite*, meaning that you create the virtual copy (proxy) of the existing OData Service using an already imported input file.
- You can *create a Virtual OData Service using an importer*, which is a utility that generates the Virtual OData Service from an imported archive file.

- You can *create a Virtual OData Service using a command line tool*, which is a utility that generates the Virtual OData Service from an input EDMX file.

Adding Virtual OData Service using an Existing Native OData Service

You can create a virtual OData service asset from an existing native OData service asset. However, if you want to configure the run-time policies for the newly created virtual service asset using CentraSite, the `RuntimeComponentSetting` in the `centrasite.xml` file must be enabled. For information on enabling run-time aspects from CentraSite, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

When the run-time aspects configured from CentraSite is enabled, then the creation of a virtual OData service asset from a native OData service asset would involve the following three steps. Else, only the first of the following three can be performed from CentraSite. In such scenario, the run-time policies can be configured from gateways to which the APIs are published.

1. Creating a virtual copy (proxy) of the existing Native OData Service asset. For procedures, see [“Create Virtual OData Service” on page 1021](#).
2. Configuring the run-time policy actions for the Virtual OData Service asset. For procedures, see [“Assign Policy Actions for Virtual OData Service” on page 1023](#).
3. Implementing the virtualization of the existing Native OData Service asset, and publishing the Virtual OData Service asset to one or more gateways. For procedures, see [“Virtualize and Publish Virtual OData Service to Gateways” on page 1024](#).

Create Virtual OData Service

You use panel 1 of the Virtualize Your API page to specify the proxy and invocation aliases, and endpoints for the new Virtual OData Service.

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, OData Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **OData Service**.

- c. Click **OK**.
5. Click the OData Service you want to virtualize.

This opens the OData Service details page. Also, the actions bar displays a set of actions that are available for working with the OData Service.
 6. On the actions bar of the OData Service details page, click **Virtualize**.
 7. In the **Virtualize <Service_Name> (Step 1 of 3)** wizard, provide the required information for each of the displayed data fields.

Field	Description
Create a New Virtual Alias	<p>Name of the Virtual OData Service asset (also, termed as Virtual Alias).</p> <p>The name of a Virtual OData Service asset must be NCName-conformant, meaning that:</p> <ul style="list-style-type: none"> ■ The name must begin with a letter or the underscore character (_). ■ The remainder of the name can contain any combination of letters, digits, or the following characters: . - _ (that is, period, dash, or underscore). It can also contain combining characters and extender characters (for example, diacriticals). ■ The name cannot contain any spaces. ■ Furthermore, if the Virtual OData Service name contains any non-conformant character, upon publishing the Virtual OData Service to any gateway, the non-conformant character is simply replaced with the underscore character (_) in Mediator. However, in CentraSite the Virtual OData Service name defined by you is displayed.

For more information about the NCName type, see <http://www.w3.org/TR/xmlschema-2/#NCName>

Note:

The name of a Virtual OData Service asset must be unique within an organization. If, for example, a Virtual OData Service with the same name already exists within the CentraSite registry, a warning message will be issued.

By default, CentraSite populates the **Create a New Virtual Alias** field with the display name that was specified for the Native OData Service.

Field	Description
Endpoint prefix for invocation alias	(Optional). Prefix for the Virtual OData Service. This field accepts URL paths. Example: <ul style="list-style-type: none"> ■ /testmethod/myprefix/ ■ test@1234
Endpoints of <Service_name> to be virtualized	The Native OData Service endpoint you want to use for invoking the Virtual OData Service. The Endpoints list displays a list of the Endpoint URLs available for the Native OData Service.

8. In the **Resources of <Service_name> to Virtualize** field, select the resources you want to use for the Virtual OData Service, or select **All** to use all of the available resources.
9. Do one of the following:
 - Click **Next** and proceed to panel 2 to proceed with configuring run-time policy actions. This button is enabled only if the run-time aspects from CentraSite is enabled. For procedure to enable CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).
 - Click **Virtualize** to create virtual (proxy) copy of Web Service (without publishing the newly created Virtual Service to the selected gateways).

A Virtual OData Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the Virtual OData Service that you just created is displayed.

Note:

The **Virtualize** button is displayed in this panel only if the CentraSite run-time policies are not enabled.

Assign Policy Actions for Virtual OData Service

You use panel 2 of the Virtualize Your API page to configure the policy actions for the Virtual OData Service.

1. Navigate to **(Step 2 of 3)** of the **Virtualize <Service_Name>** wizard.
2. Drag and drop the policy actions you want CentraSite to execute at runtime.
3. Specify the parameters for each policy action as described in [“Configure Policy Action Parameters” on page 1046](#), and then click **Next**.

Virtualize and Publish Virtual OData Service to Gateways

You use panel 3 of the Virtualize Your API page to configure the gateways for publishing the Virtual OData Service. The publishing operation allows the API Providers to expose the Virtual OData Service in the selected gateways. Clients can then access and examine the usage of the exposed Virtual OData Service.

1. Navigate to **(Step 3 of 3)** of the **Virtualize <Service_Name>** wizard.
2. In the **Gateway** list, select the gateways to publish the Virtual OData Service.

The available gateways are:

- API Portal
- Mediator
- Insight Server

3. In the **Sandbox** list, select the category by which the gateways are classified.
4. Select the check box of a single gateway, or select the check boxes of multiple gateways to publish the Virtual OData Service.
5. (Applicable for an API Portal gateway only). Configure the sandbox categories for publishing the Virtual OData Service to API Portal gateways. Follow these steps:
 - a. In the field labeled **Sandbox**, click the **Configure** icon.
 - b. In the **API Portal Settings** dialog box, select the sandbox category to publish the Virtual OData Service.
6. (Applicable for a Mediator or Insight Server gateway only). Select the **Expose to Consumers** option button to allow clients to simply access and examine the usage of the exposed Virtual OData Service.
7. Do one of the following:
 - Click **Virtualize** to create virtual (proxy) copy of Web Service (without publishing the newly created Virtual Service to the selected gateways).
 - Click **Publish** to create virtual (proxy) copy of Web Service, and simultaneously publish the newly created Virtual Service to the selected gateways.

A Virtual OData Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the Virtual OData Service that you just created is displayed.

8. Configure the extended attributes of the Virtual OData Service asset as described later in this topic.

Adding Virtual OData Service using an Archive

Pre-requisites:

To add a Virtual OData Service asset to an organization's asset catalog, you must belong to a role that has the Create Assets or Manage Assets permission for that organization.

Note:

By default, users with the CentraSite Administrator or Asset Administrator role have this permission.

Before you import a Virtual OData Service asset to the catalog, you must have the archive file (.zip file). This file must reside on the file system of the computer where your browser is running.

You can import a Virtual OData Service using the archive file (.zip file) to which the OData service was previously exported. You can import OData services into the same CentraSite registry from which they were originally exported or to a different CentraSite registry.

➤ To add a Virtual OData Service asset using importer

1. In the CentraSite Business UI activity bar, click **Create Asset**.

This opens the **Create New Asset** wizard. The bottom panel of the **Create New Asset** wizard displays the option to import an archive file.

2. To import an archive file, click **Choose** to navigate to the folder where the exported archive file of a Virtual OData Service asset resides, and choose the file.

When you choose a file to import, the fields in the area labeled **Basic Information** cannot be modified.

3. Click **Next**.

The **Create New Asset** wizard displays a list of the referenced objects for the Service to import. The check box next to each referenced object indicates whether the object should be imported. By default, all objects displayed are included in the import set.

4. To exclude any referenced object from the import set, clear the corresponding check box.
5. Click the **Advanced Settings** chevron to expand the additional options that are available for the OData Service asset. Provide values for each of the displayed options:

Option	Description
Change Owner	<p>The imported OData Service can be assigned to the same owner as in the source CentraSite registry, or you can assign a new owner.</p> <p>The Change Owner field is type-ahead field. As you type characters in this field, the dialog box lists the user names that match the characters you specify.</p>
Change Organization	<p>When you import a Virtual OData Service, you can import it into the same organization in the target CentraSite registry as in the source CentraSite registry from which they were exported, or you can assign a new organization.</p> <p>The Change Organization field is type-ahead field. As you type characters in this field, the dialog box lists the organization names that match the characters you specify.</p>
Retain lifecycle state	<p>This option determines whether the lifecycle state of the imported OData Service is preserved. Enable the option to retain the lifecycle state of the OData Service which is imported.</p>
Overwrite existing entities	<p>This option specifies that an existing OData Service with the same uuid in the target CentraSite registry will be overwritten, even if the OData Service in the archive is older than the one in the target CentraSite registry. Enable the option to overwrite the existing OData Service.</p>
Import groups that the user belongs to	<p>This option determines that when you import a user, whether you want to import the groups that the user belongs to. This applies only to user-defined groups. System-defined groups are not imported. Enable the option to import the groups.</p>
Ignore API keys and OAuth2 tokens	<p>This option determines whether the API keys and OAuth2 tokens of the imported assets are to be imported into the target CentraSite registry. Enable the option to ignore the API keys and OAuth2 tokens during the import process.</p>

- Click **Import** to import the OData Service.

After the import operation completes, the import wizard informs you if the import was successful or if there were any errors and warnings.

- Click **Download Import Log** to view the import logs.

The import log lists the status of all the objects stating whether they were successfully imported or if there were errors and warnings.

- Click **OK** to terminate the import wizard.

An OData Service asset instance is created in the specified organization and registered with the CentraSite registry/repository. The details page for the OData Service asset that you just created is displayed.

Tip:

If you had previously imported an EDMX file that has an associated schema file and you now re-import just the schema file with modifications, your browser might not display the updated contents of the schema file. This can happen if the browser cache is not being updated automatically. To rectify the problem, you can change your browser settings so that pages are always updated on every visit.

Viewing Virtual OData Service List

You use the Search Results page to display the list of Virtual OData Service assets.

> To view the list of Virtual OData Service assets

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, Virtual OData Service, click **Choose**.

This opens the **Choose Asset Types** dialog box.

A list of scopes that are available to you is displayed. By default, CentraSite contains the following predefined scopes:

Scope	Description
Everything	Displays a list of the currently available assets, organizations, and users in CentraSite registry.
Assets	Displays a list of the currently available assets in CentraSite registry. This is the default scope.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:

- a. Click the chevron next to **Assets** option button.
- b. In the displayed list of asset types, select **Virtual OData Service**.
- c. Click **OK**.

A list of defined Virtual OData Service assets is displayed in the Search Results page.

5. To filter the list to see just a subset of the available Virtual OData Service assets, type a partial string in the **Keyword** text box. Add the specified keyword to the Search Recipe, by clicking the plus symbol (+) next to the text box, or press Enter.

The Search Results page provides the following information about each Virtual OData Service asset:

Column	Description
Name	Name of the Virtual OData Service asset.
Description	The description for the Virtual OData Service.
Asset Type	The asset type, Virtual OData Service .
Last Updated Date	The date on which the Virtual OData Service was last modified.
Owner	The user who owns the Virtual OData Service.
Organization	The organization which owns the Virtual OData Service.
Version	The user-assigned version identifier for the Virtual OData Service.

To specify the sorting preference, select an attribute from the **Sort by** list.

Note:

Use the **View** menu to display the additional attributes.

Viewing and Modifying Virtual OData Service Details

You use the details page of a Virtual OData Service asset to examine and modify the EDMX specification.

The asset type **Virtual OData Service** has a unique set of profiles. However, an administrator can configure this asset type to display a customized set of profiles and attributes.

The following general guidelines apply when modifying the details of a Virtual OData Service asset in CentraSite Business UI:

- If you are not the owner of the Virtual OData Service, you cannot examine or modify the details of the Virtual OData Service, unless you have the View or Modify permission on the Virtual Service (granted through either a role-based permission or an instance-level permission).
- When you view the details page of a Virtual OData Service, you will only see profiles for which you have the profile-level View permission. You will only be able to modify the attributes of the profiles on which you have the Modify permission.
- When you hover over an attribute, CentraSite displays the tooltip text that provides a short description of the attribute's purpose.
- Some attributes accept only specific types of information. For example, if the Virtual OData Service asset type includes a URL type attribute, you must supply a URL when you modify

that attribute. Other attribute types that require a specific type of value include the Date and Email attributes.

- Some attributes are designed to be read-only. You cannot modify the read-only attributes even if you have the CentraSite Administrator role.

In this task you modify the basic and type-specific attributes that are associated with a Virtual OData Service.

➤ **To examine and modify the details of a Virtual OData Service asset**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.This displays a list of assets in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Virtual OData Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **Virtual OData Service**.
 - c. Click **OK**.
5. Click the Virtual OData Service you want to examine and modify the attributes.

This opens the Virtual OData Service details page. Also, the actions bar displays a set of actions that are available for working with the Virtual OData Service.

You can hover over the **info** symbol next to an attribute to display the tooltip text, which describes the purpose of the attribute. The tooltip text displays the values of the attribute's Name, and Description fields that are contained in the Virtual OData Service type definition.

6. To modify the generic attributes that are displayed in the **Basic Information** profile, on the actions bar, click **Edit**. Change values of the attributes in the respective data fields as required.
7. To modify the extended attributes that are displayed in the individual profiles, follow these steps:
 - a. Select the profile that contains the attribute(s) you want to modify.

- b. Change values of the attributes in the respective data fields as necessary.
 - c. Repeat steps 7.a and 7.b for each profile for which you want to modify the attributes.
8. Click **Save**.

The details page of a Virtual OData Service asset includes the following additional information:

Identification Profile (for Assets with Key-based Authentication)

Field	Description
API Key String	<i>Read-only. String.</i> The confidential secret key used to securely authenticate the consumer. The API Key String field is visible only to the consumer who requested an API key.
Expiry Date	<i>Read-only. String.</i> An expiration date for the API key.

Identification Profile (for Assets with OAuth-based Authentication)

Field	Description
Client Id	<i>Read-only. String.</i> The unique identifier that is used by the client to fetch access tokens for the virtual API.
Client Secret	<i>Read-only. String.</i> The secret key value that is used with the client identifier, serves as a password to fetch access tokens for the virtual API.
Client Name	<i>Read-only. String.</i> The name of the client (consumer application) that is attempting to get access to the virtual API.
Scope	<i>Read-only. String.</i> The scope value is the name of the virtual API. If the scope value is valid, Mediator obtains the access token. If no scope value is provided, Mediator provides the access token to the scope in which the client is allowed and adds the scope to the response.
Refresh Token	<i>Read-only. String.</i> The unique identifier used by the client to obtain a new access token when the current access token becomes invalid or expires.

API Key Scope Profile

Field	Description
API Service	<i>Read-only. String.</i> The name of the virtual API that is associated with the API key. To view details of the virtual API, click its hyperlinked name.

Deleting Virtual OData Services

If you are not the owner of a Virtual OData Service asset, you cannot delete the Virtual OData Service unless you have Full permission on the Virtual OData Service (granted through either a role-based permission or instance-level permission).

The following general guidelines apply when deleting a (Virtual) Service asset in CentraSite Control:

- A Virtual OData Service can only be deleted if it is not the target of an association from another registry object.
- When you delete a Virtual OData Service, CentraSite removes the catalog entry for the Virtual OData Service (that is, it removes the instance of the Virtual OData Service from CentraSite's object database). Also note that:
 - The performance metrics and event information of the Virtual OData Service are also deleted.

Note:

When you delete the Virtual OData Service, this information is deleted by the built-in action **Delete RuntimeEvents and RuntimeMetrics of Service**, which is installed with CentraSite.

- When you delete a composite Virtual OData Service, all of its nonshared components are also deleted.
- Deleting a Virtual OData Service will *not* remove:
 - Other assets to which the Virtual OData Service refers (unless the reference is to an asset that is a nonshared component of the Virtual OData Service you are deleting). For example, if you are deleting a Virtual OData Service which has a Consumes or Consumed By relationship with other services in the registry, the related services will not be deleted.
 - Supporting documents that are attached to the Virtual OData Service.
 - Earlier versions of the Virtual OData Service. Only the latest version of the Virtual OData Service can be deleted; to remove earlier versions, they must be purged.
- You cannot delete a Virtual OData Service if:
 - The Virtual OData Service is in a pending state (for example, awaiting approval).
 - Any user in your CentraSite registry is currently modifying the Virtual OData Service.

➤ To delete Virtual OData Service assets

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Virtual OData Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **Virtual OData Service**.
 - c. Click **OK**.

A list of defined Virtual OData Service assets is displayed in the Search Results page.

5. Select one or multiple Virtual OData Services you want to delete.
6. On the actions bar of the Search Results page, click **Delete**.

You can also delete a single Virtual OData Service assets from the actions bar of its details page.

7. When you are prompted to confirm the delete operation, click **Yes**.

Note:

If you have selected a set of Virtual OData Services, where one or multiple Virtual OData Services are in a pending state (for example, awaiting approval), CentraSite ignores the pending list of Virtual OData Services, and deletes any remaining Virtual OData Services for which you have the required permission.

General Procedures across Assets

This section outlines the general procedures across assets performed through CentraSite Business UI.

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

Configuring API Consumption Settings for Client Authentication

APIs in CentraSite require an unique identifier for consumption. A unique identifier can do the following:

- **Identify** the client requesting for an API consumption.
- **Authenticate and validate** the client for accessing and consuming the API.
- **Authorize** if the client making the consumption request is allowed to access and consume the API.

CentraSite Business UI supports the following mechanisms for client authentication:

- API Key
- OAuth 2.0 token

An API Provider or a user with appropriate permission defines and enforces the type of client authentication that is required for consuming the particular API. Based on the client authentication that is enforced for the API, a client will fetch the API key or OAuth 2.0 token and consume (call) the API.

You configure the client authentication mechanisms using the **API Consumption Settings** action in the details page of Virtual Service (also, called Virtual API) asset.

To configure the client authentication mechanisms for a Virtual Service asset, you must have the following roles or permissions:

- CentraSite Administrator
- Organization Administrator
- Asset Provider
- Full instance-level permission on the Virtual Service asset

Note:

Until you have the instance-level Modify permission on an API asset (at a minimum), you will not be able to see the **API Consumption Settings** action in the API details page.

Configuring Key-Based Authentication

You configure the following information to enable API key authentication:

- Specify the approval requirements for clients requesting API keys.
You can specify that requests must be approved by approver groups of your choosing, or you can specify that requests will be automatically approved.
- Configure email messages to be sent to:
 - The approver groups when requests are submitted for approval.
 - The clients to inform them of their approval status.
- Specify the expiration of the API key.

To use an API key to access and consume the API in CentraSite, clients must:

1. Register as a consumer for the API.

When the client registration request is approved, the client receives an API key (a base64-encoded string of the `consumer-key:consumer-secret` combination). It works for both SOAP and REST calls.

2. To call the API, the client must pass the API key in an HTTP request header or as a query string parameter. The use of this key establishes the client's identity and authentication.

➤ **To configure API for Key-based client authentication**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Virtual Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **Virtual Service**.
 - c. Click **OK**.

A list of defined Virtual Service assets is displayed in the Search Results page.

5. Click the asset you want to configure the API key authentication.

This opens the Virtual Service details page. Also, the actions bar displays a set of actions that are available for working with the Virtual Service.

6. On the actions bar of the Virtual Service details page, click **API Consumption Settings**.

This opens the **API Consumption Settings** dialog box.

7. Select **API Keys**.

You might wish to modify the authentication type for an API at a later stage. When modifying the authentication type (say, **OAuth2**), if the API has one or more API keys generated using the currently configured authentication mechanism, CentraSite issues a warning message.

The message states that the modification to the configured authentication mechanism would deactivate all of the existing API keys that were generated using the authentication mechanism that you now intend to modify.

8. In the **Usage Contract Expires After** field, specify the maximum time that an API key will be valid for use with the API. The key expires after the set number of seconds, minutes, hours, days, weeks, months, or years.

The time is specified in the following form: *#y #m #w #d #h #min #s*, where:

- *#* is a real number
- *y* indicates the year
- *m* indicates the month
- *w* indicates the week
- *d* indicates the day
- *h* indicates the hour
- *min* indicates the minute
- *s* indicates the second

An example of a time: 1y 2m 3w 7d 5h 30min 15s. This value indicates that the API key will remain active until 1 year, 2 months, 3 weeks, 7 days, 5 hours, 30 minutes, and 15 seconds.

The default value of **Unlimited** denotes that the API key never expires.

When an API key expires, you can renew the expired API key in the following ways:

- You can re-submit an API key request for consumption.
 - You (as an API provider) can renew the API key using the **Renew** action.
9. Select the **Require Approval** checkbox if you want to initiate an approval workflow for generating and renewing the API key.

When a client requests for generating or renewing an API key that triggers an approval, CentraSite initiates an approval workflow and submits the client's request to the designated group of approvers.

Approvers receive the approval request in the **Pending Approval Requests** in the API details page. Approvers whose user account includes a valid email address also receive an email message informing them that a request is awaiting their approval.

CentraSite does not execute the client's requested operation until it obtains the necessary approvals. If an approver rejects the request, CentraSite notifies the requestor.

- If you do not select the **Require Approval** checkbox, the request is automatically approved and CentraSite executes the client's registration request.

10. If you select the **Require Approval** checkbox, complete the following fields:

Field	Description	
Approval is needed from	All	Requests must be approved by all users specified in Approver Group . (It does not matter in which order the approvals are issued.) A single rejection will cause the request to be rejected.
	Any	<i>Default.</i> Requests can be approved or rejected by any single user in Approver Group . Only one user from the set of authorized approvers is required to approve or reject the request.
Approver Group	Specify the approver group. You can specify multiple approver groups.	

11. In the **Key Generation Settings** section, complete the following fields so that CentraSite will send emails consumers initially request API keys.

CentraSite automatically populates the default email settings (Subject, Template, Action) with the <API Key Settings> information from the `centrasite.xml` file.

Field	Description
Subject	The text that will appear on the subject line of the email.
Template	The template that will be used to generate the body of the email message.
	To specify an additional template, use the plus button to add additional rows.
<p>Important: CentraSite sends notifications about a request status to the consumer requesting for an API key; only if the client has enabled the Email notifications option in his User Preferences page.</p>	
Action	Specify the approval action.
<u>Value</u>	<u>Description</u>
Approved	(Default). CentraSite sends an email message to the client when requests are approved. If you choose this option, you can use the predefined template <code>APIKeyGenerationSuccess.html</code> for approval notifications if you do not want to create an email template of your own.

Field	Description
Approval Request	<p>CentraSite sends an email message to the approver(s) when requests are submitted for approval.</p> <p>If you choose this option, you can use the predefined template <code>PendingApprovalNotification.html</code> for pending-approval notifications if you do not want to create an email template of your own.</p>
Rejected	<p>CentraSite sends an email message to the client when requests are rejected.</p> <p>If you choose this option, you can use the predefined template <code>RejectionNotification.html</code> for rejection notifications if you do not want to create an email template of your own.</p>

12. In the **Key Renewal Settings** section, complete the following fields so that CentraSite will send emails when consumers request API key renewals.

CentraSite automatically populates the default email settings (Subject, Template, Action) with the <API Key Settings> information fetched from the `centrasite.xml` file.

Field	Description				
Subject	The text that will appear on the subject line of the email.				
Template	<p>The template that will be used to generate the body of the email message.</p> <p>To specify an additional template, use the plus button to add additional rows.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Important: CentraSite sends notifications to the client only if the client has enabled the Email notifications option in his User Preferences page.</p> </div>				
Action	Specify the approval action.				
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Approved</td> <td> <p>(Default). CentraSite sends an email message to the client when requests are approved.</p> <p>If you choose this option, you can use the predefined template <code>APIKeyRenewalSuccess.html</code> for approval notifications if you do not want to create an email template of your own.</p> </td> </tr> </tbody> </table>	Value	Description	Approved	<p>(Default). CentraSite sends an email message to the client when requests are approved.</p> <p>If you choose this option, you can use the predefined template <code>APIKeyRenewalSuccess.html</code> for approval notifications if you do not want to create an email template of your own.</p>
Value	Description				
Approved	<p>(Default). CentraSite sends an email message to the client when requests are approved.</p> <p>If you choose this option, you can use the predefined template <code>APIKeyRenewalSuccess.html</code> for approval notifications if you do not want to create an email template of your own.</p>				

Field	Description
Approval Request	<p>CentraSite sends an email message to the approver group(s) when requests are submitted for approval.</p> <p>If you choose this option, you can use the predefined template <code>APIKeyRenewalPendingNotification.html</code> for pending-approval notifications if you do not want to create an email template of your own.</p>
Rejected	<p>CentraSite sends an email message to the client when requests are rejected.</p> <p>If you choose this option, you can use the predefined template <code>RejectionNotification.html</code> for rejection notifications if you do not want to create an email template of your own.</p>

13. In the **Key Revocation Settings** section, complete the following fields so that CentraSite will send emails when consumers request to have API keys revoked.

CentraSite automatically populates the default email settings (Subject, Template, Action) with the `<API Key Settings>` information fetched from the `centrasite.xml` file.

Field	Description
Subject	The text that will appear on the subject line of the email.
Template	<p>The template that will be used to generate the body of the email message to the client.</p> <p>If you choose this option, you can use the predefined template <code>APIKeyRevocationSuccess.html</code> for success notifications if you do not want to create an email template of your own.</p> <p>Important: CentraSite sends notifications to the client only if the consumer has enabled the Email notifications option in his User Preferences page.</p>

14. Click the **Configure** button.

CentraSite internally creates and activates an API Key Generation Policy specific to the API. When a client registers as a consumer, this policy will start the process of approving and generating the API key.

Configuring OAuth-Based Authentication

The type of OAuth 2.0 authorization grant that CentraSite supports is *Client Credentials*. Client credentials are used as an authorization grant when the client is requesting API to protected resources based on an authorization previously arranged with the authorization server. That is,

the client application gains authorization when it successfully registers with CentraSite as a consumer.

You configure the following information to enable OAuth 2.0 authentication:

- Specify the approval requirements for client requests for client credentials.

You can specify that requests must be approved by approver groups of your choosing, or you can specify that requests will be automatically approved.

- Configure email messages to be sent to:
 - The approver groups when requests are submitted for approval.
 - The clients to inform them of their approval status.

To use OAuth 2.0 client credentials to access and consume the API in CentraSite, clients must:

1. Register as a consumer for the API.

When the client registration request is approved, the client receives client credentials (a `client_id` and `client_secret`).

2. Request an OAuth 2.0 access token by passing the client credentials to the Mediator-hosted REST service `mediator.oauth2.getOAuth2AccessToken`. This service will provide an OAuth 2.0 access token to the client.

3. To call the API, the client must pass their OAuth access token in an HTTP request header.

An OAuth 2.0 token is a unique token that a client uses to invoke APIs using the OAuth 2.0 protocol. The token contains an identifier that uniquely identifies the client. The use of a token establishes the client's identity and is used for both the authentication and authorization.

➤ To configure API for OAuth-based client authentication

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Virtual Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.

- b. In the displayed list of asset types, select **Virtual Service**.
- c. Click **OK**.

A list of defined Virtual Service assets is displayed in the Search Results page.

5. Click the asset you want to configure the API key authentication.

This opens the Virtual Service details page. Also, the actions bar displays a set of actions that are available for working with the Virtual Service.

6. On the actions bar of the Virtual Service details page, click **API Consumption Settings**.

This opens the **API Consumption Settings** dialog box.

7. Select **OAuth2**.

You might wish to modify the authentication type for an API at a later stage. When modifying the authentication type (say, **API Keys**), if the API has one or more OAuth 2.0 tokens generated using the currently configured authentication mechanism, CentraSite issues a warning message.

The message states that the modification to the configured authentication mechanism would deactivate all of the existing OAuth 2.0 tokens that were generated using the authentication mechanism that you now intend to modify.

8. In the **Refresh Token After** field, specify the maximum time that an OAuth 2.0 token will be valid for use with the API. The token automatically refreshes after the set number of seconds, minutes, hours, days, weeks, months, or years.

The time is specified in the following form: #y #m #w #d #h #min #s, where:

- # is a real number
- *y* indicates the year
- *m* indicates the month
- *w* indicates the week
- *d* indicates the day
- *h* indicates the hour
- *min* indicates the minute
- *s* indicates the second

An example of a time: 1y 2m 3w 7d 5h 30min 15s. This value indicates that the OAuth 2.0 token will expire after 1 year, 2 months, 3 weeks, 7 days, 5 hours, 30 minutes, and 15 seconds.

The default value of **Unlimited** denotes that the OAuth 2.0 token never expires.

9. Select the **Require Approval** checkbox if you want to initiate an approval workflow for generating the client credentials.

When a client request triggers an approval, CentraSite initiates an approval workflow and submits the client's request to the designated group of approvers. Approvers receive the approval request in the **Pending Approval Requests** in the API details page. Approvers whose user account includes a valid email address also receive an email message informing them that a request is awaiting their approval.

CentraSite does not execute the client's requested operation until it obtains the necessary approvals. If an approver rejects the request, CentraSite notifies the requestor.

- If you do not select the **Require Approval** checkbox, the request is automatically approved, and CentraSite executes the client's registration request.

10. If you select the **Require Approval** checkbox, complete the following fields:

Field	Description
Approval is needed from	All Requests must be approved by all users specified in Approver Group . (It does not matter in which order the approvals are issued.) A single rejection will cause the request to be rejected.
	Any (Default). Requests can be approved or rejected by any single user in Approver Group . Only one user from the set of authorized approvers is required to approve or reject the request.
Approver Group	Specify the approver group. You can specify multiple approver groups.

11. In the **Key Generation Settings** section, complete the following fields so that CentraSite will send emails when a client requests a token.

CentraSite automatically populates the default email settings (Subject, Template, Action) with the <API Key Settings> information from the `centrasite.xml` file.

Field	Description
Subject	The text that will appear on the subject line of the email.
Template	The template that will be used to generate the body of the email message. To specify another template, use the plus button to add additional rows.

Important:

CentraSite sends notifications about a request status to the client only if the client has enabled the **Email** notifications option in his User Preferences page.

Field	Description	
Action	Specify the approval action.	
	Value	Description
	Approved	(Default). CentraSite sends an email message to clients when requests are approved. If you choose this option, you can use the predefined template <code>APIKeyGenerationSuccess.html</code> for approval notifications if you do not want to create an email template of your own.
	Approval Request	CentraSite sends an email message to the approver group(s) when requests are submitted for approval. If you choose this option, you can use the predefined template <code>PendingApprovalNotification.html</code> for pending-approval notifications if you do not want to create an email template of your own.
	Rejected	CentraSite sends an email message to clients when requests are rejected. If you choose this option, you can use the predefined template <code>RejectionNotification.html</code> for rejection notifications if you do not want to create an email template of your own.

12. Click the **Configure** button.

When a client registers as a consumer, an approval request is sent to the approvers you specified above.

Assigning Run-Time Actions to a Virtual Service

This section describes how to use the **Virtualize** action to configure the policy actions for the Virtual Service.

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

To access the **Virtualize** action, you must have the following roles or permissions:

- CentraSite Administrator
- Organization Administrator

- Asset Provider
- Instance-level Modify permission on the Native Service or Virtual Service

In addition to the above described roles and permissions for accessing the **Virtualize** action in the details page of a particular Virtual Service, make sure that you have the specific API Runtime Provider role for reconfiguring the run-time actions of the Virtual Services.

Before You Begin

The **Virtualize <Service_Name> (Step 2 of 3)** wizard specifies the list of actions to govern the run-time behavior for the Virtual Service.

When you define the actions for a Virtual Service, keep the following points in mind:

- A run-time policy configuration is valid if:
 - it consists of at least one action in each of these stages - **Receive** and **Routing**.
 - there is a valid endpoint configured in the **Route to** field of the Routing action.
- When you drag an action from the **Policy Actions** area, the respective step in the **Message Flow** area highlights where the action fits in, thus making the navigation from **Policy Actions** area to the **Message Flow** area more intuitive.
- Not all stages support the full set of actions. Every action happens only within a respective step. For example, the “Evaluate” actions occur only on the **Enforce** stage; while the “Routing” actions occur only on the **Routing** stage.
- Mediator executes the policy actions configured for the Virtual Service in a predefined order.
- Some actions are mutually dependent. That is, a specific action must be used in conjunction with another particular action. For example, a **Message Flow** area that includes the Set JMS Headers action must also include the JMS Routing Rule action.
- Some actions are mutually exclusive. That is, a specific action cannot be used in conjunction with another particular action. For example, a **Message Flow** area that includes the JMS Routing Rule action cannot include the Straight Through Routing action.
- Some of the actions are allowed to appear multiple times within a message flow step.

For those actions that can appear in a message flow only once (for example, Evaluate IP Address), Mediator will choose only one, which might cause problems or unintended results.
- You can view a tooltip text for any accordion by moving the cursor over the accordion name. The tooltip text gives a summary of the accordion’s purpose.
- If you modify the policy action for a Virtual Service which is already published to a Mediator gateway, CentraSite automatically republishes the modified Virtual Service.
- If you want to enable the REST support for a Virtual Web Service, ensure that the **Enable REST Support** action is included in the **Receive** stage for the Service.

If you include the **Enable REST Support** policy action in a SOAP Service configuration, clients who can only send REST requests can now invoke a REST-enabled SOAP Service using both a SOAP request and a REST request in Mediator, and using a REST request in API Portal.

- If you are using Mediator as your gateway, you must include at least one Evaluate * action in order to identify or validate the consumers.
- Be aware that actions from the WS-I category cannot be combined with other types of actions. Also be aware that when you add a WS-I action to the action list, CentraSite will automatically add dependent actions to the list as necessary.
- When you configure certain policy actions, for example, a Context Based Routing action, a Throttling Traffic Optimization action, or a Monitor Service Level Agreement (SLA) action, the action's **Configure** dialog would exhibit the following behavior:
 - If the API provider has configured **API Key Consumption Settings** for API Key and/or OAuth2 token authentication, the **Consumer Applications** drop-down list displays the available the consumer applications that are identified by asset instances of the API Key and/or OAuth2 Client type and that are linked to the API, and all of the consumer applications that are identified by asset instances of the type Application in the CentraSite registry.
 - If the API provider has not configured **API Key Consumption Settings** for API Key and/or OAuth2 token authentication, the **Consumer Applications** drop-down lists all of the consumer applications that are identified by asset instances of the type Application in the CentraSite registry.

Ways in Which You Assign Run-Time Actions to Virtual Services

You can assign run-time actions to an existing Virtual Service in the following two ways:

- Using the **Virtualize** action in the details page of the Native Service.
- Using the **Virtualize** action in the details page of the Virtual Service.

Modify Policy Action List

You use panel 2 of the Virtualize Your API page to modify the list of policy actions that are available for the Virtual Service.

> To modify the action list for a Virtual Service

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, Virtual Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **Virtual Service**.
 - c. Click **OK**.

A list of defined Virtual Service assets is displayed in the Search Results page.

5. Click the Virtual Service whose action list you want to examine and modify.

This opens the Virtual Service details page. Also, the actions bar displays a set of actions that are available for working with the Virtual Service.

6. On the actions bar of the Virtual Service details page, click **Virtualize**.

This opens the **Virtualize <Service_Name> (Step 1 of 2)** wizard.

7. To add an action to the **Message Flow** area, proceed as follows:

- a. In the **Policy Actions** area, expand the desired accordion (Request Handling, Policy Enforcement, Response Handling, or Error Handling).
- b. Locate the action you want to add for the Virtual Service.
- c. Drag and drop the action in the appropriate stage (Receive, Enforce, Routing, or Response) in the **Message Flow** area.
- d. Repeat the above steps for each action that you want to add.

8. To configure the parameter values for any new actions that you might have added to the policy action list, or to make any necessary updates to the existing action parameter values, see [“Configure Policy Action Parameters” on page 1046](#).

9. To remove an existing action from the **Message Flow** area, proceed as follows:

- a. Hover over the action you want to remove.

This causes a **Delete** icon to appear, that you can use for removing the action.

- b. Click **Delete**.

- c. Repeat the above steps for each action that you want to remove.

10. Click **Virtualize**.

Configure Policy Action Parameters

Policy actions have parameters that you must set to configure the behavior of the action at enforcement time. You use panel 2 of the Virtualize Your API page to configure parameters of the policy actions that are available for the Virtual Service.

> To configure the action parameters

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, Virtual Service, click **Choose**.

4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:

- a. Click the chevron next to **Assets** option button.
- b. In the displayed list of asset types, select **Virtual Service**.
- c. Click **OK**.

A list of defined Virtual Service assets is displayed in the Search Results page.

5. Click the Virtual Service whose policy action parameters you want to configure.

This opens the Virtual Service details page. Also, the actions bar displays a set of actions that are available for working with the Virtual Service.

6. On the actions bar of the Virtual Service Details page, click **Virtualize**.

This opens the **Virtualize <Service_Name> (Step 1 of 2)** wizard.

7. To configure the required parameters for the policy actions displayed in the **Message Flow** area, proceed as follows:

- a. Hover over an action whose parameters you want to modify.
- b. Choose the **Configure** icon to the right of the action name.

- c. In the **<action_name>** dialog box, set the values for the parameters as necessary.
- d. Click **OK** to save the parameter settings.

Note:

If you fail to specify the required parameters, the system alerts you with a red error icon. Pointing to the error icon shows a hint with the error description.

- e. Repeat the above steps for each action that you want to modify.

Important:

If you make changes to the Service's run-time enforcement, for example, **Enable REST Support** action for a SOAP Service, you must manually republish the Virtual Service to put those changes into effect.

8. Click **Virtualize**.

Reconfiguring Virtual Services

You might want to reconfigure the details of a virtual service to modify some of its policy parameters. When reconfiguring the virtual service, CentraSite checks for any recent changes in the WSDL associated with its native service. If there is any change in the WSDL of native service, the change is updated in the WSDL of virtual service.

Note:

You cannot reconfigure virtual services if the CentraSite run-time aspects are not enabled.

You can reconfigure the details of a in one of the following ways:

- **Through the virtual service details page.** You will be allowed to directly reconfigure the displayed virtual service.
- **Through the native service details page.** You will be redirected to select the virtual service and reconfigure the selected virtual service.

To reconfigure the details of a virtual service, you must have the following roles or permissions:

- CentraSite Administrator
- Organization Administrator
- Asset Provider
- Instance-level Modify permission on the Native Service or virtual service

In addition to the above described roles and permissions for modifying the details of a virtual service, make sure that you have the specific roles and permissions required for reconfiguring the run-time actions and republishing the virtual services to API Gateway, API Portal, Mediator, and Insight Server gateways.

Note:

CentraSite does not support reconfiguring the details of a Virtual OData Service asset through the Virtual OData Service Details page.

➤ **To reconfigure a virtual service asset**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, Service, click **Choose**.

4. In the **Choose Asset Types** dialog box, select the **Assets** button, and follow these steps:

- a. Click the chevron next to **Assets** button.
- b. In the list of asset types, select any of the following types: **Virtual Service**, **Virtual REST Service**.
- c. Click **OK**.

5. Click the virtual service you want to reconfigure.

This opens the virtual service details page. Also, the Actions bar displays a set of actions that are available for working with the displayed virtual service.

6. On the actions bar of the virtual service details page, click **Virtualize**.

This opens the **Virtualize <Service_Name> (Step 1 of 3)** wizard.

7. To modify the run-time policy actions that are configured for the virtual service, click **Next**.

8. In the **Virtualize <Service_Name> (Step 2 of 3)** wizard, drag and drop the policy actions, and reconfigure the policy action parameters, as required.

9. Click **Virtualize** to save the updated changes.

Publishing and Unpublishing Services to and from Runtime Gateways

CentraSite allows you to publish and unpublish services to and from webMethods API Gateway and webMethods Mediator.

To publish and unpublish services to and from API Gateway, you must have the following roles or permissions:

- CentraSite Administrator
- Organization Administrator
- API Gateway Publisher
- Instance-level Modify permission for API Gateway

To publish and unpublish services to and from Mediator, you must have the following roles or permissions:

- CentraSite Administrator
- Organization Administrator
- Mediator Publisher
- Instance level Modify permission for Mediator gateway
- If you have the CentraSite Administrator role, you can publish and unpublish services to and from any gateways within any organization.
- If you have the Organization Administrator role, API Gateway Publisher role, or Mediator Publisher role for an organization, you have the ability to publish and unpublish services to and from the API Gateway or Mediator gateway within the specific organization.

The following section describes some aspects of runtime you need to consider when you publish and unpublish services to API Gateway and Mediator gateways:

- You can only publish virtual services to the gateways for which you have the required roles or the Modify instance-level permission.
- You cannot publish a native service to both API Gateway and Mediator gateways.
- When publishing a virtual service to both API Gateway and Mediator gateways in a single step, only if publishing to at least one Mediator gateway was successful, CentraSite will publish the service to API Gateway.
- When publishing a service to one or more gateways, if CentraSite encounters a publish failure with one of the gateway, it immediately ignores the failure gateway and proceeds with the next gateway. Any gateway that encountered failure during the publish process is displayed in the **Publish Log**.
- When trying to publish a service to a set of gateways, if the service is already published to a selected gateway, then that service is republished to that particular gateway.

Note:

When CentraSite version 9.12 is configured with Mediator version 9.10 or earlier, publishing and republishing a Virtual REST Service asset with HTTP Patch method fails in Mediator. This is because, Mediator version 9.10 and earlier do not support the Axis Free Mediation stack. Instead, the Mediator versions 9.10 and earlier support the WSStack Mediation stack. In order

to publish a Virtual REST service with HTTP Patch operation, it is mandatory that the Mediator gateway is configured for the Axis Free Mediation stack. In addition, make sure that the Mediator instance is of version 9.12 or above.

Publishing Services to Gateways

You can execute a **Publish** action through the following pages:

- Native Service asset details page, the exact rendering of user interface depends on whether or not the Native Service has an immediate Virtual Service.

If the Native Service does not have an immediate Virtual Service, you will be allowed to directly publish the Native Service to gateways. If the Native Service has an immediate Virtual Service, you will be directed to select the Virtual Services, and publish the selected Virtual Services to Mediator, API Gateway, and Insight Server gateways.

- Virtual Service asset details page, you will be allowed to directly publish the Virtual Service to Mediator, API Gateway, and Insight Server gateways.
- Search Results page, you will be allowed to directly publish Native and Virtual Services to Mediator, API Gateway, and Insight Server gateways in a single step.

> To publish a Service asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** button, and follow these steps:
 - a. Click the chevron next to **Assets** button.
 - b. Based on the type of Service asset that you want to publish to the gateway, select any of the following:
 - Service
 - REST Service
 - OData Service
 - Virtual Service

- Virtual REST Service
 - Virtual OData Service
- c. Click **OK**.
5. Click the Service asset you want to publish to the gateways.
- This opens the Service asset details page. Also, the actions bar displays a set of actions that are available for working with the displayed Service asset.
6. Click **Publish**.
- CentraSite directs you to an intermediate **Virtual Alias** dialog box, if the displayed Service asset has a Virtual Service asset. A list of the defined Virtual Service assets that are available for publishing to the gateways is displayed.
 - CentraSite directs you to the **Publish** dialog box, if the displayed Service asset does not have an immediate Virtual Service asset.
7. In the **Virtual Alias** dialog box, select one or multiple Virtual Services you want to publish to the gateways, and click **Next**.
- This opens the **Publish** dialog box.
8. In the **Gateway** list, select one or multiple gateways to which you want to publish the Virtual Service assets. (The **Gateway** list only displays the gateways for which you have the Modify permission.)

The **Gateway** list includes:

If you choose...	CentraSite displays...
API Gateway	All instances of API Gateway.
Mediator	All instances of Mediator gateway.

The default is set to **API Gateway**.

The **Gateway** list provides the following information about each gateway:

Column	Description
Name	Name of the gateway.
Type	Type of gateway. Supported gateways are: <ul style="list-style-type: none"> ■ API Gateway ■ Mediator

Column	Description
Sandbox	The category to classify deployment endpoint of the gateway.
Settings	<p>The Configure icon displays the API Portal Settings dialog box that allows you to configure a set of sandbox categories for the specified API Gateway individually.</p> <p>The dialog box also lists the set of API communities available for the specified API Gateway. You can assign a Service asset to one or more API communities.</p>

9. Applicable for API Gateways. *Optional.* In the **Sandbox** list, do the following:

- a. Click the **Configure** icon next to the field labeled **Sandbox**.

This opens the **API Gateway Settings** dialog box.

- b. Select one or multiple sandbox categories that classify the endpoint of the Service asset for publishing to API Gateways.

The **Sandbox** list includes:

If you choose...	CentraSite displays...
Development	Instances of a specific gateway whose endpoints are classified by the Development category.
Production	Instances of a specific gateway whose endpoints are classified by the Production category.
Test	Instances of a specific gateway whose endpoints are classified by the Test category.
All Sandboxes	Instances of a specific gateway whose endpoints are classified by any of the above mentioned categories.

The default is set to **All Sandboxes**.

- c. Select one or multiple communities available for a particular API Gateway.

The default is set to **Public Community**. Any new community assigned to the Service asset overwrites the existing community assignments.

- d. Click **OK**.

10. Applicable for Mediator gateways. *Optional.* Select the **Expose to Consumers** check box to allow guest users to simply access and examine the usage of Service asset through CentraSite Business UI.

11. Click **Publish**.

A **Publish Inprogress** popup displays the progress state of publishing the Service asset to selected gateways.

If the publish process logs failures, identify and correct the failure and then try publishing the Service asset again.

Unpublishing Services from Gateways

You can execute a **Unpublish** action through the following pages:

- Native Service asset details page, the exact rendering of user interface depends on whether or not the Native Service has an immediate Virtual Service.

If the Native Service does not have an immediate Virtual Service, you will be allowed to directly unpublish the Native Service from an API Gateway gateway. If the Native Service has an immediate Virtual Service, you will be directed to select the Virtual Services, and unpublish the selected Virtual Services from Mediator, API Gateway, and Insight Server gateways.

- Virtual Service asset details page, you will be allowed to directly unpublish the Virtual Service from Mediator, API Gateway, and Insight Server gateways.
- Search Results page, you will be allowed to directly unpublish multiple instances of native and Virtual Services from Mediator, API Gateway, and Insight Server gateways in a single step.

➤ To unpublish a Service asset

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** button, and follow these steps:
 - a. Click the chevron next to **Assets** button.
 - b. Based on the type of Service asset that you want to unpublish from the gateway, select any of the following:
 - Service

- REST Service
- OData Service
- Virtual Service
- Virtual REST Service
- Virtual OData Service

c. Click **OK**.

5. Click the Service asset you want to unpublish from the gateways.

This opens the Service asset details page. Also, the actions bar displays a set of actions that are available for working with the displayed Service asset.

6. Click **Unpublish**.

- CentraSite directs you to an intermediate **Virtual Alias** dialog box, if the displayed Service asset has a Virtual Service asset. A list of the defined Virtual Service assets that are available for unpublishing from the gateways is displayed.
- CentraSite directs you to the **Unpublish** dialog box, if the displayed Service asset does not have an immediate Virtual Service asset.

7. In the **Virtual Alias** dialog box, select one or multiple Virtual Service assets you want to unpublish from the gateways, and click **Next**.

This opens the **Unpublish** dialog box.

8. In the **Gateway** list, select one or multiple gateways from which you want to unpublish the Virtual Service assets. (The **Gateway** list only displays the gateways for which you have the Modify permission.)

The **Gateway** list includes:

If you choose...	CentraSite displays...
API Gateway	All instances of API Gateway.
Mediator	All instances of Mediator gateway.

The default is set to **API Gateway**.

9. Applicable for API Gateways. *Optional*. In the **Sandbox** list, do the following:

a. Click the **Configure** icon next to the field labeled **Sandbox**.

This opens the **API Gateway Settings** dialog box.

- b. Select one or multiple sandbox categories that classify the endpoint of the Service assets for unpublishing from API Gateways.

The **Sandbox** list includes:

If you choose...	CentraSite displays...
Development	Instances of a specific gateway whose endpoints are classified by the Development category.
Production	Instances of a specific gateway whose endpoints are classified by the Production category.
Test	Instances of a specific gateway whose endpoints are classified by the Test category.
All Sandboxes	Instances of a specific gateway whose endpoints are classified by any of the above mentioned categories.

The default is set to **All Sandboxes**.

- c. Click **OK**.
10. Applicable for Mediator gateways. *Optional*. Select the **Revoke Consumability** check box to revoke the accessibility of a Service asset from the Guest users.
11. Click **Unpublish**.

The **Unpublish Inprogress** popup displays the progress state of unpublishing the Service asset from the selected gateways.

If the unpublish process logs failures, identify, and correct the failure, and try unpublishing the Service asset again.

REST Service Deployment and Redeployment of REST services in CentraSite

A REST service can be deployed to Mediator in two different modes: `wsstack` and `axis-free`.

The REST service deployment mode can be configured in the configuration file, `centrasite.xml`.

```
....
<DeploymentSettings>
  <!-- axis-free or wsstack -->
  <RestServiceStack>axis-free</RestServiceStack>
</DeploymentSettings>
...
```

The configured deployment value is exposed in the **About** page of CentraSite Business UI.

At deployment, the configured value is taken for generating the virtual REST service in Mediator. The resulting mode can also be viewed in the service page of Mediator.

At redeployment, the current state together with the configuration setting is used to determine the final deployment mode. Following table illustrates the behavior on a REST service redeployment:

Configuration value	Service deployment mode	Mode after redeployment
wsstack	wsstack	wsstack
wsstack	axis-free	axis-free
axis-free	wsstack	axis-free
axis-free	axis-free	axis-free

Promoting Virtual Service

You can promote a virtual service from a gateway instance to the other using the Promote functionality.

> To promote a virtual service

1. In CentraSite Business UI, access the details page of the virtual service that you want to promote. To search for a virtual service, follow the procedure explained in the [“Viewing Virtual Service List” on page 984](#).

2. Click **Promote**.

The Promote page is displayed. The page displays the source gateways (the current instance of the service), and target gateways (to which the service can be promoted to).

3. (Optional) To view the list of gateways in a particular sandbox, select the required entry from the **Sandbox** drop-down lists in the **Source Gateway(s)** and **Target Gateway(s)** sections.
4. Select the gateway from which the service has to be promoted, from the **Source Gateway(s)** section.
5. Select the gateway to which the service has to be promoted, from the **Target Gateway(s)** section.
6. Click **Ok**.

A **Promote Inprogress** popup displays the promotion status in the selected gateways.

If the process logs fails, identify and correct the failure and then try promoting the service again.

Managing Virtual Service Assets through CentraSite Control

This section describes operations you can perform to manage virtual service assets such as, Virtual Services, Virtual REST Services, and Virtual OData Services through CentraSite Control.

Virtual SOAP Service Management

This section describes operations you can perform to manage Virtual SOAP Services through CentraSite Control.

Adding Virtual Service from an Existing Service

To create and manage a Virtual Service asset in CentraSite Control, you must have the following permissions:

- Create Assets —OR— Manage Assets
- Manage Runtime Policies (required to configure Virtual Services)
- Mediator Publisher (required to deploy Virtual Services)

Note:

If user has View permission on a Web Service and Create Assets permission for the organization where you will add the Virtual Service, then the user can virtualize that Service. However, the user will not be permitted to configure the processing steps for the Virtual Service that is virtualized unless the user also has the Manage Runtime Policies permission for that organization. Only users with Manage Runtime Policies permission can configure these steps.

Consider identifying a small group of users who will be responsible for configuring the processing steps for a Virtual Service and give this group a role that includes the Manage Runtime Policies permission. Because these users might configure the Virtual Services that other users have added to the catalog, they will also need Modify permission on the actual Virtual Service. To ensure that these users have access to the Virtual Services that they need to configure, consider creating a design/change-time policy that automatically gives this group of users Modify permission on the Virtual Service after it is virtualized.

The following general guidelines apply when adding a Virtual Service in CentraSite Control:

- Ensure that the interface for the Native Service is completely implemented, and that interface is reflected in the WSDL file that is registered for the Web Service in the CentraSite registry.
- An instance of the Web Service is deployed and running at a known point in network.
- The metadata for the Native Service is valid and up-to-date. If the metadata for the Native Service has not been completely specified or is out-of-date, you should update it before you generate the Virtual Service so that you do not carry inaccurate/incomplete data into the Virtual Service.

➤ To add a Virtual Service asset from an existing (Native) Service

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the list of asset types, select **Service**.
3. In the **Assets** pane, right-click a Web Service for that you want to create a proxy service, and then click **Details**.

The Web Service Details page is displayed.

4. On the **Actions** menu, click **Virtualize**.

Important:

You can not virtualize a Web Service asset that does not have an associated WSDL file.

5. In the **Virtualize Web Service** wizard, provide the required information for each of the displayed data fields.

Field	Description
Name	<p>(Optional). Name of the Virtual Service.</p> <p>The name of a Virtual Service asset must be NCName-conformant, meaning that:</p> <ul style="list-style-type: none"> ■ The name must begin with a letter or the underscore character (<code>_</code>). ■ The remainder of the name can contain any combination of letters, digits, or the following characters: <code>.</code> <code>-</code> <code>_</code> (that is, period, dash, or underscore). It can also contain combining characters and extender characters (for example, diacriticals). ■ The name cannot contain any spaces. ■ Furthermore, if the Virtual Service name contains any non-conformant character, upon publishing the Virtual Service to any gateway, the non-conformant character is simply replaced with the underscore character (<code>_</code>) in Mediator. However, in CentraSite the Virtual Service name defined by you is displayed. <p>For more information about the NCName type, see http://www.w3.org/TR/xmlschema-2/#NCName</p> <p>Note: The name of a Virtual Service asset must be unique within an organization. If, for example, a Virtual Service asset with the same name already exists in the CentraSite registry, a warning message will be issued.</p>
Description	(Optional). The description for the Virtual Service.

Field	Description
	<p>Note: This is the description information that users will see when they view this Virtual Service asset in the CentraSite user interfaces. Therefore, we recommend that you specify a meaningful description for each Virtual Service.</p>
Organization	<p>The organization to which you want to add the Virtual Service. (The Organization list only displays organizations for which you have the Manage Assets permission or at least the Create Assets permission.)</p> <p>Note: Select the organization with care. You cannot change the organization assignment after the Virtual Service is added to the catalog. You can, however, export a Virtual Service from one organization and import it to another.</p>
Version	<p>(Optional). The version identifier for the Virtual Service. You can enter any string in this field, that is, the version identifier does not need to be numeric. You can also leave the field blank. You can later create new versions of the Virtual Service to indicate that the Virtual Service has been updated. The default is 1.0.</p> <p>Examples:</p> <pre data-bbox="511 1081 1385 1207">0.0a 1.0.0 (beta) Pre-release 001 V1-2007.04.30</pre> <p>You can use any versioning scheme you choose. The version identifier does not need to be numeric. This is the "public" version identifier that CentraSite Control shows to users when it displays the list of services.</p> <p>In addition, CentraSite automatically generates a system version number, which is visible on the Virtual Service detail page. The system version number is independent from the version number you specify here. For more information, see the <i>CentraSite User's Guide</i>.</p>
Create a Run-Time Policy	<p>Select this check box to specify the behavior for the run-time policy (or policies) associated with the Virtual Service. If you select the check box, the run-time policy or policies associated with the Virtual Service will be created automatically when the Web Service is virtualized. If you do not select the check box, the run-time policy or policies will not be created when the Web Service is virtualized.</p>

- | Field | Description |
|-------|--|
| | This check box is not disabled, since the run-time policy creation from CentraSite is disabled by default. However, you can enable the same by modifying the <code>centrasite.xml</code> file. For the procedure, see “Enabling CentraSite Run-Time Aspects” on page 951 . If the CentraSite run-time policy configuration is enabled, you can select the check box if you have the Manage Runtime Policies organization-level permission. |
- Click **Next**.
 - In the **Attribute Mapping** dialog box, choose the attributes of the Web Service you want to copy to the Virtual Service.

Follow these general guidelines, when specifying attributes for the Virtual Service:

 - If you choose a parent attribute, by default, *all* of its child attributes are copied as well.
 - To remove a parent attribute or a child attribute, clear the corresponding check box.
 - When you cancel the selection of a parent attribute, CentraSite revokes the selection of all its child attributes.
 - Attributes defined as **required** in the asset’s type definition and all referenced objects to which the asset has an association are internally copied from Native Service to the Virtual Service. By default, the check boxes of these attributes are selected and disabled in the **Attribute Mapping** dialog box.
 - Click **Finish**.

The WSDL interface and the entry/exit protocol (for example, HTTP, HTTPS or JMS) of the Virtual Service are identical to the Web Service.

The details page of the newly created Virtual Service asset is displayed.
 - Configure the extended attributes (contained within the profiles) of the Virtual Service asset to suit your requirements.

Note:

Although the type definition of a Web Service asset contains one or more custom profiles, the newly created Virtual Service will not contain any of these custom profiles.

Adding Virtual Service from Scratch

To create and manage a Virtual Service asset in CentraSite Control, you must have the following permissions:

- Create Assets —OR— Manage Assets

- Manage Runtime Policies (required to configure Virtual Services)
- Mediator Publisher (required to deploy Virtual Services)

Note:

If user has View permission on a Web Service and Create Assets permission for the organization where you will add the Virtual Service, then the user can virtualize that Service. However, the user will not be permitted to configure the processing steps for the Virtual Service that is virtualized unless the user also has the Manage Runtime Policies permission for that organization. Only users with Manage Runtime Policies permission can configure these steps.

Consider identifying a small group of users who will be responsible for configuring the processing steps for a Virtual Service and give this group a role that includes the Manage Runtime Policies permission. Because these users might configure the Virtual Services that other users have added to the catalog, they will also need Modify permission on the actual Virtual Service. To ensure that these users have access to the Virtual Services that they need to configure, consider creating a design/change-time policy that automatically gives this group of users Modify permission on the Virtual Service after it is virtualized.

The following general guidelines apply when adding a Virtual Service in CentraSite Control:

- Ensure that the interface for the Native Service is completely implemented, and that interface is reflected in the WSDL file that is registered for the Web Service in the CentraSite registry.
- An instance of the Web Service is deployed and running at a known point in network.
- The metadata for the Native Service is valid and up-to-date. If the metadata for the Native Service has not been completely specified or is out-of-date, you should update it before you generate the Virtual Service so that you do not carry inaccurate/incomplete data into the Virtual Service.

➤ **To add a Virtual Service asset from scratch**

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. Click **Add Asset**.
3. In the **Add Asset** dialog box, provide the required information for each of the displayed data fields.

Field	Description
Type	The asset type, Virtual Service .
Name	(Optional). Name of the Virtual Service asset. The name of a Virtual Service asset must be NCName-conformant, meaning that:

Field	Description
	<ul style="list-style-type: none"> <li data-bbox="423 260 1286 327">■ The name must begin with a letter or the underscore character (<code>_</code>). <li data-bbox="423 352 1286 487">■ The remainder of the name can contain any combination of letters, digits, or the following characters: <code>.</code> <code>-</code> <code>_</code> (that is, period, dash, or underscore). It can also contain combining characters and extender characters (for example, diacriticals). <li data-bbox="423 512 948 537">■ The name cannot contain any spaces. <li data-bbox="423 571 1286 747">■ Furthermore, if the Virtual Service name contains any non-conformant character, upon publishing the Virtual Service to any gateway, the non-conformant character is simply replaced with the underscore character (<code>_</code>) in Mediator. However, in CentraSite the Virtual Service name defined by you is displayed. <p data-bbox="423 772 1286 831">For more information about the NCName type, see http://www.w3.org/TR/xmlschema-2/#NCName</p> <div data-bbox="423 856 1286 1058" style="background-color: #f0f0f0; padding: 5px;"> <p>Note: The name of a Virtual Service asset must be unique within an organization. If, for example, a Virtual Service asset with the same name already exists in the CentraSite registry, a warning message will be issued.</p> </div>
Description	<p data-bbox="423 1075 1286 1100">(Optional). The description for the Virtual Service.</p> <div data-bbox="423 1125 1286 1318" style="background-color: #f0f0f0; padding: 5px;"> <p>Note: This is the description information that users will see when they view this Virtual Service asset in the CentraSite user interfaces. Therefore, we recommend that you specify a meaningful description for each Virtual Service.</p> </div>
Organization	<p data-bbox="423 1344 1286 1478">The organization to which you want to add the Virtual Service. (The Organization list only displays organizations for which you have the Manage Assets permission or at least the Create Assets permission.)</p> <div data-bbox="423 1503 1286 1696" style="background-color: #f0f0f0; padding: 5px;"> <p>Note: Select the organization with care. You cannot change the organization assignment after the Virtual Service is added to the catalog. You can, however, export a Virtual Service from one organization and import it to another.</p> </div>
Version	<p data-bbox="423 1722 1286 1818">(Optional). The version identifier for the Virtual Service. You can enter any string in this field, that is, the version identifier does not need to be numeric. You can also leave the field blank. You can later</p>

Field	Description
	<p>create new versions of the Virtual Service to indicate that the Virtual Service has been updated. The default is 1.0.</p> <p>Examples:</p> <pre data-bbox="516 394 1365 520">0.0a 1.0.0 (beta) Pre-release 001 V1-2007.04.30</pre> <p>You can use any versioning scheme you choose. The version identifier does not need to be numeric. This is the "public" version identifier that CentraSite Control shows to users when it displays the list of services.</p> <p>In addition, CentraSite automatically generates a system version number, which is visible on the Virtual Service detail page. The system version number is independent from the version number you specify here. For more information, see the <i>CentraSite User's Guide</i>.</p>

<p>Create a Run-Time Policy</p>	<p>Select this check box to specify the behavior for the run-time policy (or policies) associated with the Virtual Service. If you select the check box, the run-time policy or policies associated with the Virtual Service will be created automatically when the Web Service is virtualized. If you do not select the check box, the run-time policy or policies will not be created when the Web Service is virtualized.</p> <p>You can only select the check box if you have the Manage Runtime Policies organization-level permission; without this permission, the check box is deactivated.</p>
--	---

4. Click **OK**.

Important:

During virtualization of a Web Service asset, CentraSite will not allow you to add the Virtual Service asset to the catalog unless you specify all of the required attributes defined in the asset type definition, and all the referenced objects to which the asset has an association.

The details page of the newly created Virtual Service asset is displayed.

5. Configure the extended attributes (contained within the profiles) of the Virtual Service asset to suit your requirements.

Viewing Virtual Service List

You use the **Asset Catalog > Browse** to display the list of Virtual Service assets in CentraSite.

> To view the list of Virtual Service assets

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Virtual Service**.

The Virtual Service assets (for which you have the View permission) are displayed in the **Assets** pane.

The **Assets** pane provides the following information about each Virtual Service asset:

Column	Description
Name	Name of the Virtual Service asset.
Type	The asset's type definition, Virtual Service.
Version	The user-defined version identifier of the Virtual Service.

You can adjust the view to show or hide the individual column by using the **Browse By and Column Selection** icon that is located in the upper-right corner of the **Assets** pane.

The shortcut menu of a particular Virtual Service asset displays one or more actions that you can perform on that Virtual Service.

Action	Description
Copy	Copies an existing Virtual Service that is similar to the one you need, and allows you to examine and modify the copy. CentraSite treats the copy just as if it were a new Virtual Service that you created from scratch.
Details	Displays the details page of the Virtual Service.
Change Lifecycle State	Changes the lifecycle state of the Virtual Service.
Revert Pending State	Reverts a request that has been submitted for approval, and that is struck in Pending mode.
Change Owner	Changes the user ownership of the Virtual Service.
Run Policy	Executes a set of policies that are applicable for the asset's type definition, Virtual Service.
Change Organization	Changes the organizational ownership of the Virtual Service.
Export	Exports the Virtual Service from the registry to an archive file on the file system.

Action	Description
Impact Analysis	Helps to easily visualize the associations that exist between the Virtual Service asset and registry objects.
Attach WSDL	Appends a WSDL document to the Virtual Service.
Add New Version	Generates a new version of the Virtual Service.
Notify Me	Sends notifications to all the registered users when changes are made to the Virtual Service.
Add to List	Adds the Virtual Service to a list in My Favorites .
Add to Favorites	Adds a shortcut to the Virtual Service you want to use routinely or otherwise keep close at hand.
Virtualize	Creates a proxy for the Virtual Service for consumption.
Download Documents	Downloads files that are attached to the Virtual Service from Supporting Document Library (SDL).

Modifying Virtual Service Details

Note:

Each tab on a Virtual Service asset's details page represents a collection of attributes called a Profile. The Virtual Service asset type has a unique set of profiles. However, an administrator can configure the Virtual Service asset type to display a customized set of profiles and attributes.

The following general guidelines apply when modifying the details of a Virtual Service asset in CentraSite Control:

- If you are not the owner of the Virtual Service, you cannot examine or modify the details of the Virtual Service, unless you have the View or Modify permission on the Virtual Service (granted though either a role-based permission or an instance-level permission).
- When you view the details page of a Virtual Service, you will only see profiles for which you have the profile-level View permission. You will only be able to modify the attributes of the profiles on which you have the Modify permission.
- Some attributes accept only specific types of information. For example, if the Virtual Service asset type includes a URL type attribute, you must supply a URL when you modify that attribute. Other attribute types that require a specific type of value include the Date and Email attributes.
- Some attributes are designed to be read-only. You cannot modify the read-only attributes even if you have the CentraSite Administrator role.
- If you want to change the name of a Virtual Service asset, make sure that the asset is unpublished from the appropriate gateways.

In this task you modify the basic and type-specific attributes for a Virtual Service asset. You can view the bindings, operations, WSDL file, associated schema files, and the external links to the WSDL and schema files.

➤ **To modify the details of a Virtual Service asset**

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Virtual Service**.
3. In the **Assets** pane, right-click the Virtual Service for which you want to modify the attributes, and click **Details**.

The Virtual Service Details page is displayed.

4. To modify the generic attributes of the Virtual Service that are displayed in the area labeled **Basic Information**, change values of the attributes in the respective data fields as required.
5. To modify the extended attributes of the Virtual Service that are displayed in the individual profiles, follow these steps:
 - a. Select the profile for which you want to modify the attributes.
 - b. Change values of the attributes in the respective data fields as necessary.
 - c. Repeat steps 5.a and 5.b for each profile for which you want to modify the attributes.
6. Click **Save** to save the updated changes.

Modifying an Input WSDL File

The Virtual Service asset includes a WSDL file and one or more associated files. You can upload a new file or update an existing file for the Virtual Service accordingly.

On the **Actions** menu, click **Attach WSDL**.

If you are attaching a WSDL file to the Virtual Service which already has a WSDL, the Virtual Service name in the new WSDL must be identical to the Virtual Service name in the existing one or the process will fail.

If you are attaching an abstract WSDL file to an abstract Service asset which already has a WSDL, the Virtual Service name in the new WSDL does not need to be identical to the Virtual Service name in the existing one. Select the **Overwrite Service name** check box to replace the name of the existing attached WSDL with the name of the new attached WSDL.

If you select the option **Interactive resolution of Imports/Includes**, and the attached WSDL contains an Import or Include reference to a WSDL that already exists in the registry, the **Attach WSDL to ...** dialog box allows you to choose whether to retain the existing WSDL or to replace

the existing WSDL by uploading a new one. If you choose to use upload a new WSDL, you can specify whether the new WSDL should overwrite the existing one, or whether a new version of the WSDL should be created. Regardless of whether you overwrite the existing version or create a new version, you can specify a user-defined version number of the uploaded WSDL in the **User Version** field.

If you select the option **Interactive resolution of Imports/Includes**, and the attached WSDL contains an Import or Include reference to a WSDL that does not already exist in the registry, the **Attach WSDL to ...** dialog box allows you to upload the WSDL. You can also specify a user-defined version number of the uploaded WSDL in the **User Version** field.

Configuring Virtual Services

The CentraSite Control enables you to configure the following processing steps for a Virtual Service asset:

Configuring the Entry Protocol Step

The **Entry Protocol** step specifies the protocol (HTTP, HTTPS or JMS) and SOAP format (1.1 or 1.2) of the requests that the Virtual Service will accept.

This step allows you to bridge protocols between the consuming application and the Native Service. For example, suppose you have a Native Service that is exposed over JMS and a consuming application that submits SOAP requests over HTTP. In this situation, you can configure the Virtual Service's Entry Protocol step to accept HTTP requests and configure its Routing Protocols step to route the request to the Web service using JMS.

Besides using the **Entry Protocol** step to resolve protocol differences between the consumer and the Native Service, you might use this step to intentionally expose a Virtual Service over a particular protocol. For example, if you have a Native Service that is exposed over HTTP, you might expose the Virtual Service over JMS simply to gain the asynchronous-messaging and guaranteed-delivery benefits that one gains by using JMS as the message transport.

➤ To configure the Entry Protocol step

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Virtual Service**.
3. In the **Assets** pane, right-click the Virtual Service you want to configure, and click **Details**.

The Virtual Service Details page is displayed.

4. In the **Processing Steps** tab, click **Entry Protocol**.
5. In the **Entry Protocol** tab, specify the protocol (HTTP, HTTPS, or JMS) for the Virtual Service to accept requests, and click **OK**.

Note: CentraSite supports HTTP version 1.1 only.

Field	Description
Protocol	<p>Select the protocol over which the Virtual Service will accept requests. The possible options are: HTTP, HTTPS, JMS.</p> <p>Note that you can select <i>both</i> HTTP and HTTPS if required.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Important: Before you deploy a Virtual Service over HTTPS, ensure that the Integration Server on which the Integration Server is running has been configured for SSL. In addition, make sure you specify an HTTPS port in the Integration Server's Ports Configuration page. (In the Integration Server Administrator, go to Solutions > Mediator > Administration > General and specify the port in the HTTPS Ports Configuration field.) For details on the Port Configuration page, see <i>Administering webMethods Mediator</i>.)</p> </div>
JMS Provider Alias	<p>If you have selected the JMS option, specify the Integration Server's JMS Trigger name. The alias must include the JNDI destination name and the JMS connection factory.</p>
Format	<p>Select the SOAP format (SOAP 1.1 or SOAP 1.2) of the requests that the Virtual Service will accept.</p>

Configuring the Request Processing Step

The **Request Processing** step specifies how the SOAP request message is to be transformed or pre-processed before it is submitted to the Native Service.

As long as a consumer sends a SOAP request to the correct Virtual Service endpoint, and the request includes a soapAction header, then Mediator can detect the correct service and operation; in this case, no message transformation is required. However, in some cases a Virtual Service might need to transform SOAP messages.

For example, you might need to accommodate differences between the message content that a consuming application is capable of submitting and the message content that a Native Service expects. For example, if the consuming application submits an order record using a slightly different structure than the structure expected by the Native Service, you can use the **Request Processing** step to transform the record submitted by the consuming application to the structure required by the Web service.

Specifically, you would need to configure the Virtual Service to:

- Transform or pre-process the request messages into the format required by the Native Service, before Mediator sends the requests to the Native Services. Additionally in this case, the transformation is required if the Virtual Service has a schema validation policy, which validates the requests.

- Transform or pre-process the Native Service's response messages into the format required by the consumer applications, before Mediator returns the responses to the consumer applications.

You can transform or pre-process a message in the following ways:

- By passing the message to an XSLT transformation file.
- By passing the message to a webMethods IS service.

➤ To configure the Request Processing step

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Virtual Service**.
3. In the **Assets** pane, right-click the Virtual Service you want to configure, and click **Details**.

The Virtual Service Details page is displayed.

4. In the **Processing Steps** tab, click **Request Processing**.
5. In the **Request Processing** tab, add one or multiple **Transform** steps, and one or multiple **webMethods IS Service** steps as follows.
 - a. Click **Add Step**.
 - b. Select one of the following kinds of request processing steps.

Request Processing Step

Description

Transform

Configure this step to perform an XSLT message transformation on the request message before it is submitted to the Native Service.

Important:

The XSL file uploaded by the user should *not* contain the XML declaration in it (e.g., `xml version="1.0" encoding="UTF-8"`). This is because when the Virtual Service is deployed to Mediator, Mediator embeds the XSL file in the Virtual Service definition (VSD), and since the VSD itself is in XML format, there cannot be an XML declaration line in the middle of it. This can lead to unexpected deployment issues which can be avoided by making sure the XSL file does not contain the declaration line.

webMethods IS Service

Configure this step to invoke a webMethods IS service to preprocess the request before it is submitted to the

- | Request Processing Step | Description |
|-------------------------|--|
| | Native Service. For more information, see “Virtual Service” on page 1149 . |
- c. Click **OK**.
 - d. In the **Step** list, click the step (**Transform** or **webMethods IS Service**), and provide the required information for each of the displayed fields:

Request Processing Step	Description
Transform	Click Browse , select the XSLT transformation file from your file system, and click OK .

Note:

If you make changes to the XSLT file in the future, you must re-deploy the Virtual Service.

webMethods IS Service	Type the fully qualified service name or, to display a list of webMethods IS services that are published to CentraSite, type a keyword phrase in the Service field. The wildcard character * is supported. For example, to return all IS services that start with Test, type Test*. (The list that appears also identifies the application server instance on which each service is located.) Then select one or more services to be used to manipulate the request (the axis2 MessageContext instance) and click OK .
------------------------------	--

Mediator will pass to the invoked IS service the request message context (the axis2 MessageContext instance), which contains the request-specific information. Thus, you can use the public IS services that accept MessageContext as input to manipulate the response contents. For more information, see [“Virtual Service” on page 1149](#).

Note:

The webMethods IS service must be running on the same Integration Server as Mediator.

- e. Configure additional request processing steps if required, and then click **Save**.

Note:

Specify the steps in the order in which they should be invoked. Use the arrow buttons to rearrange the sequence of steps. To delete a step, select the check box next to the step and click **Delete**.

Configuring the Response Processing Step

The **Response Processing** step is similar to the Request Processing step. This step specifies how the response message from the Native Service provider is to be transformed or processed before it is returned to the consuming application.

You may configure and test a Virtual Service without specifying response processing. You can go back later and specify response processing, if required.

All steps are optional. You can configure the following steps:

- **Error Messaging:** Configure this step to return a custom error message (and the native provider's service fault content) to the consuming application when the native provider returns a service fault. In addition, you can invoke one or more webMethods IS services to process the error message before and after the custom error message is added to the response.
- **Transformation:** Configure this step to perform message transformations using a specified XSLT file.
- **webMethods IS Service:** Invoke a user-defined that processes the response

➤ To configure the Response Processing step

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Virtual Service**.
3. In the **Assets** pane, right-click the Virtual Service you want to configure, and click **Details**.
The Virtual Service Details page is displayed.
4. In the **Processing Steps** tab, click **Response Processing**.
5. In the **Response Processing** tab, add one or multiple **Transform** steps, and one or multiple **webMethods IS Service** steps as follows.
 - a. Click **Add Step**.
 - b. Select one of the following kinds of request processing steps.

Step	Description
Error Messaging	You use this step to configure error responses for this particular Virtual Service. Alternatively, you can configure global error responses for <i>all</i> Virtual Services, using Mediator's Service Fault Configuration page, as described in <i>Administering webMethods Mediator</i> .

Step	Description
	<p>The precedence of the Error Messaging instructions is as follows:</p> <ul style="list-style-type: none">■ If you configure an Error Messaging step for a Virtual Service, the error messaging step takes precedence over any settings on the global <code>Service Fault Configuration</code> page.■ If you do not create an Error Messaging step for a Virtual Service, the settings on the <code>Service Fault Configuration</code> page take precedence.
Transform	Configure this step to invoke an XSLT transformation file to transform the SOAP response payloads from XML format to the format required by the consumer.
webMethods IS Service	Configure this step to invoke a webMethods IS service to process the response message from the Native Service before it is returned to the consuming application.

You can select only one step at a time, but you can go back and select one or more **Transform** steps, one or more **webMethods IS Service** steps, but only one **Error Messaging** step.

- c. Click **OK**.
6. If you have selected the **Error Messaging** step, click **Error Messaging** in the **Step** list, and configure it as follows.

You use this step to configure error responses for this particular Virtual Service. Select one or both of the following options:

Send Native Provider Fault: When you select this option, Mediator returns the Native Service provider's fault content (if available) to the consuming application. Mediator will send whatever content it received from the Native Service provider. If you select this option, the **Response** option is ignored when a fault is returned by the Native Service provider. (Faults returned by internal Mediator exceptions will still be handled by the **Response** option.)

Response: When you select this option, Mediator returns the following fault response to the consuming application:

```
Mediator encountered an error:$ERROR_MESSAGE while executing
operation:$OPERATION service:$SERVICE at time:$TIME on date:$DATE.
The client ip was:$CLIENT_IP. The current user:$USER. The consumer
application:$CONSUMER_APPLICATION".
```

This fault response is returned in both of the following cases:

- When a fault is returned by the Native Service provider.

In this case, the `$ERROR_MESSAGE` variable in the fault response will contain the message produced by the provider's exception that caused the error. This is equivalent to the `getMessage` call on the Java Exception. This maps to the `faultString` element for SOAP 1.1 or the `Reason` element for SOAP 1.2 catch. Mediator discards the Native Service provider's fault and does not return this content to the web service caller since it could be considered a security issue, especially if the native provider is returning a stack trace with its response.

- When a fault is returned by internal Mediator exceptions (such as policy violation errors, timeouts, and so on).

In this case, `$ERROR_MESSAGE` will contain the error message generated by Mediator.

The default fault response contains predefined fault handler variables (`$ERROR_MESSAGE`, `$OPERATION` and so on) which are described in the table below.

You can customize the default fault response using the following substitution variables, where Mediator replaces the variable reference with the real content at run time:

- The predefined context variables.
- Custom context variables that you declare using Mediator's API.

Note:

To reference a custom context variable that you have already defined in a context-based routing rule (as opposed to one you have declared using the Mediator API), you must add the prefix `$mx` to the variable name in order to reference the variable. For example, if you defined the variable `TAXID`, you would reference it as `$mx:TAXID`.

The fault handler variables are described below.

Note:

If no value is defined for a fault handler variable, then the returned value will be the literal string `null`. For example, `$CONSUMER_APPLICATION` will always be `null` if the service's policy does not contain the Identify Consumer action.

Fault Handler Variable	Description
<code>\$ERROR_MESSAGE</code>	The error message produced by the exception that is causing the error. This is equivalent to the <code>getMessage</code> call on the Java Exception. This maps to the <code>faultString</code> element for SOAP 1.1 or the <code>Reason</code> element for SOAP 1.2 catch.
<code>\$OPERATION</code>	The operation that was invoked when this error occurred.
<code>\$SERVICE</code>	The service that was invoked when this error occurred.
<code>\$TIME</code>	The date (as determined on the Container side) at which the error occurred.
<code>\$DATE</code>	The date (as determined on the Container side) at which the error occurred.

Fault Handler Variable	Description
\$CLIENT_IP	The IP address of the client invoking the service. This might be available for only certain invoking protocols, such as HTTP(S).
\$USER	The currently authenticated user. The user will be present only if the Transport/SOAP Message have user credentials.
\$CONSUMER_APPLICATION	The currently identified consumer application.

Pre-Processing: *Optional.* Configure this step to invoke one or more webMethods IS services to manipulate the response message before the **Error Messaging** step is invoked. The IS service will have access to the response message context (the axis2 MessageContext instance) before it is updated with the custom error message. For example, you might want to send emails or perform custom alerts based on the response payload. For more information, see [“Virtual Service” on page 1149](#).

Post-Processing: *Optional.* Configure this step to invoke one or more webMethods IS services to manipulate the service fault after the **Error Messaging** step is invoked. The IS service will have access to the entire service fault and the custom error message. You can make further changes to the fault message structure, if needed. For more information, see [“Virtual Service” on page 1149](#).

7. If you have selected the **Transform** step, click **Transform** in the **Step** list, click **Browse** and select the XSLT transformation file from your file system, then click **OK**. If you make changes to the XSLT file in the future, you must re-deploy the Virtual Service.
8. If you selected the **webMethods IS Service** step, click **webMethods IS Service** in the **Step** list and configure it as follows.

Type the fully qualified service name, or to display a list of webMethods IS services that are published to CentraSite, type a keyword phrase in the **Service** field. The wildcard character * is supported. For example, to return all IS services that start with Test, type Test*. (The list that appears also identifies the application server instance on which each service is located.) Then select one or more services to be used to manipulate the response (the axis2 MessageContext instance) and click **OK**.

Mediator will pass to the invoked IS service the response message context (the axis2 MessageContext instance), which contains the response-specific information. Thus, you can use the public IS services that accept MessageContext as input to manipulate the response contents. For more information, see [“Virtual Service” on page 1149](#).

Note:

The webMethods IS service must be running on the same Integration Server as Mediator.

9. Configure additional response processing steps if required, and then click **Save**.

Arrange the steps in the order in which they should be invoked. Use the arrow buttons to rearrange the sequence of steps. To delete a step, select the check box next to the step and click **Delete**.

Configuring Straight Through Routing

If your Entry Protocol is HTTP or HTTPS, you can select the Straight Through routing protocol. (Alternatively, you can choose Content-based, Context-based or Load Balancing routing.)

When you select the Straight Through routing protocol, the Virtual Service routes the requests directly to the Native Service endpoint you specify. You may specify how to authenticate requests (as with all routing protocols).

Alternatively, you can choose the Content-Based, Context-Based, or Load Balancing routing protocol.

➤ To configure Straight Through routing

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Virtual Service**.
3. In the **Assets** pane, right-click the Virtual Service you want to configure, and click **Details**.

The Virtual Service Details page is displayed.

4. In the **Processing Steps** tab, click **Routing Protocols**.
5. In the **Routing Protocols** tab, provide the required information for each of the displayed fields, and click **Save**:

Field	Description
HTTP or JMS	Select HTTP .
Routing Type	Select Straight Through
Default To	Click Endpoint , and select the URL of the Native Service to route the request to.

Alternatively, Mediator offers Local Optimization capability if the Native Service and the Virtual Service (in Mediator) are located on the same machine. With local optimization, service invocation happens in-memory and not through a network hop. In the **Default To** field, specify the Native Service in either of the following forms:

```
local://<service_full_path>
```

OR

Field	Description
	<pre data-bbox="380 247 1271 281">local://<server>:<port>/ws/<service_full_path></pre> <p data-bbox="375 317 545 342">For example:</p> <pre data-bbox="380 365 1271 428">local://MediatorTestServices:New MediatorTestServices_Port</pre> <p data-bbox="375 464 1276 594">which points to the endpoint service <code>NewMediatorTestServices_Port</code> which is present under the folder <code>MediatorTestServices</code> in Integration Server. This works for HTTP endpoints only, for all types of Routing Protocols.</p>
Configure Endpoint Properties (icon)	<p data-bbox="375 625 1276 693">The Configure Endpoint Properties icon displays a dialog box that enables you to configure a set of properties for the endpoint as follows:</p> <ul style="list-style-type: none"> <li data-bbox="375 720 1276 821">■ SOAP Optimization Method: Optional. Mediator can accept the following optimization methods to optimize the payloads of SOAP requests: <ul style="list-style-type: none"> <li data-bbox="423 848 1276 982">■ MTOM: Indicates that Mediator expects to receive a request with a Message Transmission Optimization Mechanism (MTOM) attachment, and will forward the attachment to the Native Service. <li data-bbox="423 1010 1276 1113">■ SwA: Indicates that Mediator expects to receive a SOAP with Attachment (SwA) request, and will forward the attachment to the Native Service. <li data-bbox="423 1140 719 1171">■ None (the default). <ol style="list-style-type: none"> <li data-bbox="423 1199 1276 1440">1. Bridging between SwA and MTOM is not supported. If a consumer sends an SwA request, Mediator can only forward SwA to the native provider. The same is true for MTOM, and applies to responses received from the native provider. That is, an SwA or MTOM response received by Mediator from a native provider will be forwarded to the caller using the same format it received. <li data-bbox="423 1467 1276 1749">2. When sending SOAP requests that do <i>not</i> contain a MTOM or SWA attachment to a Virtual Service for a native provider endpoint that returns an MTOM or SWA response, the request 'Accept' header must be set to <code>multipart/related</code> (or the Virtual Service's Request Processing Step should include an “Virtual Service” on page 1149 that sets the <code>BUILDER_TYPE</code> context variable to <code>multipart/related</code>). This is necessary so Mediator knows how to parse the response properly. <ul style="list-style-type: none"> <li data-bbox="375 1776 1276 1871">■ WSS Header Customization: Indicates whether Mediator should pass the WS-Security headers of the incoming requests to the Native Service.

Field	Description
	<ul style="list-style-type: none"> <li data-bbox="516 258 1370 394">■ Pass all security headers: Passes the security header, even if it is processed by Mediator (that is, even if Mediator processes the header according to the Virtual Service's security run-time policy). <div data-bbox="565 415 1365 617" style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;"> <p data-bbox="570 426 646 453">Note:</p> <p data-bbox="570 464 1354 596">If the Virtual Service does not contain a security run-time policy, and the <code>mustUnderstand</code> attribute of the security header is 0 or false, then Mediator will <i>always</i> forward the security header to the Native Service.</p> </div> <ul style="list-style-type: none"> <li data-bbox="516 632 1382 909">■ Remove processed security header from request before routing: Removes the security header if it is processed by Mediator (that is, if Mediator processes the header according to the Virtual Service's security run-time policy). Note that Mediator will <i>not</i> remove the security header if <i>both</i> of the following conditions are true: 1) Mediator did not process the security header, and 2) the <code>mustUnderstand</code> attribute of the security header is 0 or false). <li data-bbox="469 936 1382 1178">■ HTTP Connection Timeout: The time interval (in seconds) after which a connection attempt will timeout. If a value is not specified (or if the value 0 is specified), Mediator uses the value of the global property <code>pg.endpoint.connectionTimeout</code> located in the file <code>Integration Server_directory\packages\WmMediator\config\resources\pg-config.properties</code>. The default of that property is 30 seconds. <li data-bbox="469 1205 1373 1274">■ Read Timeout: The time interval (in seconds) after which a socket read attempt will timeout. <p data-bbox="516 1302 1338 1333">The precedence of the Read Timeout configuration is as follows:</p> <ol style="list-style-type: none"> <li data-bbox="516 1360 1382 1570">1. If a value is specified for the Read Timeout field in the routing endpoint alias, Mediator will use the value specified in the Runtime Alias > Endpoint Alias > Endpoint Properties > Read Timeout field. The read timeout value defined at an alias level takes precedence over the timeout values defined at an API level and the global configuration. <li data-bbox="516 1598 1382 1766">2. If a value 0 is specified (or if the value is not specified) for the Read Timeout field in the routing endpoint alias, then Mediator will use the value specified in the Read Timeout field of this step. The read timeout value defined at an API level takes precedence over the global configuration. <li data-bbox="516 1793 1370 1860">3. If a value 0 is specified (or if the value is not specified) for the Read Timeout field in this step (at an API level), then Mediator

Field	Description
	<p>will use the value of the global property <code>pg.endpoint.readTimeout</code> located in the file <i>Integration Server_directory\packages\WmMediator\config\resources\pg-config.properties</i> (in the Mediator Administration console, go to > Settings > Extended Settings > pg.endpoint.readTimeout property.).</p>
	<p>Note: If a value for the Read Timeout configuration is not specified in any of the above configuration parameters, then Mediator will use the default 30 seconds.</p>
	<ul style="list-style-type: none"> ■ SSL Options: To enable SSL client authentication for the endpoint, you must specify values for both the Client Certificate Alias field and the IS Keystore Alias field. If you specify a value for only one of these fields, a deployment error will occur.
	<p>Note: SSL client authentication is optional; you may leave both fields blank.</p>
	<ul style="list-style-type: none"> ■ Client Certificate Alias: The client's private key to be used for performing SSL client authentication. If you specify a client certificate alias, you must also include in the Virtual Service's policy the Require SSL action and select that action's Client Certificate Required option. The Client Certificate Required option specifies whether client certificates are required for the purposes of: 1) Verifying the signature of signed SOAP requests or decrypting encrypted SOAP requests, and 2) Signing SOAP responses or encrypting SOAP responses. ■ IS Keystore Alias: The keystore alias of the instance of Integration Server on which Mediator is running. This value (along with the value of Client Certificate Alias) will be used for performing SSL client authentication.
<p>HTTP Authentication</p>	<p>Authentication Scheme: Specify the mode of authentication: Basic Authentication (default), NTLM, OAuth2, or None.</p>
	<p>Basic Authentication. Select one of the following options:</p> <ul style="list-style-type: none"> ■ Use credentials from incoming request: (default): Authenticates requests based on the credentials specified in the HTTP header. Mediator passes the 'Authorization' header present in the original client request to the Native Service. ■ Use specified credentials: Authenticates requests according to the values you specify in the User, Password and Domain fields.

Field	Description
	<p>NTLM. Note that if Mediator is used to access a Native Service protected by NTLM (which is typically hosted in IIS), then the Native Service in IIS should be configured to use NTLM as the authentication scheme. If the authentication scheme is configured as Windows, then NTLM should be in its list. The Negotiate handshake will be supported in the near future. This note applies to all three of the following options for NTLM:</p> <ul style="list-style-type: none"> ■ Use credentials from incoming request: Default. Mediator uses the user credentials passed in the request header for an NTLM handshake with the server. ■ Use specified credentials: Mediator uses the values you specify in the User, Password and Domain fields for an NTLM handshake with the server. ■ Transparent: If the property <code>watt.pg.disableNtlmAuthHandler</code> is set to <code>false</code> (the default), then Mediator will behave in pass by mode, allowing an NTLM handshake to occur between the client and server. If the property <code>watt.pg.disableNtlmAuthHandler</code> is set to <code>true</code>, then Mediator performs the Kerberos Windows authentication (and not NTLM Windows authentication). This property is located in <code>Integration Server_directory\instances\instance_name\config\server.cnf</code>. Note: If the client is a WCF application, then the client should be configured with <code>clientCredentialType</code> set to NTLM.
	<p>OAuth2. Select one of the following options:</p> <ul style="list-style-type: none"> ■ Use credentials from incoming request: Default. This is known as pass through mode, in which the consumer includes an OAuth2 access token (a Bearer type token) in the request. Mediator then passes the access token unchanged to the native OAuth server. ■ Use specified token: In this mode, the consumer does not include an OAuth2 access token in the request. Instead, the provider generates an OAuth2 access token for each consumer, and Mediator stores the access tokens in Passman. When consumers send requests, Mediator obtains the OAuth2 access tokens from Passman and uses them to access the Native Services. Specify an OAuth access token to be deployed by Mediator. If you select this option, the consumer need not pass the OAuth token during service invocation. Click Show Token to view the OAuth access token. Users who do not have the permissions to create and manage Virtual Services will not see this button. <p>Specify an OAuth access token to be deployed by Mediator by clicking Show Token and selecting an OAuth access token. Users who do not have the permissions to create and manage Virtual Services will not see this button.</p>

Field	Description
	<p>Note: Here are some general guidelines:</p> <ul style="list-style-type: none"> ■ You must set the Integration Server property <code>watt.server.auth.skipForMediator</code> to <code>true</code> and then restart Integration Server for the change to take effect. This property is located in the server configuration file (<code>server.cnf</code>), which is located in the <code>Integration Server_directory\config</code> directory. For details, see the <i>webMethods Integration Server Administrator's Guide</i>. ■ The run-time actions "Require HTTP Basic Authentication" and "Identify Consumer" (with the value HTTP Authentication Token as the identifier) will not be enforced when using the authentication scheme OAuth2. <p>None. Select the following option:</p> <ul style="list-style-type: none"> ■ Invoke Service Anonymously: Does not authenticate requests.
HTTP Headers	<p>The HTTP headers that you want to use to authenticate the requests.</p> <ul style="list-style-type: none"> ■ Use Existing: Use the HTTP headers that are contained in the requests. ■ Customize: Use the HTTP headers that you specify in the Name and Value columns below. If you need to specify multiple headers, use the plus button to add rows.

Configuring Content-based Routing

If your Entry Protocol is HTTP or HTTPS, you can choose to use the Content-based routing protocol. (Alternatively, you can choose Straight Through, Context-based, or Load Balancing routing.)

If you have a Native Service that is hosted at two or more endpoints, you can use the Content-based routing protocol to route specific types of messages to specific endpoints.

You can route messages to different endpoints based on specific values that appear in the request message. You might use this capability, for example, to determine which operation the consuming application has requested, and route requests for complex operations to an endpoint on a fast machine.

The requests are routed according to the content-based routing rules you create. That is, they are routed based on the successful evaluation of one or more XPath expressions that are constructed utilizing the content of the request payload. For example, a routing rule might allow requests for half of the methods of a particular service to be routed to Service A, and the remaining methods to be routed to Service B.

You may also specify how to authenticate requests (as with all routing protocols).

If a SOAP request contains a WS-Security header, Mediator passes it to the Native Service.

➤ To configure Content-based routing

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Virtual Service**.
3. In the **Assets** pane, right-click the Virtual Service you want to configure, and click **Details**.
The Virtual Service Details page is displayed.
4. In the **Processing Steps** tab, click **Routing Protocols**.
5. In the **Routing Protocols** tab, provide the required information for each of the displayed fields, and click **Save**.

Field	Description
HTTP or JMS	Select HTTP .
Routing Type	Select Content Based .
Routing Rules	To create a routing rule, click Add Rule and complete the Add Routing Rule dialog box.
Default To	Specify a Native Service endpoint to route the request to in case all routing rules evaluate to False. Click Endpoint , and select the URL of the Native Service to route the request to.

Alternatively, Mediator offers Local Optimization capability if the Native Service and the Virtual Service (in Mediator) are located on the same machine. With local optimization, service invocation happens in-memory and not through a network hop. In the **Default To** field, specify the Native Service in either of the following forms:

```
local://<service_full_path>
```

OR

```
local://<server>:<port>/ws/<service_full_path>
```

For example:

```
local://MediatorTestServices:NewMediatorTestServices_Port
```

which points to the endpoint service `NewMediatorTestServices_Port` which is present under the folder `MediatorTestServices` in Integration Server. This works for HTTP endpoints only, for all types of Routing Protocols.

Field	Description
HTTP Authentication	<p>Authentication Scheme: Specify the mode of authentication: Basic Authentication (default), NTLM, OAuth2, or None.</p> <p>Basic Authentication. Click one of the following options:</p> <ul style="list-style-type: none"> ■ Use credentials from incoming request: (default): Authenticates requests based on the credentials specified in the HTTP header. Mediator passes the Authorization header present in the original client request to the Native Service. ■ Use specified credentials: Authenticates requests according to the values you specify in the User, Password and Domain fields. <p>NTLM. Note that if Mediator is used to access a Native Service protected by NTLM (which is typically hosted in IIS), then the Native Service in IIS should be configured to use NTLM as the authentication scheme. If the authentication scheme is configured as Windows, then NTLM should be in its list. The Negotiate handshake will be supported in the near future. This note applies to all three of the following options for NTLM:</p> <ul style="list-style-type: none"> ■ Use credentials from incoming request: Default. Mediator uses the user credentials passed in the request header for an NTLM handshake with the server. ■ Use specified credentials: Mediator uses the values you specify in the User, Password and Domain fields for an NTLM handshake with the server. ■ Transparent: If the property <code>watt.pg.disableNtlmAuthHandler</code> is set to <code>false</code> (the default), then Mediator will behave in pass by mode, allowing an NTLM handshake to occur between the client and server. If the property <code>watt.pg.disableNtlmAuthHandler</code> is set to <code>true</code>, then Mediator performs the Kerberos Windows authentication (and not NTLM Windows authentication). This property is located in <code>Integration Server_directory\instances\instance_name\config\server.cnf</code>. Note: If the client is a WCF application, then the client should be configured with <code>clientCredentialType</code> set to <code>NTLM</code>. <p>OAuth2. Click one of the following options:</p> <ul style="list-style-type: none"> ■ Use credentials from incoming request: Default. This is known as pass through mode, in which the consumer includes an OAuth2 access token (a “Bearer” type token) in the request. Mediator then passes the access token unchanged to the native OAuth server. ■ Use specified token: In this mode, the consumer does not include an OAuth2 access token in the request. Instead, the provider generates an OAuth2 access token for each consumer, and Mediator stores the access tokens in Passman. When consumers send requests,

Field	Description
	<p>Mediator obtains the OAuth2 access tokens from Passman and uses them to access the Native Services. Specify an OAuth access token to be deployed by Mediator. If you select this option, the consumer need not pass the OAuth token during service invocation. Click Show Token to view the OAuth access token. Users who do not have the permissions to create and manage Virtual Services will not see this button.</p> <p>Specify an OAuth access token to be deployed by Mediator by clicking Show Token and selecting an OAuth access token. Users who do not have the permissions to create and manage Virtual Services will not see this button.</p> <div style="background-color: #f0f0f0; padding: 10px;"> <p>Note: Here are some general guidelines:</p> <ul style="list-style-type: none"> ■ You must set the Integration Server property <code>watt.server.auth.skipForMediator</code> to <code>true</code> and then restart Integration Server for the change to take effect. This property is located in the server configuration file (<code>server.cnf</code>), which is located in the <code>Integration Server_directory\config</code> directory. For details, see the <i>webMethods Integration Server Administrator's Guide</i>. ■ The run-time actions "Require HTTP Basic Authentication" and "Identify Consumer" (with the value HTTP Authentication Token as the identifier) will not be enforced when using the authentication scheme OAuth2. </div> <p>None. Select the following option:</p> <ul style="list-style-type: none"> ■ Invoke Service Anonymously: Does not authenticate requests.
HTTP Headers	<p>The HTTP headers that you want to use to authenticate the requests.</p> <ul style="list-style-type: none"> ■ Use Existing: Use the HTTP headers that are contained in the requests. ■ Customize: Use the HTTP headers that you specify in the Name and Value columns below. If you need to specify multiple headers, use the plus button to add rows.

Creating Content-based Routing Rule

> To create a content-based routing rule

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Virtual Service**.

3. In the **Assets** pane, right-click the Virtual Service you want to configure, and click **Details**.
The Virtual Service Details page is displayed.
4. In the **Processing Steps** tab, click **Routing Protocols**.
5. Click **Endpoint** (next to the **Route To** column).
6. In the **Search for Endpoint** dialog box, click **Search** to search for the Web service endpoint to route the requests to.
7. Select a Web Service and click **OK**.
8. Click **Configure Endpoint Properties** (next to the **Endpoint** button) to configure a set of properties for each endpoint individually. In the dialog box, click the endpoint you want to configure and specify an appropriate information for each of the displayed fields:

Field	Description
SOAP Optimization Method	<p><i>Optional.</i> Mediator can accept the following optimization methods to optimize the payloads of SOAP requests:</p> <ul style="list-style-type: none">■ MTOM: Indicates that Mediator expects to receive a request with a Message Transmission Optimization Mechanism (MTOM) attachment, and will forward the attachment to the Native Service.■ SwA: Indicates that Mediator expects to receive a “SOAP with Attachment” (SwA) request, and will forward the attachment to the Native Service.■ None (the default). <ol style="list-style-type: none">a. Bridging between SwA and MTOM is not supported. If a consumer sends an SwA request, Mediator can only forward SwA to the native provider. The same is true for MTOM, and applies to responses received from the native provider. That is, an SwA or MTOM response received by Mediator from a native provider will be forwarded to the caller using the same format it received.b. When sending SOAP requests that do <i>not</i> contain a MTOM or SWA attachment to a Virtual Service for a native provider endpoint that returns an MTOM or SWA response, the request Accept header must be set to <code>multipart/related</code> (or the Virtual Service's Request Processing Step should include an “Virtual Service” on page 1149 that sets the <code>BUILDER_TYPE</code> context variable to <code>multipart/related</code>). This is necessary so Mediator knows how to parse the response properly.

Field	Description
WSS Header Customization	<p data-bbox="505 254 1388 323">Indicates whether Mediator should pass the WS-Security headers of the incoming requests to the Native Service.</p> <ul style="list-style-type: none"> <li data-bbox="505 348 1388 457">■ Pass all security headers: Passes the security header, even if it is processed by Mediator (that is., even if Mediator processes the header according to the Virtual Service's security run-time policy). <div data-bbox="553 470 1365 674" style="background-color: #f0f0f0; padding: 5px;"> <p data-bbox="553 485 634 516">Note:</p> <p data-bbox="553 520 1357 653">If the Virtual Service does not contain a security run-time policy, and the <code>mustUnderstand</code> attribute of the security header is 0 or false, then Mediator will <i>always</i> forward the security header to the Native Service.</p> </div> <ul style="list-style-type: none"> <li data-bbox="505 688 1388 968">■ Remove processed security header from request before routing: Removes the security header if it is processed by Mediator (that is, if Mediator processes the header according to the Virtual Service's security run-time policy). Note that Mediator will <i>not</i> remove the security header if <i>both</i> of the following conditions are true: 1) Mediator did not process the security header, and 2) the <code>mustUnderstand</code> attribute of the security header is 0 or false).
HTTP Connection Timeout	<p data-bbox="505 993 1388 1203">The time interval (in seconds) after which a connection attempt will timeout. If a value is not specified (or if the value 0 is specified), Mediator uses the value of the global property <code>pg.endpoint.connectionTimeout</code> located in the file <code>Integration Server_directory/packages/WmMediator/config/resources/pg-config.properties</code>. The default of that property is 30 seconds.</p>
Read Timeout	<p data-bbox="505 1228 1388 1297">The time interval (in seconds) after which a socket read attempt will timeout.</p> <p data-bbox="505 1323 1325 1354">The precedence of the Read Timeout configuration is as follows:</p> <ol style="list-style-type: none"> <li data-bbox="505 1379 1388 1589">a. If a value is specified for the Read Timeout field in the routing endpoint alias, Mediator will use the value specified in the Runtime Alias > Endpoint Alias > Endpoint Properties > Read Timeout field. The read timeout value defined at an alias level takes precedence over the timeout values defined at an API level and the global configuration. <li data-bbox="505 1614 1388 1791">b. If a value 0 is specified (or if the value is not specified) for the Read Timeout field in the routing endpoint alias, then Mediator will use the value specified in the Read Timeout field of this step. The read timeout value defined at an API level takes precedence over the global configuration. <li data-bbox="505 1816 1388 1881">c. If a value 0 is specified (or if the value is not specified) for the Read Timeout field in this step (at an API level), then Mediator

Field	Description
	<p>will use the value of the global property <code>pg.endpoint.readTimeout</code> located in the file <code>Integration Server_directory\packages\WmMediator\config\resources\pg-config.properties</code> (in the Mediator Administration console, go to > Settings > Extended Settings > pg.endpoint.readTimeout property.).</p>

Note:

If a value for the Read Timeout configuration is not specified in any of the above configuration parameters, then Mediator will use the default 30 seconds.

SSL Options	<p>To enable SSL client authentication for the endpoint, you must specify values for both the Client Certificate Alias field and the IS Keystore Alias field. If you specify a value for only one of these fields, a deployment error will occur.</p>
--------------------	---

Note:

SSL client authentication is optional; you may leave both fields blank.

- **Client Certificate Alias:** The client's private key to be used for performing SSL client authentication. If you specify a client certificate alias, you must also include in the Virtual Service's policy the Require SSL action and select that action's Client Certificate Required option. The Client Certificate Required option specifies whether client certificates are required for the purposes of: 1) Verifying the signature of signed SOAP requests or decrypting encrypted SOAP requests, and 2) Signing SOAP responses or encrypting SOAP responses.
- **IS Keystore Alias:** The keystore alias of the instance of Integration Server on which Mediator is running. This value (along with the value of **Client Certificate Alias**) will be used for performing SSL client authentication.

To create an XPath expression for the routing rule, do the following:

1. Click the **Edit** button (next to the **If True** column).
2. In the XPath Editor that appears, view the **Namespace Map** tab, which displays all predefined namespaces (for example, `soapenv`, `soapenc`, and so on.). To add custom namespaces, click **Add Custom Namespace/prefix**, specify a name and value for the namespace and click **OK**. If you need to add additional rows, use the plus button.
3. In the XPath Editor's **All Nodes** tab, expand the namespace's node, choose the method you want for the XPath expression, and click **OK**.

4. In the XPath Editor's **XPATH Evaluator** tab, evaluate the XPath expression by specifying an argument in the **XPath Expression** field, and then click **Evaluate**.
5. After you have evaluated the XPath expression, click **OK**.

Configuring Context-based Routing

If your Entry Protocol is HTTP or HTTPS, you can choose to use the Context-based routing protocol. (Alternatively, you can choose Straight Through, Content-based, or Load Balancing routing.)

If you have a Native Service that is hosted at two or more endpoints, you can use the Context-Based protocol to route specific types of messages to specific endpoints.

The requests are routed according to the context-based routing rules you create. A routing rule specifies where requests should be routed, and the criteria by which they should be routed there. For example, requests can be routed according to certain consumers, certain dates/times, or according to requests that exceed/fall below a specified metric (Total Count, Success Count, Fault Count, and so on.). You can create one or more rules.

You may also specify how to authenticate requests (as with all routing protocols).

If a SOAP request contains a WS-Security header, Mediator passes it to the Native Service.

» To configure Context-based routing

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Virtual Service**.
3. In the **Assets** pane, right-click the Virtual Service you want to configure, and click **Details**.

The Virtual Service Details page is displayed.

4. In the **Processing Steps** tab, click **Routing Protocols**.
5. In the **Routing Protocols** tab, provide the required information for each of the displayed fields, and click **Save**:

Field	Description
HTTP or JMS	Select HTTP .
Routing Type	Select Context Based .
Routing Rules	To create a context-based routing rule, click Add Rule and complete the Add Routing Rule dialog box.
Route To	Specify a Native Service endpoint to route the request to in case all routing rules evaluate to False. Click Endpoint , and select the URL

Field	Description
	<p>of the Native Service to route the request to. To specify additional services, use the plus button next to the field to add rows.</p> <p>Alternatively, Mediator offers Local Optimization capability if the Native Service and the Virtual Service (in Mediator) are located on the same machine. With local optimization, service invocation happens in-memory and not through a network hop. In the Default To field, specify the Native Service in either of the following forms:</p> <pre>local://<service_full_path></pre> <p>OR</p> <pre>local://<server>:<port>/ws/<service_full_path></pre> <p>For example:</p> <pre>local://MediatorTestServices:NewMediatorTestServices_Port</pre> <p>which points to the endpoint service <code>NewMediatorTestServices_Port</code> which is present under the folder <code>MediatorTestServices</code> in Integration Server. This works for HTTP endpoints only, for all types of Routing Protocols.</p>
HTTP Authentication	<p>Authentication Scheme: Specify the mode of authentication: Basic Authentication (default), NTLM, OAuth2 or None.</p> <p>Basic Authentication. Select one of the following options:</p> <ul style="list-style-type: none"> ■ Use credentials from incoming request: (default): Authenticates requests based on the credentials specified in the HTTP header. Mediator passes the “Authorization” header present in the original client request to the Native Service. ■ Use specified credentials: Authenticates requests according to the values you specify in the User, Password and Domain fields. <p>NTLM. Note that if Mediator is used to access a Native Service protected by NTLM (which is typically hosted in IIS), then the Native Service in IIS should be configured to use NTLM as the authentication scheme. If the authentication scheme is configured as Windows, then NTLM should be in its list. The Negotiate handshake will be supported in the near future. This note applies to all three of the following options for NTLM:</p> <ul style="list-style-type: none"> ■ Use credentials from incoming request: Default. Mediator uses the user credentials passed in the request header for an NTLM handshake with the server.

Field	Description
	<ul style="list-style-type: none"> <li data-bbox="505 258 1388 359">■ Use specified credentials: Mediator uses the values you specify in the User, Password and Domain fields for an NTLM handshake with the server. <li data-bbox="505 386 1388 735">■ Transparent: If the property <code>watt.pg.disableNtlmAuthHandler</code> is set to <code>false</code> (the default), then Mediator will behave in pass by mode, allowing an NTLM handshake to occur between the client and server. If the property <code>watt.pg.disableNtlmAuthHandler</code> is set to <code>true</code>, then Mediator performs the Kerberos Windows authentication (and not NTLM Windows authentication). This property is located in <code>Integration Server_directory\instances\instance_name\config\server.cnf</code>. Note: If the client is a WCF application, then the client should be configured with <code>clientCredentialType</code> set to NTLM.

OAuth2. Select one of the following options:

- **Use credentials from incoming request:** Default. This is known as pass through mode, in which the consumer includes an OAuth2 access token (a Bearer type token) in the request. Mediator then passes the access token unchanged to the native OAuth server.
- **Use specified token:** In this mode, the consumer does not include an OAuth2 access token in the request. Instead, the provider generates an OAuth2 access token for each consumer, and Mediator stores the access tokens in Passman. When consumers send requests, Mediator obtains the OAuth2 access tokens from Passman and uses them to access the Native Services. Specify an OAuth access token to be deployed by Mediator. If you select this option, the consumer need not pass the OAuth token during service invocation. Click **Show Token** to view the OAuth access token. Users who do not have the permissions to create and manage Virtual Services will not see this button.

Specify an OAuth access token to be deployed by Mediator by clicking **Show Token** and selecting an OAuth access token. Users who do not have the permissions to create and manage Virtual Services will not see this button.

Note:

Keep the following in mind:

- You must set the Integration Server property `watt.server.auth.skipForMediator` to `true` and then restart Integration Server for the change to take effect. This property is located in the server configuration file (`server.cnf`), which is located in the `Integration Server_directory\config` directory. For details, see the *webMethods Integration Server Administrator's Guide*.

Field	Description
	<ul style="list-style-type: none"> The run-time actions Require HTTP Basic Authentication and Identify Consumer (with the value HTTP Authentication Token as the identifier) will not be enforced when using the authentication scheme OAuth2. <p>None. Select the following option:</p> <ul style="list-style-type: none"> Invoke Service Anonymously: Does not authenticate requests.
HTTP Headers	<p>The HTTP headers that you want to use to authenticate the requests.</p> <ul style="list-style-type: none"> Use Existing: Use the HTTP headers that are contained in the requests. Customize: Use the HTTP headers that you specify in the Name and Value columns below. If you need to specify multiple headers, use the plus button to add rows.

Creating Context-based Routing Rule

➤ To create a context-based routing rule

- In the **Variable** column, select **Time, IP Address, Date, Consumer, Predefined Context Variable** or **Custom Context Variable**. For more information, see [“Virtual Service” on page 1157](#).
- In the **Value** column, specify an applicable value. For **Date** select **Before, After** or **Equal To** and provide a date. For **Time** choose **Before** or **After** and provide a time. For **IP Address**, type numeric values for **Between** and **And**. For **Consumer**, click **Browse** and select a consumer application name. For **Predefined Context Variable** or **Custom Context Variable**, choose the **String** or **Integer** data type. Select a predefined variable name or custom variable name from the drop-down list. For **String**, choose **Equal To** or **Not Equal To** and type a value. For **Integer**, choose **Greater Than, Less Than, Not Equal To, Equal To** or and type a value.
 - For the list of the predefined context variables, see [“Virtual Service” on page 1157](#).
 - The predefined context variable `PROTOCOL_HEADER` is not available in the drop-down list; to include `PROTOCOL_HEADER` in the rule, define the variable as Custom Context Variable. For more information, see [“Context Variables in Virtual Services” on page 1157](#).
 - If you define a custom context variable in the routing rule, you must write a [“Virtual Service” on page 1149](#) and invoke it in the Virtual Service's Request Processing step. In this Integration Server service, use the API to get/set the custom context variable.
- In the **Combination Uses** field, choose an operator for the expression: **AND** (the default) or **OR**.

4. In the **Route To** field, click **Endpoint**, and then choose the URL of the Native Service to route the request to, if the rule criteria are met.
5. Click **Configure Endpoint Properties** (next to the **Endpoint** button) to configure a set of properties for each endpoint individually. In the dialog box, click the endpoint you want to configure and specify an appropriate information for each of the displayed fields:

Field	Description
SOAP Optimization Method	<p data-bbox="584 504 1385 577"><i>Optional.</i> Mediator can accept the following optimization methods to optimize the payloads of SOAP requests:</p> <ul style="list-style-type: none"> <li data-bbox="584 598 1385 745">■ MTOM: Indicates that Mediator expects to receive a request with a Message Transmission Optimization Mechanism (MTOM) attachment, and will forward the attachment to the Native Service. <li data-bbox="584 766 1385 871">■ SwA: Indicates that Mediator expects to receive a “SOAP with Attachment” (SwA) request, and will forward the attachment to the Native Service. <li data-bbox="584 892 1385 934">■ None (the default). <ul style="list-style-type: none"> <li data-bbox="584 955 1385 1207">a. Bridging between SwA and MTOM is not supported. If a consumer sends an SwA request, Mediator can only forward SwA to the native provider. The same is true for MTOM, and applies to responses received from the native provider. That is, an SwA or MTOM response received by Mediator from a native provider will be forwarded to the caller using the same format it received. <li data-bbox="584 1228 1385 1543">b. When sending SOAP requests that do <i>not</i> contain a MTOM or SWA attachment to a Virtual Service for a native provider endpoint that returns an MTOM or SWA response, the request Accept header must be set to <code>multipart/related</code> (or the Virtual Service's Request Processing Step should include an “Virtual Service” on page 1149 that sets the <code>BUILDER_TYPE</code> context variable to <code>multipart/related</code>). This is necessary so Mediator knows how to parse the response properly.
WSS Header Customization	<p data-bbox="584 1564 1385 1638">Indicates whether Mediator should pass the WS-Security headers of the incoming requests to the Native Service.</p> <ul style="list-style-type: none"> <li data-bbox="584 1659 1385 1799">■ Pass all security headers: Passes the security header, even if it is processed by Mediator (that is, even if Mediator processes the header according to the Virtual Service's security run-time policy).

Note:

Field	Description
	<p data-bbox="548 262 1258 399">If the Virtual Service does not contain a security run-time policy, and the <code>mustUnderstand</code> attribute of the security header is 0 or false, then Mediator will <i>always</i> forward the security header to the Native Service.</p> <ul style="list-style-type: none"> <li data-bbox="500 430 1289 745">■ Remove processed security header from request before routing: Removes the security header if it is processed by Mediator (that is, if Mediator processes the header according to the Virtual Service's security run-time policy). Note that Mediator will <i>not</i> remove the security header if <i>both</i> of the following conditions are true: 1) Mediator did not process the security header, and 2) the <code>mustUnderstand</code> attribute of the security header is 0 or false).
HTTP Connection Timeout	<p data-bbox="500 772 1289 1018">The time interval (in seconds) after which a connection attempt will timeout. If a value is not specified (or if the value 0 is specified), Mediator uses the value of the global property <code>pg.endpoint.connectionTimeout</code> located in the file <code>Integration Server_directory/packages/WmMediator/config/resources/pg-config.properties</code>. The default of that property is 30 seconds.</p>
Read Timeout	<p data-bbox="500 1045 1289 1108">The time interval (in seconds) after which a socket read attempt will timeout.</p> <p data-bbox="500 1134 1289 1165">The precedence of the Read Timeout configuration is as follows:</p> <ol style="list-style-type: none"> <li data-bbox="500 1192 1289 1438">a. If a value is specified for the Read Timeout field in the routing endpoint alias, Mediator will use the value specified in the Runtime Alias > Endpoint Alias > Endpoint Properties > Read Timeout field. The read timeout value defined at an alias level takes precedence over the timeout values defined at an API level and the global configuration. <li data-bbox="500 1465 1289 1638">b. If a value 0 is specified (or if the value is not specified) for the Read Timeout field in the routing endpoint alias, then Mediator will use the value specified in the Read Timeout field of this step. The read timeout value defined at an API level takes precedence over the global configuration. <li data-bbox="500 1665 1289 1869">c. If a value 0 is specified (or if the value is not specified) for the Read Timeout field in this step (at an API level), then Mediator will use the value of the global property <code>pg.endpoint.readTimeout</code> located in the file <code>Integration Server_directory/packages/WmMediator/config/resources/pg-config.properties</code> (in the Mediator

Field	Description
	<p>Administration console, go to > Settings > Extended Settings > pg.endpoint.readTimeout property.).</p> <p>Note: If a value for the Read Timeout configuration is not specified in any of the above configuration parameters, then Mediator will use the default 30 seconds.</p>
SSL Options	<p>To enable SSL client authentication for the endpoint, you must specify values for both the Client Certificate Alias field and the IS Keystore Alias field. If you specify a value for only one of these fields, a deployment error will occur.</p> <p>Note: SSL client authentication is optional; you may leave both fields blank.</p> <ul style="list-style-type: none"> ■ Client Certificate Alias: The client's private key to be used for performing SSL client authentication. If you specify a client certificate alias, you must also include in the Virtual Service's policy the Require SSL action and select that action's Client Certificate Required option. The Client Certificate Required option specifies whether client certificates are required for the purposes of: 1) Verifying the signature of signed SOAP requests or decrypting encrypted SOAP requests, and 2) Signing SOAP responses or encrypting SOAP responses. ■ IS Keystore Alias: The keystore alias of the instance of Integration Server on which Mediator is running. This value (along with the value of Client Certificate Alias) will be used for performing SSL client authentication.

6. Click **OK**.

Configuring Load Balancing Routing

If your Entry Protocol is HTTP or HTTPS, you can choose to use the Load Balancing routing protocol. (Alternatively, you can choose Straight Through, Content-Based, or Context-Based routing.)

If you have a Web service that is hosted at two or more endpoints, you can use the Load Balancing option to distribute requests among the endpoints.

Requests are distributed across multiple endpoints. The requests are intelligently routed based on the Round-Robin execution strategy. The load for a service is balanced by directing requests to two or more services in a pool, until the optimum level is achieved. The application routes

requests to services in the pool sequentially, starting from the first to the last service without considering the individual performance of the services. After the requests have been forwarded to all the services in the pool, the first service is chosen for the next loop of forwarding.

Load-balanced endpoints also have automatic Failover capability. If a load-balanced endpoint is unavailable (for example, if a connection is refused), then that endpoint is marked as Down for the number of seconds you specify in the **Suspend the Failed Endpoint** field (during which the endpoint will not be used for sending the request), and the next configured endpoint is tried. If all the configured load-balanced endpoints are down, then a SOAP fault is sent back to the client. After the suspension period expires, each endpoint marked will be available again to send the request.

> To configure Load Balancing routing

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Virtual Service**.
3. In the **Assets** pane, right-click the Virtual Service you want to configure, and click **Details**.
The Virtual Service Details page is displayed.
4. In the **Processing Steps** tab, click **Routing Protocols**.
5. In the **Routing Protocols** tab, provide the required information for each of the displayed fields, and click **Save**:

Field	Description
HTTP or JMS	Select HTTP .
Routing Type	Select Load Balancing .
Route To	The URLs of two or more Native Services in a pool to which the requests will be routed. The application routes the requests to services in the pool sequentially, starting from the first to the last service without considering the individual performance of the services. After the requests have been forwarded to all the services in the pool, the first service is chosen for the next loop of forwarding.

To specify the first service, click **Endpoint** and select the URL of the Web service to route the request to. To specify additional services, use the plus button next to the field to add rows.

Alternatively, Mediator offers Local Optimization capability if the Native Service and the Virtual Service (in Mediator) are located on the same machine. With local optimization, service invocation happens in-memory and not through a network hop. In the **Default To** field, specify the Native Service in either of the following forms:

Field	Description
	<p><code>local://<service_full_path></code></p> <p>OR</p> <p><code>local://<server>:<port>/ws/<service_full_path></code></p> <p>For example:</p> <pre>local://MediatorTestServices:NewMediatorTestServices_Port</pre> <p>which points to the endpoint service <code>NewMediatorTestServices_Port</code> which is present under the folder <code>MediatorTestServices</code> in Integration Server. This works for HTTP endpoints only, for all types of Routing Protocols.</p>
<p>Configure Endpoint Properties (icon)</p>	<p>The Configure Endpoint Properties icon displays a dialog box that enables you to configure a single set of properties that will be shared by all the endpoints. In the dialog box, specify the following fields:</p> <ul style="list-style-type: none"> ■ SOAP Optimization Method: Optional. Mediator can accept the following optimization methods to optimize the payloads of SOAP requests: <ul style="list-style-type: none"> ■ MTOM: Indicates that Mediator expects to receive a request with a Message Transmission Optimization Mechanism (MTOM) attachment, and will forward the attachment to the Native Service. ■ SwA: Indicates that Mediator expects to receive a “SOAP with Attachment” (SwA) request, and will forward the attachment to the Native Service. ■ None (the default). <ol style="list-style-type: none"> 1. Bridging between SwA and MTOM is not supported. If a consumer sends an SwA request, Mediator can only forward SwA to the native provider. The same is true for MTOM, and applies to responses received from the native provider. That is, an SwA or MTOM response received by Mediator from a native provider will be forwarded to the caller using the same format it received. 2. When sending SOAP requests that do <i>not</i> contain a MTOM or SWA attachment to a Virtual Service for a native provider endpoint that returns an MTOM or SWA response, the request Accept header must be set to <code>multipart/related</code> (or the Virtual Service's Request Processing Step should include an “Virtual Service” on page 1149 that sets the <code>BUILDER_TYPE</code> context variable to <code>multipart/related</code>). This is necessary so Mediator knows how to parse the response properly.

Field	Description
	<ul style="list-style-type: none"> <li data-bbox="373 252 1289 357">■ WSS Header Customization: Indicates whether Mediator should pass the WS-Security headers of the incoming requests to the Native Service. <li data-bbox="373 378 1289 525">■ Pass all security headers: Passes the security header, even if it is processed by Mediator (that is, even if Mediator processes the header according to the Virtual Service's security run-time policy). <div data-bbox="470 546 1266 745" style="background-color: #f0f0f0; padding: 5px; margin: 10px 0;"> <p>Note: If the Virtual Service does not contain a security run-time policy, and the <code>mustUnderstand</code> attribute of the security header is 0 or false, then Mediator will <i>always</i> forward the security header to the Native Service.</p> </div> <ul style="list-style-type: none"> <li data-bbox="373 756 1289 1039">■ Remove processed security header from request before routing: Removes the security header if it is processed by Mediator (that is, if Mediator processes the header according to the Virtual Service's security run-time policy). Note that Mediator will <i>not</i> remove the security header if <i>both</i> of the following conditions are true: 1) Mediator did not process the security header, and 2) the <code>mustUnderstand</code> attribute of the security header is 0 or false). <li data-bbox="373 1060 1289 1312">■ HTTP Connection Timeout: The time interval (in seconds) after which a connection attempt will timeout. If a value is not specified (or if the value 0 is specified), Mediator uses the value of the global property <code>pg.endpoint.connectionTimeout</code> located in the file <code>Integration Server_directory\packages\WmMediator\config\resources\pg-config.properties</code>. The default of that property is 30 seconds. <li data-bbox="373 1333 1289 1402">■ Read Timeout: The time interval (in seconds) after which a socket read attempt will timeout.

The precedence of the Read Timeout configuration is as follows:

1. If a value is specified for the Read Timeout field in the routing endpoint alias, Mediator will use the value specified in the **Runtime Alias > Endpoint Alias > Endpoint Properties > Read Timeout** field. The read timeout value defined at an alias level takes precedence over the timeout values defined at an API level and the global configuration.
2. If a value 0 is specified (or if the value is not specified) for the Read Timeout field in the routing endpoint alias, then Mediator will use the value specified in the Read Timeout field of this step.

Field	Description
	<p>The read timeout value defined at an API level takes precedence over the global configuration.</p> <p>3. If a value 0 is specified (or if the value is not specified) for the Read Timeout field in this step (at an API level), then Mediator will use the value of the global property <code>pg.endpoint.readTimeout</code> located in the file <i>Integration Server_directory/packages/WmMediator/config/resources/pg-config.properties</i> (in the Mediator Administration console, go to > Settings > Extended Settings > pg.endpoint.readTimeout property.).</p> <p>Note: If a value for the Read Timeout configuration is not specified in any of the above configuration parameters, then Mediator will use the default 30 seconds.</p> <ul style="list-style-type: none"> ■ SSL Options: To enable SSL client authentication for the endpoint, you must specify values for both the Client Certificate Alias field and the IS Keystore Alias field. If you specify a value for only one of these fields, a deployment error will occur. <p>Note: SSL client authentication is optional; you may leave both fields blank.</p> <ul style="list-style-type: none"> ■ Client Certificate Alias: The client's private key to be used for performing SSL client authentication. If you specify a client certificate alias, you must also include in the Virtual Service's policy the Require SSL action and select that action's Client Certificate Required option. The Client Certificate Required option specifies whether client certificates are required for the purposes of: 1) Verifying the signature of signed SOAP requests or decrypting encrypted SOAP requests, and 2) Signing SOAP responses or encrypting SOAP responses. ■ IS Keystore Alias: The keystore alias of the instance of Integration Server on which Mediator is running. This value (along with the value of Client Certificate Alias) will be used for performing SSL client authentication.
Suspend the Failed Endpoint	<p>A numeric timeout value (in seconds).</p> <p>Default: 30. When this timeout value expires, the system routes the execution of the Virtual Service to the next consecutive Web service endpoint specified in the Route To field.</p>

Field	Description
HTTP Authentication	<p data-bbox="375 260 1256 323">Authentication Scheme: Specify the mode of authentication: Basic Authentication (default), NTLM, OAuth2 or None.</p> <p data-bbox="375 354 1133 386">Basic Authentication. Select one of the following options:</p> <ul data-bbox="375 417 1289 646" style="list-style-type: none"> <li data-bbox="375 417 1289 554">■ Use credentials from incoming request: (default): Authenticates requests based on the credentials specified in the HTTP header. Mediator passes the Authorization header present in the original client request to the Native Service. <li data-bbox="375 579 1289 646">■ Use specified credentials: Authenticates requests according to the values you specify in the User, Password and Domain fields. <p data-bbox="375 678 1289 877">NTLM. Note that if Mediator is used to access a Native Service protected by NTLM (which is typically hosted in IIS), then the Native Service in IIS should be configured to use NTLM as the authentication scheme. If the authentication scheme is configured as Windows, then NTLM should be in its list. The Negotiate handshake will be supported in the near future. This note applies to all three of the following options for NTLM:</p> <ul data-bbox="375 909 1289 1478" style="list-style-type: none"> <li data-bbox="375 909 1289 1010">■ Use credentials from incoming request: Default. Mediator uses the user credentials passed in the request header for an NTLM handshake with the server. <li data-bbox="375 1041 1289 1142">■ Use specified credentials: Mediator uses the values you specify in the User, Password and Domain fields for an NTLM handshake with the server. <li data-bbox="375 1173 1289 1478">■ Transparent: If the property <code>watt.pg.disableNtlmAuthHandler</code> is set to <code>false</code> (the default), then Mediator will behave in pass by mode, allowing an NTLM handshake to occur between the client and server. If the property <code>watt.pg.disableNtlmAuthHandler</code> is set to <code>true</code>, then Mediator performs the Kerberos Windows authentication (and not NTLM Windows authentication). This property is located in <code>Integration Server_directory\instances\instance_name\config\server.cnf</code>. Note: If the client is a WCF application, then the client should be configured with <code>clientCredentialType</code> set to <code>NTLM</code>. <p data-bbox="375 1509 938 1541">OAuth2. Click one of the following options:</p> <ul data-bbox="375 1572 1289 1866" style="list-style-type: none"> <li data-bbox="375 1572 1289 1709">■ Use credentials from incoming request: Default. This is known as pass through mode, in which the consumer includes an OAuth2 access token (a Bearer type token) in the request. Mediator then passes the access token unchanged to the native OAuth server. <li data-bbox="375 1734 1289 1866">■ Use specified token: In this mode, the consumer does not include an OAuth2 access token in the request. Instead, the provider generates an OAuth2 access token for each consumer, and Mediator stores the access tokens in Passman. When consumers send requests,

Field	Description
	<p>Mediator obtains the OAuth2 access tokens from Passman and uses them to access the Native Services. Specify an OAuth access token to be deployed by Mediator. If you select this option, the consumer need not pass the OAuth token during service invocation. Click Show Token to view the OAuth access token. Users who do not have the permissions to create and manage Virtual Services will not see this button.</p> <p>Specify an OAuth access token to be deployed by Mediator by clicking Show Token and selecting an OAuth access token. Users who do not have the permissions to create and manage Virtual Services will not see this button.</p> <div style="background-color: #f0f0f0; padding: 10px;"> <p>Note: Here are some general guidelines:</p> <ul style="list-style-type: none"> ■ You must set the Integration Server property <code>watt.server.auth.skipForMediator</code> to <code>true</code> and then restart Integration Server for the change to take effect. This property is located in the server configuration file (<code>server.cnf</code>), which is located in the <code>Integration Server_directory\config</code> directory. For details, see the <i>webMethods Integration Server Administrator's Guide</i>. ■ The run-time actions "Require HTTP Basic Authentication" and "Identify Consumer" (with the value HTTP Authentication Token as the identifier) will not be enforced when using the authentication scheme OAuth2. </div> <p>None. Select the following option:</p> <ul style="list-style-type: none"> ■ Invoke Service Anonymously: Does not authenticate requests.
HTTP Headers	<p>The HTTP headers that you want to use to authenticate the requests.</p> <ul style="list-style-type: none"> ■ Use Existing: Use the HTTP headers that are contained in the requests. ■ Customize: Use the HTTP headers that you specify in the Name and Value columns below. If you need to specify multiple headers, use the plus button to add rows.

Configuring Routing Protocol for Web Services Exposed over JMS

You can use the Routing Protocols step to specify a JMS queue to which the Mediator is to submit the request, and the destination to which the Native Service is to return the response.

➤ **To configure the Routing Protocols step for Web Services exposed over JMS**

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Virtual Service**.
3. In the **Assets** pane, right-click the Virtual Service you want to configure, and click **Details**.
The Virtual Service Details page is displayed.
4. In the **Processing Steps** tab, click **Routing Protocols**.
5. Click the **Routing Protocols** tab. Select **JMS**. Specify an appropriate information for each of the displayed fields, and then click **Save**:

Note:

For reasons of legibility some of the examples below contain break lines and may not work when pasted into applications or command line tools.

Field	Description
JMS URI	<p>A connection alias for connecting to the JMS provider (for example, an Integration Server alias or a JNDI URL). For example, a JNDI URL of the form:</p> <pre style="background-color: #f0f0f0; padding: 5px;">jms:queue:dynamicQueues/MyRequestQueue? wm-wsendpointalias=MediatorConsumer &targetService=vs-jms-in-echo</pre> <p>Note that the <code>wm-wsendpointalias</code> parameter is required for Integration Server/Mediator to look up the JMS consumer alias to send the request to the specified queue (for example, <code>MyRequestQueue</code>), which is a dynamic queue in ActiveMQ. Also, the <code>targetService</code> parameter is required if sending to another Virtual Service that uses JMS as the entry protocol.</p>
Priority	Optional. A numeric value that specifies the priority of the JMS message in the queue.
Reply to Destination	Optional. A queue name for the incoming JMS request.
Time to Live	Optional. A numeric value (in milliseconds) that specifies the lifespan of the JMS message.
Delivery Mode	Optional. The type of message delivery to the endpoint. <ul style="list-style-type: none"> ■ None (default). ■ Persistent: The message is stored by the JMS server before delivering it to the consumer. ■ Non-Persistent: The message is not stored before delivery.

Field	Description
Message Properties	The message properties to use. <ul style="list-style-type: none"> ■ Use Existing (default): Use existing properties. ■ Customize: Specify one or more property names and values. To add additional rows, use the plus button.
JMS Headers	The JMS headers that you want to use to authenticate the requests. <ul style="list-style-type: none"> ■ Use Existing (default): Use existing headers. ■ Customize: Specify one or more header names and values. To add additional rows, use the plus button.

Deploying Virtual Services to Targets

To deploy Virtual Services to targets, you must belong to a role with the following permissions:

- Create Assets —OR— Manage Assets
- Manage Runtime Policies (required to configure Virtual Services)
- Manage Lifecycle Models (required to change state of Virtual Services)
- Mediator Publisher (required to deploy Virtual Services)

Note: Mediator exposes several Web service operations to allow CentraSite to manage deployed assets. This Web service is invoked by CentraSite any time a user deploys or undeploys a Virtual Service to Mediator. A Mediator's configuration details page includes the `User Name` and `Password` fields which identify an Integration Server user who is permitted to execute the Integration Server services associated with Mediator's deployer service. After installation, only members of the Integration Server Administrators group are allowed to invoke these services. However, administrators have the flexibility to allow their own users or groups to invoke them. Access to these services is controlled by an ACL, called `MediatorDeployer`. Initially, only the predefined Administrators group is assigned to this ACL. An Integration Server administrator can remove this group and add other groups or individual users. For example, you can create your own deployer group, (for example, `MyDeployers`) and add Integration Server user IDs to this group. Then, the user must update the `MediatorDeployer` ACL by removing the Administrators group and adding the `MyDeployers` group. Now, on the target's configuration detail page, you can specify any user ID that belongs to `MyDeployers` group.

You cannot deploy a Virtual Service in CentraSite Control until you meet the following conditions:

- Ensure that you have the Mediator Publisher role. Only users with this permission can deploy a Virtual Service. CentraSite will not enable the deployment controls for any other users.
- Ensure that the run-time policies for the Virtual Service are active. This is indicated in the **Policies** profile on the Virtual Service's detail page. If a policy is inactive, you must activate it.

- Ensure that the gateway to which the Virtual Service will be deployed has already been created.
- Ensure that the gateway's specified deployment URL is active and the user credentials of Integration Server are valid. To check this, go to the gateway's detail page and click the **Publish** button. If the connection is not active and valid, activate the deployment endpoint and modify the user credentials as required.
- If the Virtual Service is under the control of an active lifecycle model (LCM), ensure that:
 - The Virtual Service is in a deployable lifecycle state. If you are not certain of what the deployable lifecycle state is, consult your CentraSite administrator.
 - The Virtual Service has a design-time policy that includes the Change Deployment Status action and it is set to Yes. This action specifies whether the service is eligible for deployment.

If these conditions are not satisfied, all or part of the deployment user interface controls will be disabled when you view the Virtual Service.

If you Migrate Virtual Services from a Pre-8.2 Release

If you have Virtual Services that were created prior to version 8.2, those Virtual Services continue to hold the deployment metadata generated by CentraSite 8.0. Although this metadata is not applicable in CentraSite 8.2, and will not affect deployment in 8.2, we strongly recommend that you perform the following steps:

1. Undeploy all Virtual Services from CentraSite 8.0.
2. Upgrade to CentraSite 8.2.
3. Ensure that all gateway endpoints are configured correctly.
4. Deploy all Virtual Services from CentraSite 8.2.

Note:

Please be aware that the new synchronous deployment model does not support subscriptions and subscription services.

Deploying Virtual Services from the Details Page

To deploy a Virtual Service asset to targets, ensure that:

- All of the run-time policies that are available for the Virtual Service are Active.
- The Virtual Service has a design-time policy configured with the **Change Deployment Status** action and it is set to **Yes**. This action specifies whether the service is eligible for deployment. For more information, see the description of this action in the *CentraSite Developer's Guide*.

The following procedure describes how to deploy, undeploy, and redeploy a single Virtual Service to one or more Mediator targets, using the Virtual Service's **Deployment** profile.

➤ To deploy, undeploy, or redeploy Virtual Service from the details page

1. In CentraSite Control, go to **Asset Catalog > Browse**.

2. In the displayed list of asset types, select **Virtual Service**.
3. In the **Assets** pane, right-click the Virtual Service you want to deploy, and click **Details**.

The Virtual Service Details page is displayed.

4. Click the **Policies** tab.
5. In the **Policies** tab, switch the Virtual Service to an active, ready-to-deploy state, as follows. (If you do not know which state to select, you will need to examine your organization's lifecycle model for the Virtual Service or consult an administrator.)
 - a. On the **Actions** menu, click **Change Lifecycle State**.

A list of lifecycle states you are permitted to assign to the Virtual Service is displayed.

- b. Select the lifecycle state to which you want to switch the asset, and then click **OK**.

If the state change requires approval, CentraSite will initiate an approval workflow, and your request for a state change will be submitted to the appropriate approvers. While the request is awaiting approval, the Virtual Service will appear in a pending state.

After the request is approved by designated approvers, asset is switched to the selected lifecycle state.

6. Click the **Deployment** tab.

The **Deployment** tab displays the following information for the Virtual Service:

Column	Description
Target	The target on which the Virtual Service is deployed.
Target Type	The type of target on which the Virtual Service is deployed (for example, webMethods Integration Server or webMethods Insight).
Description	The description for the target.
Deployment Status	The deployment status of the Virtual Service (for example, Deployed or Failed).
Date Deployed	The date/time that the deployment occurred.

7. In the **Deployment** tab, click the **Deploy** button.
8. In the **Deploy to Targets** dialog box, select one or more Mediator targets to which you want to deploy the Virtual Service, and then click **OK**.

The deployment process is carried out by a synchronous mechanism between CentraSite and Mediator:

- a. CentraSite pushes the Virtual Service that is ready for deployment to the Mediator.
- b. Instantly, the Mediator deploys the Virtual Service that was received from CentraSite (along with its effective run-time policy), and notifies CentraSite when the deployment process is complete.

Important:

If the status shown in the **Deployment Status** column does not automatically switch to **Deployed**, click the **Refresh** button to determine whether CentraSite was able to deploy the Virtual Service successfully. If the deployment process failed, identify and correct the error and then try deploying the Virtual Service again.

9. To undeploy the Virtual Service, select one or more targets, and on the **Actions** menu, click **Undeploy**.

If the undeployment is successful, Mediator's deployer service returns a success message, and data that is pertinent to the undeployed Virtual Service. In addition, CentraSite's Deployment Manager logs information about the undeployment in the Deployment log. If the undeployment is unsuccessful, the deployer service returns a failure message.

Important:

If the status shown in the **Deployment Status** column does not automatically switch to **Undeployed**, click the **Refresh** button to determine whether CentraSite was able to undeploy the Virtual Services successfully. If the undeployment process failed, identify, and correct the error and then try undeploying the Virtual Service again.

10. To redeploy the Virtual Service, select one or more targets, and on the **Actions** menu, click **Redeploy**.

Important:

If the status shown in the **Deployment Status** column does not automatically switch to **Deployed**, click the **Refresh** button to determine whether CentraSite was able to redeploy the Virtual Services successfully. If the redeployment process failed, identify and correct the error and then try redeploying the Virtual Service again.

11. In the **Deployment Log**, check for any errors that occurred during the deployment process.

Deploying Virtual Services from the Deployment Page

The following procedure describes how to deploy, undeploy, and redeploy Virtual Services to Mediator target.

> To deploy, undeploy, or redeploy a Virtual Service

1. In CentraSite Control, go to **Operations > Deployment**.

- In the **Deployed Assets** tab, click **Deploy**.

The **Deployed Assets** tab displays the following information for a Virtual Service:

Column	Description								
Pending Changes	Indicates the deployment status of the Virtual Service.								
	<table> <thead> <tr> <th>Icon</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>The service is deployed to the target.</td> </tr> <tr> <td></td> <td>The service is pending deployment to the target.</td> </tr> </tbody> </table>	Icon	Description		The service is deployed to the target.		The service is pending deployment to the target.		
Icon	Description								
	The service is deployed to the target.								
	The service is pending deployment to the target.								
Deployment ID	The deployment ID of the Virtual Service.								
Name	Name of the Virtual Service.								
Status	The deployment status of the Virtual Service (Deployed/Failed).								
Date/Time	The date/time that the deployment occurred.								
User ID	The ID of the Integration Server user to be used for deployment operation.								
Type	The modification that is performed on a deployed Virtual Service.								
	<table> <thead> <tr> <th>Label</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Processing Step Changes</td> <td>Reflects any change that is performed in the Processing Steps tab of a Virtual Service. For example, modifying the HTTP authentication mode.</td> </tr> <tr> <td>Runtime Policy Changes</td> <td>Reflects any change that is performed in the runtime policy associated to a Virtual Service. For example, deactivating an associated runtime policy.</td> </tr> <tr> <td>WSDL Changes</td> <td>Reflects any change that is performed in the input WSDL file of a Virtual Service. For example, modifying an existing asset file or uploading a new asset file.</td> </tr> </tbody> </table>	Label	Description	Processing Step Changes	Reflects any change that is performed in the Processing Steps tab of a Virtual Service. For example, modifying the HTTP authentication mode.	Runtime Policy Changes	Reflects any change that is performed in the runtime policy associated to a Virtual Service. For example, deactivating an associated runtime policy.	WSDL Changes	Reflects any change that is performed in the input WSDL file of a Virtual Service. For example, modifying an existing asset file or uploading a new asset file.
Label	Description								
Processing Step Changes	Reflects any change that is performed in the Processing Steps tab of a Virtual Service. For example, modifying the HTTP authentication mode.								
Runtime Policy Changes	Reflects any change that is performed in the runtime policy associated to a Virtual Service. For example, deactivating an associated runtime policy.								
WSDL Changes	Reflects any change that is performed in the input WSDL file of a Virtual Service. For example, modifying an existing asset file or uploading a new asset file.								

- In the **Select a Target and Services to be Deployed on the Selected Target** dialog box, do a keyword search or an advanced search, as described later in this topic, for the list of available Virtual Services, and then click **OK**.

A list of Virtual Services that are ready for deployment is displayed.

The deployment process is carried out by a synchronous mechanism between CentraSite and Mediator:

- CentraSite pushes the Virtual Service that is ready for deployment to Mediator.

- b. Instantly, the Mediator deploys the Virtual Service that was received from CentraSite (along with its effective run-time policy), and notifies CentraSite when the deployment process is complete.

Important:

If the status shown in the **Deployment Status** column does not automatically switch to **Deployed**, click the **Refresh** button to determine whether CentraSite was able to deploy the Virtual Service successfully. If the deployment process failed, identify and correct the error and then try deploying the Virtual Service again.

4. To undeploy the Virtual Service, select one or more targets, and on the **Actions** menu, click **Undeploy**.

If the undeployment is successful, Mediator's deployer service returns a success message, and data that is pertinent to the undeployed Virtual Service. In addition, CentraSite's Deployment Manager logs information about the undeployment in the Deployment log. If the undeployment is unsuccessful, the deployer service returns a failure message.

Important:

If the status shown in the **Deployment Status** column does not automatically switch to **Undeployed**, click the **Refresh** button to determine whether CentraSite was able to undeploy the Virtual Services successfully. If the undeployment process failed, identify, and correct the error and then try undeploying the Virtual Service again.

5. To redeploy the Virtual Service, select one or more targets, and on the **Actions** menu, click **Redeploy**.

Important:

If the status shown in the **Deployment Status** column does not automatically switch to **Deployed**, click the **Refresh** button to determine whether CentraSite was able to redeploy the Virtual Services successfully. If the redeployment process failed, identify and correct the error and then try redeploying the Virtual Service again.

6. In the **Deployment Log**, check for any errors that occurred during the deployment process.

Performing Keyword Search for Deployment

The keyword search is an easy to use search facility in which you can specify arbitrary search patterns.

You can search for all Virtual Services that contain one or more specified keywords (text strings) in the Virtual Service's string attributes (Name, Description).

Here are some general guidelines:

- A keyword search consists of 1-n search keywords. Multiple keywords are space separated. If multiple keywords are given, a logical disjunction (OR) is implied.
- A keyword is treated as partial text which can occur at the beginning of the searched strings. The starts with semantics are implied.

Example: If the keyword is `AustralianPostCode`, then the following matches are returned: `A sample VS for AustralianPostCode` as well as `AustralianPostCodeVService`.

- If quotes (" ") exist around a phrase, then a search is performed on the exact phrase within the quotes. A space within a quoted phrase is considered as a space character and not as a logical operation. To force the keyword search to treat the quote characters as a normal character, precede the quote character with a backslash (\). If you want to include the backslash character itself in the search, type two backslashes.
- You can mix and match any number of words and quoted phrases within the keyword field.
- The search is neither case nor accent sensitive, even within a quoted phrase. Example: A search for `AustralianPostCode` will return the same results as a search for `AUSTRALIANPOSTCODE` or `Australianpostcode`.
- If you enter a string that contains an odd number of double-quote characters, then the last double-quote character is ignored when the search is performed.
- If the keyword search input field is empty when the search is executed, the search returns all available Virtual Services.
- The keyword search can include wildcard characters.

Wildcard Characters

The available wildcard characters are:

Character	Usage
* or %	If you use the percent symbol (%) or the asterisk (*), CentraSite replaces the wildcard symbol with as many characters as necessary to find a match. For example, an entry of <code>A%n</code> returns both <code>Amazon</code> and <code>American</code> . If you enter <code>*al</code> , then <code>CalcService</code> , <code>Calendar</code> and <code>AustralianPostCode</code> all fit the search criteria.
? or _	If you use the question mark (?) or the underscore (_), CentraSite replaces the wildcard symbol with a single character in order to find a match. Example: <code>AustralianPostCode?VService</code> matches any character for <code>?</code> .

You can use a wildcard character at any point in the keyword text, and multiple times throughout the keyword text. If you enter a wildcard character in the middle of a string, for example `cat*dog`, then at least one of the searched attributes must contain the string in order for the asset or supporting document to be included in the result set.

If a wildcard character between two words is surrounded by spaces, such as `word1 * word2`, the wildcard will match one word.

Note:

Here are some general guidelines when using wildcard characters:

- Certain non-alphanumeric characters that can appear in the name of a Virtual Service are currently ignored by CentraSite's wildcard mechanism when you include them in a keyword search. In particular, the hyphen (-) is ignored. Thus, if you have created the Virtual Services

AustralianPostCodeVService-1 and AustralianPostCodeVService_1, the wildcard search for AustralianPostCodeVService?1 will find AustralianPostCodeVService_1 but not AustralianPostCodeVService-1.

- The percent (%) character acts as a word delimiter when it appears in the text to be searched. Thus, for example, if the description field of a Virtual Service contains the text abc%def (the characters a, b, c, %, d, e, f), this is treated by the search mechanism as two adjacent words abc and def. A wildcard search such as abc*def looks for a single word beginning with abc and ending with def, so the search will not find this asset.

➤ To search using keyword

1. In CentraSite Control, go to **Operations > Deployment**.
2. In the **Deployed Assets** tab, click **Deploy**.

This opens up the **Select a Target and Services to be Deployed on the Selected Target** dialog box.

3. Type the keywords to search for Virtual Services in the text box. You can use one or more wildcards to specify the keywords.

If you leave the text box blank, or enter just a wildcard, the entire set of Virtual Services is returned.

CentraSite returns the Virtual Services that match the search criteria. The search looks for the keywords in the Name and Description attributes of a Virtual Service.

Performing Advanced Search for Deployment

CentraSite's advanced search capabilities allow you to search for Virtual Services on the basis of asset types and targets.

➤ To search using type and target

1. In CentraSite Control, go to **Operations > Deployment**.
2. In the **Deployed Assets** tab, click **Deploy**.
3. In the **Select a Target and Services to be Deployed on the Selected Target** dialog box, do the following:
 - a. In the field labeled **Browse By**, select the asset type, **Virtual Service**. A list of Virtual Services is displayed.
 - If you do not specify the asset type, **Virtual Service**, in the field labeled **Browse by**, CentraSite Control displays a list of Virtual Services, Virtual REST Services, Virtual OData Services, and Virtual XML Services available in the CentraSite registry.

There are several generic entries in the drop-down list for the **Browse by** field. These are:

- **[All]**

A list of all Virtual Services that are available in a deployable state.

- **[Virtual Service]**

A list of all virtual Web services that are available in a deployable state.

- **[Virtual REST Service]**

A list of all Virtual REST Services that are available in a deployable state.

- **[Virtual OData Service]**

A list of all Virtual OData Services that are available in a deployable state.

- **[Virtual XML Service]**

A list of all Virtual XML Services that are available in a deployable state.

- If you specify an asset type in the field labeled **Browse by**, CentraSite Control displays a list of all Virtual Services of the specified asset type.
 - b. In the **Target** list, select a target for deploying the Virtual Services.
 - c. Select one or more Virtual Services you want to deploy on the selected target.
4. Click **OK**.

Deploying Virtual Services Using Batch Process

Use `Runtime.deployment.Deployer` when you do not have access to a browser or graphical user interface environment, and you want to perform deployment tasks. You can also use `Runtime.deployment.Deployer` when you want to automate deployment tasks through batch processes.

An automated deployment through a batch mode can be initiated by configuring the `DeploymentConfiguration.properties` file located in the URL `http://<host>:53307/CentraSite/CentraSite/ino:dav/ino:dav/projects/CentraSite/configuration`.

Specifying a Deploy Batch Size

`BatchSize` is the maximum number of Virtual Services to be pushed to the Mediator before a syncpoint is taken. The default `BatchSize` is 50. To improve performance, you can set a `BatchSize` to define the maximum number of Virtual Services to be pushed between two syncpoints using the property line:

```
com.softwareag.centrasite.runtime.deployment.DeployBatchSize=50
```

The `BatchSize` property can be set at any time. If a bulk deployment is already in progress, the current batch is sized according to the previous batch size. Subsequent batches use the new size.

Suppose if the BatchSize is set to zero and changed while a deploy operation is already in progress, that operation loads the data as a single batch. Any subsequent deploy operations on the same CentraSite Control use the new BatchSize.

Specifying a Transaction Timeout

TransactionTimeout specifies the maximum time, in milliseconds, allowed for deployment operations (deployment or undeployment) that were pushed to Mediator to respond. Any such operations that do not respond before this timeout occurs are rolled back. The default TransactionTimeout is 6000 (ms). To improve performance, you can set a TransactionTimeout to define the maximum time for the deployment operations to respond using the property line:

```
com.softwareag.centrasite.runtime.deployment.TransactionTimeout=60000
```

For example, if a deployment operation attempts to set a transaction timeout of 360 seconds, and the TransactionTimeout setting is 300 seconds, the TransactionTimeout setting of 300 seconds is used. After the TransactionTimeout of 300 seconds the deployment operations roll back.

Note:

If set to 0, the transaction will not time out.

Deleting Activity Logs Through Deployed Assets Tab

To view the Deployment History Log, you must have the CentraSite Administrator, Organization Administrator, or Operations Administrator role.

Follow these general guidelines while deleting deployment activity logs:

- Deleting an activity log through the **Deployment History** profile will not affect the deployment status of a Virtual Service. However, deleting an activity log through the **Deployed Assets** tab will undeploy the Virtual Service from the Mediator target, and will stich the deployment status to **Undeployed**.
- The **Deployment History** tab contains log entries for every deployment operation (Deployed, Undeployed) of a Virtual Service. However, the **Deployed Assets** tab contains log entry only for the Deployed operation of the Virtual Service. After you undeploy the Virtual Service, the log entry gets automatically get removed from the **Deployed Assets** tab.
- You can never delete the last Deployed activity log of a Virtual Service through the **Deployment History** tab.

> To delete activity logs

1. In CentraSite Control, go to **Operations -> Deployment**.
2. In the **Deployed Assets** tab, select one or more activity logs you want to delete.
3. On the **Actions** menu, click **Delete**.
4. Click **OK** in the confirmation dialog box.

Each selected log is permanently removed from the CentraSite Registry Repository. The activity logs in the **Deployment** profile of the Virtual Service, and the **Service** profile of the target, are not affected.

Deleting Activity Logs Through Deployment History Tab

To view the Deployment History Log, you must have the CentraSite Administrator, Organization Administrator, or Operations Administrator role.

Follow these general guidelines while deleting deployment activity logs:

- Deleting an activity log through the **Deployment History** profile will not affect the deployment status of a Virtual Service. However, deleting an activity log through the **Deployed Assets** tab will undeploy the Virtual Service from the Mediator target, and will stich the deployment status to **Undeployed**.
- The **Deployment History** tab contains log entries for every deployment operation (Deployed, Undeployed) of a Virtual Service. However, the **Deployed Assets** tab contains log entry only for the Deployed operation of the Virtual Service. After you undeploy the Virtual Service, the log entry gets automatically get removed from the **Deployed Assets** tab.
- You can never delete the last Deployed activity log of a Virtual Service through the **Deployment History** tab.

> To delete activity logs

1. In CentraSite Control, go to **Operations -> Deployment**.
2. In the **Deployment History** tab, select one or more activity logs you want to delete.
3. On the **Actions** menu, click **Delete**.
4. Click **OK** in the confirmation dialog box.

Each selected log is permanently removed from the CentraSite Registry Repository. The activity logs in the **Deployment** profile of the Virtual Service, and the **Service** profile of the target, are not affected.

Deleting Virtual Services

If you are not the owner of a Virtual Service asset, you cannot delete the Virtual Service unless you have Full permission on the Virtual Service (granted though either a role-based permission or instance-level permission).

The following general guidelines apply when deleting a Virtual Service asset in CentraSite Control:

- A Virtual Service asset can only be deleted if it is not the target of an association from another registry object.

- When you delete a Virtual Service asset, CentraSite removes the catalog entry for the Virtual Service (that is, it removes the instance of the Virtual Service asset from CentraSite's object database). Also note that:
 - The performance metrics and event information of the Virtual Service are also deleted.
- Note:**
When you delete the Virtual Service, this information is deleted by the built-in action **Delete RuntimeEvents and RuntimeMetrics of Service**, which is installed with CentraSite.
- When you delete a composite Virtual Service asset, all of its nonshared components are also deleted.
 - Deleting a Virtual Service asset will *not* remove:
 - Other assets to which the Virtual Service refers (unless the reference is to an asset that is a nonshared component of the Virtual Service you are deleting). For example, if you are deleting a Virtual Service which has a Consumes or Consumed By relationship with other services in the registry, the related services will not be deleted.
 - Supporting documents that are attached to the Virtual Service.
 - Earlier versions of the Virtual Service. Only the latest version of an asset can be deleted; to remove earlier versions, they must be purged.
 - You cannot delete a Virtual Service asset if:
 - The Virtual Service is in a pending state (for example, awaiting approval).
 - Any user in your CentraSite registry is currently modifying the Virtual Service.

➤ To delete Virtual Service assets

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the displayed list of asset types, select **Virtual Service**.

The Virtual Service assets (for which you have the View permission) are displayed in the **Assets** pane.

3. Right-click a Virtual Service you want to delete, and then click **Delete**.

You can also select multiple Virtual Services, click the **Actions** menu, and click **Delete**.

4. Click **OK** in the confirmation dialog box.

Note:

If you have selected a set of Virtual Services, where one or multiple Virtual Services are in a pending state (for example, awaiting approval), CentraSite ignores the pending list of

Virtual Services, and deletes any remaining Virtual Services for which you have the required permission.

General Procedures across Assets

This section outlines the general procedures across assets performed through CentraSite Control.

Displaying Event Information

The **Events** profile displays the runtime events of an asset. You can filter the list by target, event type and time period.

The **Events** profile contains information about runtime events that have occurred in a target (that is, a policy-enforcement point (PEP) or a run-time monitoring component).

The target publishes to CentraSite the runtime events that have occurred (assuming that the target type contains a MIB file in its target definition file).

CentraSite provides predefined event types for use with webMethods Mediator, or any third-party policy-enforcement point (PEP), or runtime monitoring component such as Insight that is integrated with CentraSite. In addition, you can create custom event types using CentraSite Control.

For procedures on viewing runtime event information for assets, see [“Displaying Event Information for Assets” on page 1500](#).

Displaying Performance Metrics

The **Performance** profile displays the Key Performance Indication (KPI) metrics of an asset.

Gateways capture runtime metrics for assets. If you are using the Mediator gateway, Mediator's data collector captures KPI metrics for each asset and publishes them to CentraSite at regular intervals. If you are using a runtime monitoring component such as Insight, the monitoring component captures the KPI metrics of all rogue assets and publishes them to CentraSite at regular intervals.

For procedures on viewing runtime metrics information for assets, see [“Displaying Performance Metrics for Assets” on page 1502](#).

Displaying Policy Information

The **Policies** profile displays the following information for an asset:

- For a Virtual Service asset, this profile displays a list of all design-time and run-time policies that apply to it.
- If your site uses Insight or some other run-time monitoring device (such as Layer 7) to capture rogue assets, this profile displays the rogue WS-Policy assets.

➤ **To display policy information for an asset**

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the **Assets** pane, right-click an asset whose policy information you want to display, and click **Details**.

You can also select multiple assets, click the **Actions** menu, and click **Details**.

3. Click the **Policies** tab.
4. Click the name of a policy to view its details.

Viewing Deployment History Log

To view the Deployment History Log, you must have the CentraSite Administrator, Organization Administrator, Operations Administrator, or Mediator Publisher role.

The **Deployment History Log** contains information about Virtual Services that CentraSite has pushed to the Mediator target for deployment.

➤ To view the Deployment History Log

1. In CentraSite Control, go to **Operations > Deployment**.
2. Click the **Deployment History** tab.

A list of Virtual Services that are pushed to the Mediator target for deployment is displayed.

3. To filter the list, type a partial string in the **Search** field. CentraSite applies the filter to the **Name** column. The **Search** field is a type-ahead field, so as soon as you enter any characters, the display will be updated to show only those Virtual Services whose name contains the specified characters. The wildcard character “%” is supported.

The **Deployment History** tab displays the following information about a Virtual Service:

Column	Description
Rule ID	The synchronization rule-ID that CentraSite automatically generates on creation of a Mediator target in CentraSite Control.
Name	The name assigned to a Virtual Service.
Type	The type of a Virtual Service (Web Service, XML Service, REST Service, OData Service).
Version	The user-assigned version identifier for a Virtual Service.
Target Name	The name of the target on which a Virtual Service is deployed.

Column	Description
Action	The deployment action of a Virtual Service on the Mediator target. The possible values are: Deployed, Undeployed.
Status	The deployment status of a Virtual Service on the Mediator target. The possible values are: Success, Failed.
Date	The date/time that the Virtual Service has deployed, redeployed, or undeployed.

4. To view details of a particular deployment workflow, click the hyperlinked value in the **Name** column.

Deleting Activity Logs Through Deployed Assets Tab

To view the Deployment History Log, you must have the CentraSite Administrator, Organization Administrator, or Operations Administrator role.

The following general guidelines apply when deleting a deployment activity log in CentraSite Control:

- Deleting an activity log through the **Deployment History** profile will not affect the deployment status of a Virtual Service. However, deleting an activity log through the **Deployed Assets** tab will undeploy the Virtual Service from the Mediator target, and will stich the deployment status to **Undeployed**.
- The **Deployment History** tab contains log entries for every deployment operation (Deployed, Undeployed) of a Virtual Service. However, the **Deployed Assets** tab contains log entry only for the Deployed operation of the Virtual Service. After you undeploy the Virtual Service, the log entry gets automatically get removed from the **Deployed Assets** tab.
- You can never delete the last Deployed activity log of a Virtual Service through the **Deployment History** tab.

➤ To delete activity logs through the Deployed Assets tab

1. In CentraSite Control, go to **Operations -> Deployment**.
2. In the **Deployed Assets** tab, select one or more activity logs you want to delete.
3. On the **Actions** menu, click **Delete**.
4. Click **OK** in the confirmation dialog box.

Each selected log is permanently removed from the CentraSite registry/repository. The activity logs in the **Deployment** profile of the Virtual Service, and the **Service** profile of the target, are not affected.

Deleting Activity Logs Through Deployment History Tab

To view the Deployment History Log, you must have the CentraSite Administrator, Organization Administrator, or Operations Administrator role.

The following general guidelines apply when deleting a deployment activity log in CentraSite Control:

- Deleting an activity log through the **Deployment History** profile will not affect the deployment status of a Virtual Service. However, deleting an activity log through the **Deployed Assets** tab will undeploy the Virtual Service from the Mediator target, and will stich the deployment status to **Undeployed**.
- The **Deployment History** tab contains log entries for every deployment operation (Deployed, Undeployed) of a Virtual Service. However, the **Deployed Assets** tab contains log entry only for the Deployed operation of the Virtual Service. After you undeploy the Virtual Service, the log entry gets automatically get removed from the **Deployed Assets** tab.
- You can never delete the last Deployed activity log of a Virtual Service through the **Deployment History** tab.

> To delete activity logs through the Deployment History Tab

1. In CentraSite Control, go to **Operations -> Deployment**.
2. In the **Deployment History** tab, select one or more activity logs you want to delete.
3. On the **Actions** menu, click **Delete**.
4. Click **OK** in the confirmation dialog box.

Each selected log is permanently removed from the CentraSite registry/repository. The activity logs in the **Deployment** profile of the Virtual Service, and the **Service** profile of the target, are not affected.

Managing Virtual Service Assets through the Command Line Interface

This section describes operations you can perform to manage virtual service assets, such as, Virtual Services, Virtual REST Services, and Virtual OData Services through the CentraSite Command Line Interface (CLI).

Deploying a Virtual Service to Gateway

Pre-requisites:

To deploy (that is, publish) a virtual service to an API Gateway or a Mediator gateway through the CentraSite CLI, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `deploy` for this purpose.

➤ To deploy a virtual service through the command line

- Run the command `deploy`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd deploy [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -virtualService <VIRTUAL-SERVICE> -gateway <GATEWAY> -deploytimeout <DEPLOY_TIMEOUT>`

The input parameters are:

Input Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
VIRTUAL-SERVICE	The name or UDDI key of a virtual service to be deployed.
GATEWAY	The gateway to which a virtual service identified by the parameter <code>VIRTUAL-SERVICE</code> is to be deployed.
DEPLOY_TIMEOUT	(Optional). The maximum time, in seconds, allowed for the deployment operation to respond. If the deployment operation does not respond before this timeout, the operation is rolled back. By default, <code>DEPLOY_TIMEOUT</code> is set to 60 seconds.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd deploy -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-virtualService VS1 -gateway Gateway1 -deploytimeout 60
```

Undeploying a Virtual Service from Gateway

Pre-requisites:

To undeploy (that is, unpublish) a virtual service from an API Gateway or a Mediator gateway through the CentraSite CLI, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `undeploy` for this purpose.

➤ To undeploy a virtual service from a gateway

- Run the command `undeploy`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd undeploy [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -virtualService <VIRTUAL-SERVICE> -gateway <GATEWAY>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
VIRTUAL-SERVICE	The name or key of a virtual service to be undeployed.
GATEWAY	The gateway from which a virtual service identified by the parameter <code>VIRTUAL-SERVICE</code> is to be undeployed.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd undeploy -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-virtualService VS1 -gateway Gateway1
```

Bulk Deploying Virtual Services to Gateway

Pre-requisites:

To deploy (that is, publish) multiple virtual services to an API Gateway or a Mediator gateway through the CentraSite CLI, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `bulk deploy` for this purpose.

➤ To deploy multiple virtual services through the command line

- Run the command `bulk deploy`.

The syntax is of the format:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd bulk deploy [-url
<CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -gateway <GATEWAY>
-deploytimeout <DEPLOY_TIMEOUT> -deploybatchsize <DEPLOY_BATCH_SIZE>
```

The input parameters are:

Input Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
GATEWAY	The gateway to which all the virtual services are to be deployed.
DEPLOY_TIMEOUT	(Optional). The maximum time, in seconds, allowed for the deployment operation to respond. If the deployment operation does not respond before this timeout, the operation is rolled back. By default, <code>DEPLOY_TIMEOUT</code> is set to 60 seconds.
DEPLOY_BATCH_SIZE	(Optional). The maximum number of virtual services to be deployed to the gateway before a syncpoint is taken. By default, <code>DEPLOY_BATCH_SIZE</code> is set to 50.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd bulk deploy -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-gateway Gateway1 -deploytimeout 60 -deploybatchsize 50
```

Bulk Undeploying Virtual Services from Gateway

Pre-requisites:

To undeploy (that is, unpublish) multiple virtual services from an API Gateway or a Mediator gateway through the CentraSite CLI, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `bulk undeploy` for this purpose.

➤ To undeploy multiple virtual services from a gateway

- Run the command `bulk undeploy`.

The syntax is of the format:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd bulk undeploy [-url
<CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -gateway <GATEWAY>
```

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
GATEWAY	The gateway from which all the virtual services are to be undeployed.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd bulk undeploy -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-gateway Gateway1
```

Bulk Redeploying Virtual Services to Gateway

Pre-requisites:

To redeploy (that is, republish) multiple virtual services to an API Gateway or a Mediator gateway through the CentraSite CLI, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `bulk redeploy` for this purpose.

You can also use the command tool named `bulk clean redeploy` to undeploy and redeploy all virtual services that are deployed to a specific gateway.

> To redeploy multiple virtual services through the command line

- Run the command `bulk redeploy`.

The syntax is of the format:

- `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd bulk redeploy [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -gateway <GATEWAY> -deploytimeout <DEPLOY_TIMEOUT> -deploybatchsize <DEPLOY_BATCH_SIZE>`
- `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd bulk clean redeploy [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -gateway <GATEWAY> -deploytimeout <DEPLOY_TIMEOUT> -deploybatchsize <DEPLOY_BATCH_SIZE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
GATEWAY	The gateway to which all the virtual services are to be redeployed.
DEPLOY_BATCH_SIZE	(Optional). The maximum number of virtual services to be redeployed to the gateway before a syncpoint is taken. By default, <code>DEPLOY_BATCH_SIZE</code> is set to 50.
DEPLOY_TIMEOUT	(Optional). The maximum time, in seconds, allowed for the redeployment operation to respond. If the redeployment operation does not respond before this timeout, the operation is rolled back. By default, <code>DEPLOY_TIMEOUT</code> is set to 60 seconds.

Examples (all in one line):

- `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd bulk redeploy -url http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage -gateway Gateway1 -deploytimeout 60 -deploybatchsize 50`
- `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd bulk clean redeploy -url http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage -gateway Gateway1 -deploytimeout 60 -deploybatchsize 50`

Important Considerations when Configuring Virtual Services

Handling Services with Multiple Ports and Bindings

Mediator implicitly assumes that there is a one-to-one mapping between a WSDL Service and a Virtual Service. A problem arises if a `<service>` element contains multiple `<port>` elements that point to different bindings (and consequently different port types) -- the problem is that Mediator creates a Virtual Service that has the operations from only *one* of the `portTypes`. Mediator chooses the first port available under `<service>` and exposes the operations corresponding to the equivalent binding/portType.

For example, consider the following WSDL fragment that shows the structure of the `portType`, `binding` and `service` elements in the WSDL. Note that there are:

- Two distinct `<portType>` elements: `SystemPortType` and `CustomerPortType`.
- Two equivalent bindings defined for each `<portType>`: `SystemBinding` and `CustomerBinding`.

- A single `<service>` element that defines the two ports with distinct endpoints (one for each binding available).

Example `<portType>` Elements

```
<portType name="SystemPortType">
  <operation name="ping">
    ...
  </operation>
</portType>
<portType name="CustomerPortType">
  <operation name="getOperation1">
    ...
  </operation>
  <operation name="getOperation2">
    ...
  </operation>
  <operation name="getOperation3">
    ...
  </operation>
  <operation name="getOperation4">
    ...
  </operation>
</portType>
```

Example `<binding>` Elements

```
<binding name="SystemBinding" type="tns:SystemPortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="ping">
    ...
  </operation>
</binding>
<binding name="CustomerBinding" type="tns:CustomerPortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getOperation1">
    ...
  </operation>
  <operation name="getOperation2">
    ...
  </operation>
  <operation name="getOperation3">
    ...
  </operation>
  <operation name="getOperation4">
    <soap:operation soapAction="urn:customer.service.cm.be/getOperation4"/>
    ...
  </operation>
</binding>
```

Example `<service>` Element

```
<service name="CustomerRefService">
  <port name="SystemPort" binding="tns:SystemBinding">
    <soap:address location="http://.../v4/SystemPort"/>
  </port>
  <port name="CustomerPort" binding="tns:CustomerBinding">
```

```
<soap:address location="http://.../v4/CustomerRefPort"/>
  </port>
</service>
```

Workaround Option 1

You can create two different Virtual Services. That is, you can expose this Native Service as two Virtual Services -- one for each operation that needs to be invoked:

- A service for the `getXXX` operations (for example, a Virtual Service called `VS_Customer`).
- A service for the `ping` operation (for example, a Virtual Service called `VS_ping`).

➤ To create a different Virtual Service for each operation

1. Create a Virtual Service for the Native Service `CustomerRefService` and name it `VS_Customer`, for example.
2. Configure `VS_Customer` and configure its routing protocol as `Straight Through`.
3. Specify the routing address as `http://... /v4/CustomerRefPort` (for `CustomerBinding`, where all the `getXXX` operations are supported).
4. On the Virtual Service's **Summary** profile, click on the URL for the WSDL (this a copy of the Virtual Service template WSDL, very similar to the original Native Service WSDL), and download or save it to your local file system.

(Make an additional copy of this downloaded WSDL in case you make a mistake in your editing.)

5. Remove the following elements from the WSDL and then save it:

- `SystemPortType`
- `SystemBinding`
- `SystemPort`

Note:

Make sure your browser or XML tool can read this modified WSDL without any syntax error.

6. Attach the modified WSDL file to the Virtual Service by selecting the **Attach WSDL** command in the Virtual Service's **Actions** menu.
7. Create another Virtual Service for the Native Service `CustomerRefService` and name it `VS_ping`, for example. Repeat the above steps but with the following differences:
 - Specify the routing address as `http://... /v4/SystemPort`.

- Remove the following elements from the WSDL:
 - CustomerPortType
 - CustomerBinding
 - CustomerPort

8. Deploy both Virtual Services to Mediator.

Workaround Option 2

With this option, you expose the Native Service as one Virtual Service. The Web Service client accesses the service through one address to the Virtual Service for all the possible operations (ping and getXXX). The Virtual Service then takes care of routing to the correct endpoint for the different operations. This is accomplished by using Content-based routing (instead of Straight Through routing) to determine the operation being called (based on the SOAP request content) and then forwarding the request to the correct endpoint.

➤ To create a Virtual Service with Content-based routing

1. Create a Virtual Service for the Native Service `CustomerRefService` and name it `VS_CustomerRefService`, for example.
2. Configure `VS_CustomerRefService` and configure its routing protocol as Content-based.
3. On the **Routing Protocols** tab, construct the routing rule as follows:
 - a. Click **Endpoint** (next to the **Route To** column).
 - b. In the **Search for Endpoint** dialog that appears, click **Search** to search for the Web service endpoint to route the requests to.
 - c. Select `http://... /v4/SystemPort` (the Accessing URI that goes to ping operation).
 - d. To create an XPath expression for the routing rule, click **Edit** (next to the **If True** column).
 - e. In the XPath Editor that appears, click the **All Nodes** tab, expand the namespace's node, click to highlight the `tns:ping` element, and click **OK**.
 - f. Double check that you have something like this in the rule and it is routed to `SystemPort`:

```
/soapenv:Envelope/soapenv:Body/tns:pingRequest
```
4. Set the **Default To** routing field to the routing address `http://... /v4/CustomerRefPort` (for `CustomerBinding`, where `getXXX` operations are supported).

5. On the Virtual Service's **Summary** profile, click on the URL for the WSDL (this a copy of the Virtual Service template WSDL, very similar to the original Native Service WSDL), and download or save it to your local file system.

(Make an additional copy of this downloaded WSDL in case you make a mistake in your editing.)

6. Modify the WSDL as follows and then save it:
 - a. Copy the ping operation from `SystemPortType` and add into `CustomerPortType`.
 - b. Delete the `SystemPortType`. The objective here is to make *one* port type only.
 - c. Update the `SystemBinding` to also refer to `CustomerPortType`, since `SystemPortType` has been deleted.

Note:

The `soapAction` attribute must be specified for the `soap:operation` element to ensure that Mediator can resolve the operation being invoked for this service.

- d. Save the WSDL.

Note:

Ensure that your browser or XML tool can read this modified WSDL without any syntax error.

7. Attach the modified WSDL file to the Virtual Service by selecting the **Attach WSDL** command in the Virtual Service's **Actions** menu.
8. Deploy the Virtual Service to Mediator.

Important Considerations when Configuring Virtual REST Services

With Web REST Services, the Native REST Service endpoint that is sent by CentraSite to Mediator inside a Virtual REST Service definition is a static element. Once the Virtual REST Service is successfully deployed to Mediator, the real endpoint is returned to CentraSite as part of the response message during deployment. At run time, when a SOAP request is received, that request is POSTed to the endpoint that is statically defined in the Virtual REST Service definition.

However, with REST Services or XML REST Services, the endpoint is flexible. The REST Services or XML REST Services describe data as resources. The resources are accessible through logical endpoints that have application meaning to users. For example, a collection of textbooks might be defined as a resource with the following URL:

```
http://{host}:{port}/books
```

A specific book with an identifier of 1234 would be accessible with the following URL:

```
http://{host}:{port}/books/1234
```

Due to this difference and others, you should keep the following topics described in this chapter in mind when you configure a REST Virtual REST Service.

Endpoint Manipulation of REST Virtual REST Services

When you configure a Virtual REST Service, you specify the Native REST Service name, an endpoint, and the HTTP method type(s) that are included in the message (GET, POST, PUT, PATCH, DELETE). From this information, CentraSite generates a Virtual REST Service definition that includes REST Service and operation elements, as well as an endpoint and binding element pair for each HTTP method specified.

CentraSite generates an operation name to be included when the Virtual REST Service definition deployment message is sent to Mediator. This means that if you create a Virtual REST Service called VS1, and you specify a Native endpoint, then the endpoint exposed by Mediator for calling the Virtual REST Service will be /ws/VS1/Invoke.

For example, assume the following endpoints are deployed:

- Native REST Service endpoint: `http://localhost:8080/RESTServices/mtc/member`
- Virtual REST Service endpoint: `http://localhost:5555/ws/VS1/Invoke`

Assume that the example Virtual REST Service is deployed with two HTTP method bindings: GET and POST. Both of these bindings have operation elements that include the same HTTP location attribute: member. To better illustrate the functionality, the examples below show a series of sample requests from a consumer including the requests' HTTP method. (At run time, REST message detection is dependent upon a consumer using the correct Content-Type when a request is sent.) Each example shows the expected endpoint that Mediator will send after it has rewritten the endpoint prior to Native REST Service invocation.

Example 1

For a GET, assume that:

The request Content-Type is: `application/x-www-form-urlencoded`

The endpoint received by Mediator is: `http://localhost:5555/ws/VS1/Invoke`

The Native REST Service endpoint sent by Mediator is: `http://localhost:8080/RESTServices/mtc/member`

The application function is: The Native REST Service returns a collection of members with summary information.

Example 2

For a GET, assume that:

The request Content-Type is: `application/x-www-form-urlencoded`

The endpoint received by Mediator is: `http://localhost:5555/ws/VS1/Invoke/1234`

The Native REST Service endpoint sent by Mediator is: `http://localhost:8080/REST/Services/mtc/member/1234`

The application function is: The Native REST Service returns summary data for a member with this key.

Example 3

For a GET, assume that:

The request Content-Type is: `application/x-www-form-urlencoded`

The endpoint received by Mediator is: `http://localhost:5555/ws/VS1/Invoke/1234?detail=true`

The Native REST Service endpoint sent by Mediator is: `http://localhost:8080/REST/Services/mtc/member/1234?detail=true`

The application function is: Query parameters remain intact. Returns a response message with more member details.

Example 4

For a POST, assume that:

The request Content-Type is: `application/xml` or `application/json`

The endpoint received by Mediator is: `http://localhost:5555/ws/VS1/Invoke/1234`

The Native REST Service endpoint sent by Mediator is: `http://localhost:8080/REST/Services/mtc/member/1234`

The application function is: The request message provides the contents needed to create the member resource.

Example 5

For a GET, assume that:

The request Content-Type is: `application/x-www-form-urlencoded`

The endpoint received by Mediator is: `http://localhost:5555/ws/VS1/Invoke/joe`

The Native REST Service endpoint sent by Mediator is: `http://localhost:8080/REST/Services/mtc/member/joe`

The application function is: Fetches the member defined with the login joe. (Mediator contains no metadata in its REST Service deployment to differentiate between the login vs. key GET requests.)

Example 6

For a GET, assume that:

The request Content-Type is: `application/x-www-form-urlencoded`

The endpoint received by Mediator is: `http://localhost:5555/ws/VS1/Invoke?type=login&value=joe`

The Native REST Service endpoint sent by Mediator is: `http://localhost:8080/REST Services/mtc/member?type=login&value=joe`

The application function is: The Native REST Service might also support a static endpoint with constraints defined in query parameters. Mediator also supports this approach.

The Request Message's HTTP Methods and Content-Types for REST and XML REST Services

When you configure the Entry Protocol step of a Virtual REST Service, it is important to specify all the HTTP methods that are supported for the REST Service. For example, if the Virtual REST Service is deployed to Mediator and you selected only the GET method in the Virtual REST Service's details page, then Mediator permits GET invocations. In this case, a POST request will be rejected with a return of statusCode 405 even if the Native REST Service happens to support POSTs.

It is important that the client's requests contain an HTTP Content-Type header. At run time, Mediator determines which message builder to use based on the message's HTTP method and its Content-Type. (The absence of the soapAction header will indicate to Mediator that the message is an XML message.)

The valid HTTP method/Content-Type combinations are as follows:

This method... Can be included in a message of this Content-Type...

POST application/xml
 application/json
 application/x-www-form-urlencoded
 multipart/form-data
 or text/xml

PUT application/xml
 application/json
 application/x-www-form-urlencoded
 multipart/form-data
 or text/xml

GET application/x-www-form-urlencoded

Note:
 For Axis Free Mediation Content-Type is not required.

DELETE application/x-www-form-urlencoded

Note:
 For Axis Free Mediation Content-Type is not required.

Note:
 Keep the following points in mind:

- If Mediator receives a request sent with an HTTP method that is not specified in the Virtual REST Service or Virtual XML REST Service definition, it will return a 405 error.

- If Mediator receives a request sent with a wrong Content-Type, it will return a 415 error. In addition, if the wrong Content-Type is used with a GET or DELETE, then the query parameters contained in the message (if any) will not be processed.

Changing the HTTP Method of a REST Request

When configuring a REST Virtual REST Service, you specify whether to route the requests to the Native REST Service with the same HTTP method that is contained in the requests (GET, POST, PUT, PATCH, DELETE), or whether to route the requests with a different HTTP method.

Typically you want to pass each request to the Native REST Service with the same HTTP method that is contained in the request. For example, if a request contains a GET method, you allow the GET method to be passed to the Native REST Service. However, there might be rare cases in which you want to change the HTTP method of a request to different HTTP method. For example, you might want to:

- Expose an XML REST Service as a REST Service.

In this case, the REST Service you create would be a Virtual XML REST Service that exposes the HTTP methods GET, POST, PUT, PATCH and DELETE, but the routing method would always be POST.

- Expose a REST Service whose Virtual REST Service only exposes the POST method.

➤ To change the HTTP method of a REST request

- On the REST Virtual REST Service, set the value of the HTTP Method field either statically (by explicitly setting the value to **GET**, **POST**, **PUT**, or **DELETE**) or dynamically (by setting the value to **Use Context Variable**).

In order to use the **Use Context Variable** option to set the field dynamically, you must write a webMethods IS REST Service that sets a value of GET, POST, PUT, PATCH or DELETE for a predefined context variable named ROUTING_METHOD. You need to invoke this REST Service in the Virtual REST Service's **Invoke webMethods IS REST Service** action. For more information on using contexts and variables dynamically, see the Changing HTTP Methods in Requests Dynamically using a Context Variable section later in this topic.

CAUTION:

Use this feature carefully, since changing HTTP methods to certain other HTTP methods could result in unintended results or errors.

For example, changing an inbound GET request to a DELETE request would be a serious mistake if the deletion was not intended and the Native REST Service actually deleted a resource when invoked with a DELETE method. Additionally, an incoming POST or PUT request cannot be translated into a GET or DELETE if the request has nested elements.

The Implications of Changing HTTP Methods

When changing this incoming HTTP method...	To...	Note that...
GET	POST	<ul style="list-style-type: none"> ■ The Content-Type of the changed request is sent as application/xml or application/json, and the charset is UTF-8. ■ Depending on the structure of the Native REST Service, be aware that the Native REST Service might not be expecting the same payload structure that is being sent. In this case, you would need to transform the request message into the format required by the Native REST Service before Mediator sends the requests to the Native REST Service. For more information, see the Sample XSLT Transformation for GET-to-POST or GET-to-PUT section later in this topic.
GET	PUT	Identical to GET-to-POST, except that Mediator changes the request's HTTP method from GET to PUT.
GET	DELETE	No comment.
POST	GET	<ul style="list-style-type: none"> ■ Mediator will translate the POSTed request elements into query string parameters, in a root element.

Note:

An incoming POST or PUT request cannot be translated into a GET or DELETE if the request has nested elements. For example:

```
(this is correct)
<person>
  <lastName>Smith</lastName>
</person>
(this is incorrect)
<person>
  <name>
    <last>Smith</last>
  </name>
</person>
```

- If you want to send additional parameters as part of the request URL, you can transform this payload. To do this, you can use an XSLT file or a webMethods IS REST Service call to add

When changing this incoming HTTP method...	To...	Note that...
		parameters before the request is sent to the Native REST Service.
POST	DELETE	Identical to POST-to-GET, except that Mediator changes the request's HTTP method from POST to DELETE.
POST	PUT	The Content-Type of the changed request is sent as application/xml or application/json, and the charset is UTF-8.
PUT	GET	Identical to POST-to-GET, except that Mediator changes the request's HTTP method from PUT to GET.
PUT	POST	The Content-Type of the changed request is sent as application/xml or application/json, and the charset is UTF-8.
PUT	DELETE	Identical to POST-to-DELETE, except that Mediator changes the request's HTTP method from PUT to DELETE.
DELETE	GET	No comment.
DELETE	POST	Identical to GET-to-POST, except that Mediator changes the request's HTTP method from DELETE to POST.
DELETE	PUT	Identical to GET-to-PUT, except that Mediator changes the request's HTTP method from DELETE to PUT.
GET, POST, PUT or DELETE	Use Context Variable	See the Changing HTTP Methods in Requests Dynamically using a Context Variable section later in this topic.
GET or DELETE	POST or PUT	Note that the query parameters will be picked off the URL and stored as top-level elements when the message is sent to the Native REST Service. The query parameters are ignored on the endpoint URL and lost when we POST to the Native provider (that is, don't change the protocol method).
PATCH	POST or PUT	Note that the query parameters will be picked off the URL and stored as top-level elements when the message is sent to the Native REST Service. The query parameters are ignored on the endpoint URL and lost when we POST to the Native provider (that is, don't change the protocol method).

Changing HTTP Methods in Requests Dynamically using a Context Variable

Alternatively, instead of changing an HTTP method explicitly (statically) to PUT, POST, GET, PATCH, or DELETE, you can change the HTTP method to the value of a predefined context variable (ROUTING_METHOD) that dynamically resolves to a different HTTP method (PUT, POST, GET, PATCH, or DELETE, as appropriate).

To change the HTTP method dynamically, you create a webMethods IS REST Service and invoke it in the Virtual REST Service's **Invoke webMethods IS REST Service** action. This webMethods IS REST Service should reference the predefined context variable ROUTING_METHOD. To set the value of ROUTING_METHOD, use the setContextVariableValue method, which is defined in the following class:

com/softwareag/mediator/REST Service/MediatorRuntimeFacade.java

For example:

```
public static final void updateHttpMethod(IData pipeline)
    throws REST ServiceException {
    String mcKey = "Message Context";
    Object obj = IDataUtil.get(pipeline.getCursor(), mcKey);
    if (obj!=null && obj instanceof org.apache.synapse.MessageContext) {
        MessageContext msgCtx = (MessageContext) obj;
        QName varName =
            new QName(MediatorContextVariableType.ROUTING_METHOD.getName());
        MediatorRuntimeFacade.setContextVariableValue(varName, "PUT", msgCtx );
    }
}
```

Sample XSLT Transformation for GET-to-POST or GET-to-PUT

As stated in the above table, depending on the structure of the Native REST Service, the Native REST Service might not be expecting the same payload structure that is being sent. In this case, you would need to transform the request message into the format required by the Native REST Service before Mediator sends the requests to the Native REST Service. To do this, you invoke an XSLT file at run time.

Assume that:

- The Native REST Service name is authors.
- The Virtual REST Service or Virtual XML REST Service for authors is named vs-authors and is made available in Mediator at this endpoint: <http://localhost:5555/ws/vs-authors/Invoke>. The targetNamespace of the Virtual REST Service or Virtual XML REST Service is <http://example.com/authors>.

Following is a sample XSLT transformation file for the GET-to-POST or GET-to-PUT scenario:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:ns="http://example.com/authors"
    version="1.0">

    <xsl:output method="xml" omit-xml-declaration="no" standalone="yes"
        indent="yes"/>
```

```

<xsl:strip-space elements="*" />
<xsl:template match="node()|@*">
  <xsl:copy>
    <xsl:apply-templates select="node()|@*" />
  </xsl:copy>
</xsl:template>

<xsl:template match="//ns:invoke/node()">
  <xsl:element name="{local-name(.)}">
    <xsl:value-of select="." />
  </xsl:element>
</xsl:template>

<xsl:template match="//ns:invoke">
  <xsl:element name="authors">
    <xsl:apply-templates />
  </xsl:element>
</xsl:template>
</xsl:stylesheet>

```

Working with the JSON Content-Type

Mediator can accept a REST Service request that specifies the Content-Type application/json (or application/json/badgerfish) and the HTTP methods PUT, GET, DELETE, PATCH, and POST.

Assuming that the Native REST Service supports both JSON and the HTTP method(s) specified in the request, Mediator can determine the correct REST Service, operation and output format (JSON) to return to the consuming application. There are different ways in which a Native REST Service provider can be prompted to return response content. It will vary with the provider. For example, some providers may rely on the Accept transport header to specify the format the consumer wants. Others may use an element in the request or a query parameter on the URL.

However, suppose for example that the Native REST Service does not support the HTTP method specified in the request (for example, POST). As a workaround, you can configure the Virtual REST Service so that it bridges this difference between the consumer request and the Native REST Service. In this case, you can configure the Virtual REST Service so that it takes the POST and bridges it into an HTTP GET query, and then returns the REST Service to the consumer in the expected JSON format. To implement this, you set the following predefined context variables in a user-defined webMethods IS REST Service that you can invoke in the Virtual REST Service's

Invoke webMethods IS REST Service action:

- **MESSAGE_TYPE**: A Content-Type defined in axis2.xml for a message formatter. This value must be a key in the axis2 message formatters list, since it is used to control message serialization. (The valid choices are defined as attributes of the <messageFormatters/> group in the Integration Server's axis2.xml configuration.)
- **BUILDER_TYPE**: A Content-Type defined in axis2.xml for a message builder. This value must be a key in the axis2 message builders list since it is used to control building of Native REST Service response messages. (The valid choices are defined as attributes of the <messageBuilders/> group in the Integration Server's axis2.xml configuration.)

This and other bridging scenarios are discussed in this section.

Note:

Mediator does not support JSON to XML or XML to JSON transformation as part of Request or Response processing. However, you can perform JSON to XML or XML to JSON transformation using the **Invoke webMethods Integration Server Service** as part of Request or Response processing. You must create these services using Software AG Designer and configure the full name of the service in the **Invoke webMethods Integration Server** policy.

How Mediator Determines Which Builder and Formatter Classes to Use (and How You Can Override Them)

Mediator makes these determinations at run time as follows:

This table also summarizes how you can override the default determinations.

Run-time Step	Description
Mediator receives the request from the client	It is important that the client's PUT or POST requests contain the HTTP header Content-Type because the Content-Type header determines the message builder Mediator uses to parse the input stream. (GET or DELETE request do not require a Content-Type header.)
Mediator sends the request to the REST Service provider	<p>Mediator uses a message formatter to serialize the request, and then sends the serialized request to the Native REST Service provider.</p> <p>Mediator determines the message formatter to use as follows:</p> <ul style="list-style-type: none"> ■ If you explicitly specify a message formatter (by setting the MESSAGE_TYPE context variable in a webMethods IS REST Service and invoking this REST Service in the Virtual REST Service's Invoke webMethods IS REST Service action), then Mediator uses that formatter. The Content-Type header that Mediator sends to the Native provider is the one that is associated with the MESSAGE_TYPE context variable. ■ Else, Mediator uses the message formatter associated with the Content-Type sent by the client (and sends the Content-Type to the Native provider). ■ Else, if no Content-Type was sent by the client, then: <ul style="list-style-type: none"> ■ For PUT requests, the default formatter used (and the Content-Type header that Mediator sends to the Native provider) is <code>application/xml</code>. ■ For POST requests, the default formatter used (and the Content-Type header that Mediator sends to the Native provider) is SOAP (and the request will fail).

Run-time Step	Description
	<ul style="list-style-type: none"> ■ For GET or DELETE requests (which do not require a Content-Type header), the default formatter used (and the Content-Type header that Mediator sends to the Native provider) is <code>application/x-www-form-urlencoded</code>.
<p>Mediator receives a response from the REST Service provider</p>	<p>When the provider returns a response to Mediator, a message builder parses the response stream into an Axiom message to be stored in the message context. Mediator determines which message builder to use as follows:</p> <ol style="list-style-type: none"> 1. Mediator selects the message builder associated with the request's Accept transport header, if one was specified. 2. Else, you can set the <code>BUILDER_TYPE</code> context variable in a webMethods IS REST Service, and invoke this REST Service in the Virtual REST Service's Invoke webMethods IS REST Service action. Mediator will check that the builder type is a valid Content-Type for the list of builders in <code>axis2.xml</code>. This variable takes priority over the current setting specified in the Accept transport header. That is, Mediator will only use the Accept header to determine the builder type needed to parse a Native provider response if no IS REST Service was written to set the <code>BUILDER_TYPE</code> context variable. 3. Else, Mediator uses the builder associated with the Content-Type specified in the request (assuming that the Content-Type is one of the types that is mapped in <code>axis2.xml</code>). 4. Else, if no Content-Type was specified in the request (for example, a PUT or POST request with no Content-Type, or any GET or DELETE request), or if the Content-Type is not one of the types that is mapped in <code>axis2.xml</code>, then Mediator will default to <code>application/xml</code>.
<p>Mediator sends a response to the client</p>	<p>Mediator serializes the response and sends it to the client.</p> <p>By default, Mediator uses the formatter that was used to serialize the request sent to the provider. (If the formatter was <code>application/x-www-form-urlencoded</code> (for a GET or DELETE request), then Mediator will instead use <code>application/xml</code> so it can send the response.)</p> <p>You can override the Content-Type that is sent to the client by setting the <code>MESSAGE_TYPE</code> context variable in a webMethods IS REST Service, and invoking this REST Service</p>

Run-time Step	Description
	in the Virtual REST Service's Invoke webMethods IS REST Service action.

Scenarios for Requesting JSON Type REST Services

Following are some of the possible scenarios in which JSON type REST Services can be requested. Many scenarios require that you bridge the differences between consumer requests and the Native REST Service (that is, differing HTTP methods and Content-Types). Three of the scenarios are discussed in more detail following the table:

Consumer Sends	Mediator Sends Request to Provider	Mediator Receives Response from Provider	Mediator Sends Response to Consumer	Requirement for Bridging?
GET	GET	JSON	JSON	Request Processing step bridging
POST/JSON	POST/JSON	JSON	JSON	No bridging needed
GET	GET	XML	JSON	Response Processing step bridging
POST/JSON	GET	JSON	JSON	Request Processing step bridging
POST/JSON	GET	XML	JSON	Request Processing step bridging and Response Processing step bridging
POST/JSON	XSLT/GET *	JSON	JSON	Request Processing step bridging
POST/JSON	XSLT/POST/XML *	XML	XML	Request Processing step bridging
POST/JSON	POST/JSON	JSON/XSLT *	JSON	Response Processing step bridging

Consumer Sends	Mediator Sends Request to Provider	Mediator Receives Response from Provider	Mediator Sends Response to Consumer	Requirement for Bridging?
GET	GET	JSON	XML	Request Processing step bridging and Response Processing step bridging
POST/XML	POST/JSON	JSON	JSON	Request Processing step bridging

* The XSLT references indicate where you can perform an XSLT message transformation at either the **Request Transformation** or **Response Transformation** action.

In the table above, the required Content-Type settings are not shown, but assume the following:

HTTP Method/Request Content	Required Axis2 Content Type
GET or DELETE	application/x-www-form-urlencoded
POST or PUT/XML	application/xml
POST or PUT/Mapped JSON	application/json
POST or PUT/Badgerfish	application/json/badgerfish

JSON Example 1: GET Request, JSON Response

In this example, a consumer sends a GET request to get a Native JSON REST Service. Mediator will send the response to the consumer in the requested JSON format (as indicated by the "output=json" parameter in the query).

The request looks like this:

```
http://localhost:5555/ws/YahooVS/search?query=wsdl20&output=json
```

and because this is a GET request, the Content-Type defaults to `application/x-www-form-urlencoded`.

Note:

For GET or DELETE requests for REST Services, it is not necessary to specify the Content-Type in the request; Mediator will default to `application/x-www-form-urlencoded` for GET or DELETE requests.

The run-time processing will be as follows:

Consumer Sends	Mediator Sends Request to Provider	Mediator Receives Response from Provider	Mediator Sends Response to Consumer	Requirement for Bridging
GET	GET	JSON	JSON	Request Processing step bridging

Since the request is a GET (that is, of Content-Type `application/x-www-form-urlencoded`), but Mediator expects to receive a JSON stream from the provider, you must send the `BUILDER_TYPE application/json` to the Native provider. To do this, write and invoke a `webMethods IS REST Service` in the Virtual REST Service's **Invoke webMethods IS REST Service** action. The IS REST Service should include the following predefined context variable set to this value:

Context Variable	Value
<code>BUILDER_TYPE</code>	<code>application/json</code>

JSON Example 2: POST/JSON Request, JSON Response (where POST is not supported)

In this example, a consumer sends a POST request (of Content-Type `application/json`) to a Native REST Service, but the Native REST Service does not support POST.

The request's Content-Type is `application/json` and its output parameter is set to `xml`. The reason for this is explained below.

The run-time processing will be as follows:

Consumer Sends	Mediator Sends Request to Provider	Mediator Receives Response from Provider	Mediator Sends Response to Consumer	Requirement for Bridging
POST/JSON	GET	XML	JSON	<ul style="list-style-type: none"> ■ Request Processing step bridging ■ Response Processing step bridging

Configure the Virtual REST Service as follows:

- In the Virtual REST Service's Routing Protocols tab, set the value of the HTTP Method field to the value GET. Doing this instructs Mediator to change the POST to an HTTP GET before Mediator sends it to the Native REST Service (which is necessary because the Native REST Service does not support POST).

- Write and invoke a webMethods IS REST Service in the Virtual REST Service's **Invoke webMethods IS REST Service** action. The IS REST Service should include the following predefined context variables set to the values shown below. The request's "query": "wsdl20" and "output": "xml" parameters are transformed into query parameters on the URL before the Native REST Service is invoked. Thus, although the consumer is sending a JSON request, the Native REST Service is instructed to return an XML response to Mediator.

Context Variable	Value
MESSAGE_TYPE	application/x-www-form-urlencoded
BUILDER_TYPE	application/xml

- Write and invoke a webMethods IS REST Service in the Virtual REST Service's **Invoke webMethods IS REST Service** action. The IS REST Service should include the following predefined context variable set to the value shown below. Doing this instructs Mediator to bridge the XML response from the Native REST Service into JSON format, to be returned to the consumer:

Context Variable	Value
MESSAGE_TYPE	application/json

JSON Example 3: GET Request, XML Response

In this example, a consumer sends a GET request to get a Native REST Service. Mediator will send the response to the consumer in the requested XML format (as indicated by the "output=xml" parameter in the query).

The request looks like this:

```
http://localhost:5555/ws/YahooVS/search?query=wsdl20&output=xml
```

and because this is a GET request, the Content-Type defaults to application/x-www-form-urlencoded.

The run-time processing will be as follows:

Consumer Sends	Mediator Sends Request to Provider	Mediator Receives Response from Provider	Mediator Sends Response to Consumer	Requirement for Bridging
GET	GET	JSON	XML	<ul style="list-style-type: none"> Request Processing step bridging Response Processing step bridging

Since the request is a GET (that is, of Content-Type `application/x-www-form-urlencoded`), but Mediator expects to receive a JSON stream from the provider, you must instruct Mediator to send the `BUILDER_TYPE` `application/json` to the Native provider. To do this, write and invoke a `webMethods IS REST Service` in the Virtual REST Service's **Invoke webMethods IS REST Service** action. The IS REST Service should include the following predefined context variable set to this value:

Context Variable	Value
<code>BUILDER_TYPE</code>	<code>application/json</code>

Since the provider will return a JSON stream to Mediator, but the consumer expects to receive the REST Service in XML output format, you must set the `MESSAGE_TYPE` to `application/xml`. To do this, write and invoke a `webMethods IS REST Service` in the Virtual REST Service's **Invoke webMethods IS REST Service** action. The IS REST Service should include the following predefined context variable set to this value:

Context Variable	Value
<code>MESSAGE_TYPE</code>	<code>application/xml</code>

Characteristics of the Mapped and Badgerfish JSON Conventions

The open source library that Axis2 uses to support JSON is called Jettison. The Jettison library supports two formats of JSON: Mapped JSON and Badgerfish. Both are syntactically correct from a JSON perspective. However, for Axis Free Mediation, we only support Mapped JSON.

Note:

An important difference between the two is that the Mapped convention returns a REST Service fault if a Virtual REST Service is configured for `application/json` (Mapped convention) and it encounters a message that has namespaces, while the Badgerfish convention attempts to avoid losing any meaning encoded in XML by preserving namespace declarations. The Axis2 JSON library `MessageFormatter` will complain if Mediator attempts to transform an XML response that contains namespace declarations. So, either make sure that the requests do not include namespaces, or else set the `MESSAGE_TYPE` to `application/json/badgerfish` instead of setting it to `application/json`.

Other characteristics include the following:

Mapped JSON Convention

1. An element with no characters or child elements is represented by:

```
{ "element" : "" }
```

2. No namespaces declarations are ever written.

Note:

The Badgerfish convention does allow namespaces. If a client sends a request that contains XML namespaces, you need to bridge to the Badgerfish convention. To do this, in the Virtual

REST Service's **Set Headers** action, set the parameter **Set Headers** to **Header** and specify the Name as Content-Type and the Value as application/json/badgerfish. Doing this will override the existing Content-Type that will be sent to the Native provider.

- An element with multiple child elements of the same name is represented by an array.

Simple case:

```
<price>10.00</price>
{ "acme.price" : { "10.00" } }
```

Array case:

```
<root><child>test</child><child>test</child></root>
{ "root" : { "child" : [ "test", "test" ] } }
```

- The XML attributes for a message are prefixed with @ when a message is serialized (same as Badgerfish).

Badgerfish Convention

This convention is used to provide the means to translate between XML and JSON without losing any data (that is, namespaces).

- Element names become object properties.
- Text content of elements goes in the \$ property of an object.
- Nested elements become nested properties.
- Multiple elements at the same level become array elements.
- Attributes go in properties whose names begin with @.
- Active namespaces for an element go in the element's @xmlns property.
- The default namespace URI goes in @xmlns.\$.
- Other namespaces go in other properties of @xmlns.
- Elements with namespace prefixes become object properties, too.

Simple example:

```
<price xmlns="http://acme.com">10.00</price>
{ "price": { "@xmlns": { "$": "http://acme.com" }, "$1": "10.00" } }
```

A more complex example:

```
<alice xmlns="http://some-namespace"
  xmlns:charlie="http://another-namespace">
  <bob>david</bob>
  <charlie:edgar>frank</charlie:edgar>
</alice>

{ "alice" : { "bob" : { "$" : "david" , "@xmlns" :
{"charlie" : "http://another-namespace" , "$" : "http://some-namespace"} } } ,
  "charlie:edgar" : { "$" : "frank" , "@xmlns" :
```

```
{ "charlie": "http://another-namespace", "$": "http://some-namespace" },
"@xmlns" : { "charlie" : "http://another-namespace", "$": "http://some-namespace" } }
```

Multiple Root Nodes in JSON REST Services

With REST Virtual REST Services, when working with requests and responses of the Content-Type application/json, the message content can contain one or more root nodes. For example, a message might have the two root nodes { "firstName": "John", "lastName": "Smith" }. Note the following points for messages with multiple root nodes:

- The XSLT provided in the Request/Response Processing step should have the XPath start with //, for example //firstName. This is because during the processing of the JSON content, Mediator will wrap the given content with system-defined elements.
- webMethods IS REST Services that you invoke in the Request/Response Processing step will contain the JSON content in the variable called JSONRESTContentString. You can update the content in the IS REST Service and put the updated content into the pipeline's input variable UpdatedJSONRESTContentString, which will be sent to the Native REST Service.

Handling Virtual REST Services with Multiple Resources

The enhanced REST framework of CentraSite Business UI allows you to explicitly define multiple resources for a RESTful REST Service. Each resource within the REST Service exposes a unique URI for performing the CRUD operations on the resource.

CentraSite provides the resource path tokenizer to support the multiple resources handling at run-time. The tokenizer for substituting a resource path is automatically appended to the REST Service's endpoint (base URL) by a path variable \${sys:resource_path}. When there are multiple resources defined for an REST Service, at run time, Mediator replaces the resource path tokenizer with the appropriate resource path that is defined for the particular resource call.

Important:

Beginning with version 9.7, CentraSite supports the multi-resource handling of the REST interface at run-time (in contrast, earlier versions of CentraSite supported a single resource handling). Note that the enhanced REST interface that is implemented by current version of CentraSite is not compatible with the interface that was implemented by previous versions of Mediator prior to version 9.7. However, if you still attempt to publish a Virtual REST Service with multiple resources to an earlier version of Mediator instance, Mediator throws an exception message.

Let's now learn about the usage of resource path tokenizers in CentraSite by looking at a couple of examples using our sample phone store REST Service.

Consider you have a plain REST Service PhoneStoreREST Service with a defined set of resources for performing the CRUD operations.

Assume you have the PhoneStore REST Service with the following configuration details:

- Base URL:

```
http://www.phonestore.com/REST Service
```

- Resource:

```
phones
```

- Resource URI:

```
/phones
```

- Native Endpoint:

```
http://www.phonestore.com/REST Service/phones
```

Beginning with version 9.7, on a Virtual copy of this REST Service, CentraSite appends the resource path tokenizer `${sys:resource_path}` to represent its Route to endpoint like this:

```
http://www.phonestore.com/REST Service/${sys:resource_path}
```

In the aspect of this resource path tokenizer, there are two principal scenarios, when consumers attempt to model Virtual REST Services with multiple resources.

Scenario A

Consider you have a Virtual REST Service created using the current version of CentraSite, this REST Service will have the Route to endpoint of the Straight-Through/Content-Based/Content-Based Routing action, by default, appended with a resource path tokenizer. The Route to endpoint assigned by CentraSite has the format `<base-url>/${sys:resource_path}`, where the `resource_path` is the same as the original resource URI defined in the REST Service's first resource definition.

Whenever you try to add a new resource to this REST Service, the endpoint is automatically assigned and sent to Mediator for processing inside a Virtual REST Service Definition (VSD) in the prescribed format. At run time, when a HTTP request is received for the resource, that request will be sent to the endpoint dynamically substituted with the path variable. This dynamic substitution of the endpoint indicates that the REST interface is enhanced to support multiple resources.

Scenario B

Consider you have a Virtual REST Service created in versions of CentraSite prior to 9.7, this REST Service will continue to exhibit the old REST behavior, that is, it will continue to send the HTTP requests to the Native REST Service endpoint using the resource URI defined in the earlier version.

Now consider our sample PhoneStore REST Service with two different formats of resource URIs, say, `/phones`, `/invoke`

Now, whenever you try to add a new resource to this REST Service, CentraSite performs an internal validation of the existing resource URI. Depending on the validation, CentraSite handles the HTTP request in the following way:

- **Endpoint exactly ends with the existing resource URI** - Resource URI `/phones`. In this case, the URI is automatically substituted with the tokenizer. This ensures that the Mediator processes the HTTP request and routes the request to the appropriate Native REST Service endpoint for the requested resource.
- **Endpoint does not exactly end with the existing resource URI** - Resource URI `/invoke`. In this case, the URI is not substituted with the tokenizer and eventually results in a failure alert.

Now, based on the routing configuration for the Native REST Service, you have the following workaround options:

Straight-Through Routing Action - Straight-Through Routing Action

Workaround Option 1

Modify the Straight-Through routing action of the Native REST Service.

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

A list of defined assets in CentraSite (for which you have the View permission) is displayed in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, REST Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **REST Service**.
 - c. Click **OK**.
5. In the displayed list of REST Services, click the REST Service you want to Virtualize.

The REST Service Details page is displayed. The Actions bar displays a set of actions that are available for working with the REST Service.
6. On the Actions bar of the REST Service, click the **Virtualize** icon.
7. In the **Virtualize <REST Service_Name> (Step 1 of 3)** wizard, select the Virtual REST Service alias and its endpoint for reconfiguration.
8. Click **Next**.
9. In the **Virtualize <REST Service_Name> (Step 2 of 3)** wizard, locate the **Straight-Through** routing action in the **Message Flow** section.
10. Hover over the action name, and click the **Configure** icon next to the action name.

This opens the **Straight-Through Routing** dialog box.
11. In the **Route to** endpoint field, modify the existing endpoint. Append the resource path tokenizer `${sys:resource_path}` with the base URL in the format `<base-url>/${sys:resource_path}`.

Thus, for our sample endpoint:

```
http://www.phonestore.com/REST Service/phones
```

The modified endpoint should be:

```
http://www.phonestore.com/REST Service/${sys:resource_path}
```

12. Click **OK** and save the modified REST Service.
13. If you choose to use this option, in addition to the above steps in CentraSite Business UI, you must consider one of the following:
 - Modify the Native REST Service implementation to reconfigure the endpoint and resource URI.
 - In the details page of the Native REST Service, edit the resource URI to exactly match with the endpoint. Thus for our example, modify the resource URI to read /phones. In addition, you must inform the clients of changes that have been made to the endpoint and resource URI.

Workaround Option 2

Convert the Straight-Through Routing action of the Native REST Service to either a Context-Based Routing or Content-Based Routing action.

➤ To convert the Straight-Through routing action

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

A list of defined assets in CentraSite (for which you have the View permission) is displayed in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, REST Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **REST Service**.
 - c. Click **OK**.
5. In the displayed list of REST Services, click the REST Service you want to Virtualize.

The REST Service Details page is displayed. The Actions bar displays a set of actions that are available for working with the REST Service.
6. On the Actions bar of the REST Service, click the **Virtualize** icon.

7. In the **Virtualize <REST Service_Name> (Step 1 of 3)** wizard, select the Virtual REST Service alias and its endpoint for reconfiguration.
8. Click **Next**.
9. In the **Virtualize <REST Service_Name> (Step 2 of 3)** wizard, locate the **Straight-Through Routing** action in the **Message Flow** section.
10. Hover over the action name, and click the **Delete** icon next to the action name.
11. Drag and drop the **Content-Based Routing** action or the **Context-Based Routing** action from the **Policy Actions** accordion to the **Message Flow** section.
12. Configure the **Content-Based Routing** action or the **Context-Based Routing** action as described later in this topic.

Content-Based Routing Action

Given the Content-Based Routing action, specify a custom routing rule. Perform the following steps:

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

A list of defined assets in CentraSite (for which you have the View permission) is displayed in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, REST Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **REST Service**.
 - c. Click **OK**.
5. In the displayed list of REST Services, click the REST Service you want to Virtualize.

The REST Service Details page is displayed. The Actions bar displays a set of actions that are available for working with the REST Service.
6. On the Actions bar of the REST Service, click the **Virtualize** icon.
7. In the **Virtualize <REST Service_Name> (Step 1 of 3)** wizard, select the Virtual REST Service alias and its endpoint for reconfiguration.
8. Click **Next**.

9. In the **Virtualize <REST Service_Name> (Step 2 of 3)** wizard, locate the **Straight-Through Routing** action in the **Message Flow** section.
10. Hover over the action name, and click the **Configure** icon next to the action name. This opens the **Content-Based Routing** dialog box.
11. In the **Default Route to** endpoint field, modify the existing endpoint. Append the resource path tokenizer `${sys:resource_path}` with the base URL in the format `<base-url>/${sys:resource_path}`.

Thus, for our sample endpoint:

```
http://www.phonestore.com/REST Service/phones
```

The modified endpoint should be:

```
http://www.phonestore.com/REST Service/${sys:resource_path}
```

12. Click the **Add Routing Rule** button. In the **Routing Rule** dialog box, complete the following fields:
 - In the **XPath Expression** field, specify an XPath expression unique to the resource.
 - Add a namespace value in the **Prefix** and **URI** fields.
 - In the **Route To** field, specify the Native REST Service endpoint.

In our sample,

```
http://www.phonestore.com/REST Service/phones
```

- Click **OK** to save the new routing rule.

13. Click **OK** and save the modified REST Service.

Context-Based Routing Action

Given the Context-Based Routing action, specify a custom routing rule. Perform the following steps:

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

A list of defined assets in CentraSite (for which you have the View permission) is displayed in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, REST Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:

- a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **REST Service**.
 - c. Click **OK**.
5. In the displayed list of REST Services, click the REST Service you want to Virtualize.
- The REST Service Details page is displayed. The Actions bar displays a set of actions that are available for working with the REST Service.
6. On the Actions bar of the REST Service, click the **Virtualize** icon.
 7. In the **Virtualize <REST Service_Name> (Step 1 of 3)** wizard, select the Virtual REST Service alias and its endpoint for reconfiguration.
 8. Click **Next**.
 9. In the **Virtualize <REST Service_Name> (Step 2 of 3)** wizard, locate the **Straight-Through Routing** action in the **Message Flow** section.
 10. Hover over the action name, and click the **Configure** icon next to the action name. This opens the **Context-Based Routing** dialog box.
 11. In the **Default Route to** endpoint field, modify the existing endpoint. Append the resource path tokenizer `${sys:resource_path}` with the base URL in the format `<base-url>/${sys:resource_path}`.

Thus, for our sample endpoint:

```
http://www.phonestore.com/REST Service/phones
```

The modified endpoint should be:

```
http://www.phonestore.com/REST Service/${sys:resource_path}
```

12. Click the **Add Routing Rule** button and provide the following information:
 - Type a name for the new routing rule.
 - For the **Variable** field, select **Predefined Context Variable** from the drop-down list.
 - In the **Condition** section of the dialog box, perform the following steps.
 - Select `String` in the first **Data Type** field.
 - Select the `Virtual REST Service Operation` in the second **Predefined Context** field.
 - Specify the `Equal To` operator in the third field.
 - Specify the exact resource name in the **Variable Value** field. In our example, `invoke`.
 - In the **Route To** field, specify the Native REST Service endpoint.

In our sample,

```
http://www.phonestore.com/REST Service/phones
```

- Click **OK** to save the new routing rule.

13. Click **OK** and save the modified REST Service.

webMethods IS Services in Virtual Services

A webMethods Integration Server (IS) service is a user-defined Integration Server flow service that you can invoke in:

- **Request Processing step:** Pre-process the request message before the request is submitted to a Native Service.
- **Response Processing step:** Pre-process the response message from a Native Service before the request is returned to the consuming application.

A webMethods IS service must be running on the same Integration Server as webMethods Mediator. It can call out a C++ or Java or .NET function. It can also call other Integration Server services to manipulate the SOAP message.

The input pipeline for a webMethods IS service should have the following input variables:

- **proxy.name:** This is the name of the virtual service.
- **SOAPEnvelope:** Contains the SOAP envelope. This is of the Java type `org.apache.axiom.soap.SOAPEnvelope`.
- **EnvelopeString:** Contains the SOAP envelope as a string.

Limitation: **EnvelopeString** will *not* be sent to IS services if a request uses the MTOM SOAP Optimization Method and if the Integration Server property `watt.server.SOAP.MTOMStreaming.enable` is set to true.

- **MessageContext:** Mediator places a MessageContext variable into the pipeline before executing the webMethods IS service call. MessageContext is of the Java type `org.apache.axis2.context.MessageContext`.

Integration Server users can use the Axis2 MessageContext object to manipulate the incoming SOAP request. The Integration Server provides built-in services (that is, the `pub.soap.*` services) to work with the MessageContext object to get, set, or modify the SOAP body, header, properties, and so on. Integration Server users should use these services to extract the information they need from the MessageContext to build the necessary business logic. Users do not need to understand Axis2 or Axiom (the xml object model based on StAX) to work with the SOAP request, because if they are familiar with the Integration Server `pub.soap` services, they can accomplish most of the tasks. For more information about these related Integration Server services, see the *webMethods Integration Server Built-In Services Reference*.

- **JSONRESTContentString:** Will appear only for REST services with the Content-Type `application/json` or `application/json/badgerfish`.
- **UpdatedJSONRESTContentString:** Will appear only for REST services with the Content-Type `application/json` or `application/json/badgerfish`.

You can use the following constructs in a webMethods IS service:

- Predefined or custom context variables.
- The Security API provided by Mediator (for Web services only).

Using the Security API in webMethods IS Services

Note:

This API is for Web services only.

Mediator provides Java services that you can use to support WS-Security functionality in a webMethods IS service that you invoke in the Request Processing step.

pub.mediator.security.ws:AddUsernameToken

Adds the WS-Username Token 1.0 and 1.1 to the request. This service includes the following input parameters:

Note:

For reasons of legibility some of the examples below contain break lines and might not work when pasted into applications or command line tools.

In the parameter descriptions, the data type is listed first, followed by the Java type in parenthesis, for example, "Object (org.apache.axis2.context.MessageContext)".

Input Parameters

<i>username</i>	<p>String (String). (Required). The value that will be added as the Username element in the token.</p> <p>The default value is: ""</p>
<i>MessageContext</i>	<p>Object (org.apache.axis2.context.MessageContext) (Required) Mediator will place a <i>MessageContext</i> variable into the pipeline before executing the webMethods IS service call.</p> <p>The default value is: org.apache.axis2.context.MessageContext instance</p>
<i>password</i>	<p>String (String). (Optional). The password for the token. You must specify <i>password</i> if the <i>passwordType</i> is set to either TEXT or DIGEST.</p> <p>The default value is: ""</p>
<i>passwordType</i>	<p>String (String). (Optional). Specifies how the password will be added in the token. Specify one of the following values:</p> <ul style="list-style-type: none"> ■ NONE: The password will not be added. ■ TEXT: The password is added in plain text. ■ DIGEST: The password is added in digested form (as specified in the UsernameToken profile). <p>The default value is: NONE</p>

<i>addNonce</i>	<p>Boolean (Boolean). (Optional). Specifies whether the Nonce element will be added to the token.</p> <p>The default value is: <code>false</code></p>
<i>addCreated</i>	<p>Boolean (Boolean). (Optional). Specifies whether the Created element will be added to the token</p> <p>The default value is: <code>false</code></p>
<i>salt</i>	<p>byte[] (byte[]). (Optional). The value for the /wsse11:UsernameToken/wsse:Salt element. Its value is a 128 bit number serialized as xs:base64Binary.</p> <p>The default value is: <code>null</code></p>
<i>iteration</i>	<p>int (Integer). (Optional). Indicates the number of times the hashing operation is repeated when deriving the key. It is expressed as a xs:unsignedInteger value. If it is not present, a value of 1000 is used for the iteration count.</p> <p>The default value is: <code>1000</code></p>
<i>useMac</i>	<p>Boolean (Boolean). (Optional). Indicates if the derived key will be used as a Message Authentication Code (MAC) or as a symmetric key for encryption.</p> <p>The default value is: <code>false</code></p>
<i>useBasic AuthCredentials</i>	<p>Boolean (Boolean). (Optional). If this parameter is set to <code>true</code>, Mediator will try to use the username and password from the "Authorization" HTTP header. In this case the 'username' and 'password' fields need not be specified.</p> <p>The default value is: <code>false</code></p>
<i>actor</i>	<p>String (String). (Optional). Indicates the value of the SOAP actor attribute if a new security header is being added to the SOAP request. If the request already has a security header with the actor specified in it, then this value will not overwrite it.</p> <p>The default value is: <code>""</code></p>
<i>mustUnderstand</i>	<p>Boolean (Boolean). (Optional). Specifies whether the security header will have the <code>mustUnderstand</code> attribute set to 0 or 1 (<code>false</code> / <code>true</code>). If the security header already has this attribute set, this value will not overwrite it.</p> <p>The default value is: <code>false</code></p>

pub.mediator.security.ws:AddX509Token

Adds a X.509 certificate (or certificate chain) as a BinarySecurityToken (BST) element in the outbound SOAP request. This service includes the following input parameters:

Note:

For reasons of legibility some of the examples below contain break lines and may not work when pasted into applications or command line tools.

In the parameter descriptions, the data type is listed first, followed by the Java type in parenthesis, for example, “Object (org.apache.axis2.context.MessageContext)”.

Input Parameters

<i>MessageContext</i>	Object (org.apache.axis2.context.MessageContext). (Required). Mediator will place a <i>MessageContext</i> variable into the pipeline before executing the webMethods IS service call. The default value is: org.apache.axis2.context.MessageContext instance
<i>keystoreFile</i>	String (String). (Required). The absolute path to a keystore file on the system where Mediator is running. The default value is: ""
<i>keystorePassword</i>	String (String). (Required). The password for the keystore. The default value is: ""
<i>keystoreType</i>	String (String). (Optional). The type of keystore represented by the file (can be JKS, JCEKS, or PKCS12). The default value is: JKS
<i>keyAlias</i>	String (String). (Required). The key alias whose X509 certificate will be sent in the soap request as a BST. The default value is: ""
<i>useCertificatePath</i>	Boolean (Boolean). (Optional). If set to true will use the entire certificate chain represented by the key alias instead of just a single certificate. The default value is: false
<i>actor</i>	String (String). (Optional). Indicates the value of the SOAP actor attribute if a new security header is being added to the SOAP request. If the request already has a security header with the actor specified in it, then this value will not overwrite it. The default value is: ""
<i>mustUnderstand</i>	Boolean (Boolean). (Optional). Specifies whether the security header will have the mustUnderstand attribute set to 0 or 1 (false / true). If the security header already has this attribute set, then this value will not overwrite it. The default value is: false

pub.mediator.security.ws:AddSamlSenderVouchesToken

This service enables a Security Token Service (STS) client to send a WS-Trust request to a configured STS to obtain a SAML v1/v2 assertion. For the details about configuring Mediator to act as an STS client, see *Administering webMethods Mediator*.

This service adds the obtained SAML assertion to the original request that is sent by the client to the native service, and includes the following parameters:

Note:

For reasons of legibility some of the examples below contain break lines and may not work when pasted into applications or command line tools.

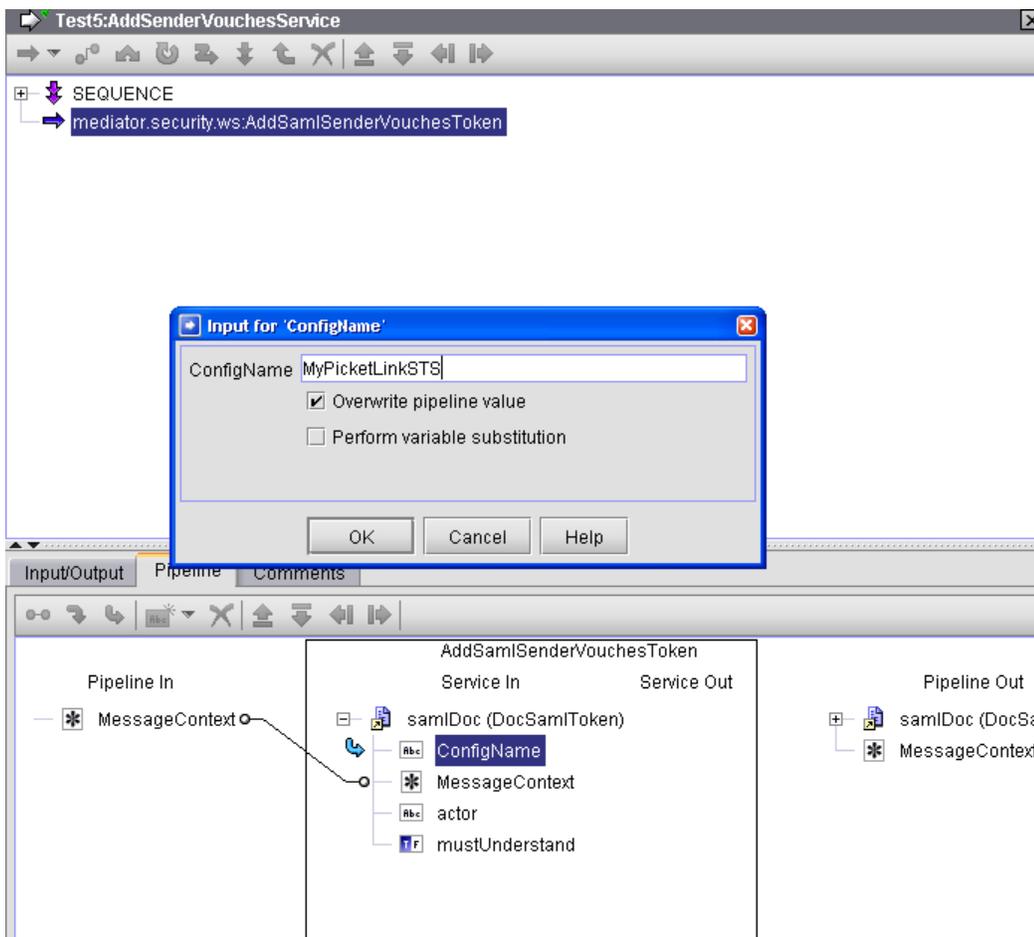
In the parameter descriptions, the data type is listed first, followed by the Java type in parenthesis, for example, "Object (org.apache.axis2.context.MessageContext)".

Input Parameters

<i>ConfigName</i>	String (String). (Required). References a previously configured STS configuration name. The default value is: ""
<i>MessageContext</i>	Object (org.apache.axis2.context.MessageContext). (Required). Mediator will place a <i>MessageContext</i> variable into the pipeline before executing the webMethods IS service call. The default value is: org.apache.axis2.context.MessageContext instance
<i>addTimeStamp</i>	Boolean (Boolean). (Optional). Adds a Timestamp element (with the duration specified in <i>timeToLive</i>) to the WS-Security header of the request and includes it in the signature. (The other items that are signed are the body and SAML assertion.) The default value is: false
<i>timeToLive</i>	Integer (Integer). (Optional). If <i>addTimeStamp</i> is true, <i>timeToLive</i> specifies the duration (in seconds) for which the request is valid. The default value is: 300 (5 minutes)
<i>actor</i>	String (String). (Optional). Indicates the value of the SOAP actor attribute if a new security header is being added to the SOAP request. If the request already has a security header with the actor specified in it, then this value will not overwrite it. The default value is: ""
<i>mustUnderstand</i>	Boolean (Boolean). (Optional). Specifies whether the security header will have the mustUnderstand attribute set to 0 or 1 (false / true). If the security header already has this attribute set, then this value will not overwrite it. The default value is: false

Example of using AddSamlSenderVouchesToken

The sample service shown below is configured by providing the `MessageContext` and `ConfigName` parameters. The value of `ConfigName` must be the name of a previously configured STS name, which is configured on the Mediator Configuration page.



pub.mediator.security.ws:AddTimestamp

Adds a timestamp to the outbound SOAP request WS-Security header. This service includes the following input parameters:

Note:

For reasons of legibility some of the examples below contain break lines and may not work when pasted into applications or command line tools.

In the parameter descriptions, the data type is listed first, followed by the Java type in parenthesis, for example, “Object (org.apache.axis2.context.MessageContext)”.

Input Parameters

timeToLive

Integer (Integer). (Optional). Specifies the duration (in seconds) for which the request is valid.

The default value is: 300 (5 minutes)

<i>signTimestamp</i>	<p>Boolean (Boolean). (Optional). Indicates whether the generated timestamp must be signed by Mediator using the configured keystore and signing alias.</p> <p>Note: For <i>signTimestamp</i> to work, you must ensure that a valid IS keystore and signing alias are configured in Mediator. For details, see <i>Administering webMethods Mediator</i>.</p> <p>The default value is: <code>false</code></p>
<i>useMillisecondPrecision</i>	<p>Boolean (Boolean). (Optional). Indicates whether the generated timestamp must have millisecond precision.</p> <p>The default value is: <code>true</code></p>
<i>MessageContext</i>	<p>Object (org.apache.axis2.context.MessageContext). (Required). Mediator will place a <i>MessageContext</i> variable into the pipeline before executing the webMethods IS service call.</p> <p>The default value is: <code>org.apache.axis2.context.MessageContext</code> instance</p>
<i>actor</i>	<p>String (String). (Optional). Indicates the value of the SOAP actor attribute if a new security header is being added to the SOAP request. If the request already has a security header with the actor specified in it, then this value will not overwrite it.</p> <p>The default value is: <code>""</code></p>
<i>mustUnderstand</i>	<p>Boolean (Boolean). (Optional). Specifies whether the security header will have the <code>mustUnderstand</code> attribute set to 0 or 1 (false / true). If the security header already has this attribute set, then this value will not overwrite it.</p> <p>The default value is: <code>false</code></p>

pub.mediator.addressing:AddWSAddressingHeaders

Adds WS-Addressing headers to a SOAP request sent by the client before Mediator forwards the request to the native service.

This service includes the following input parameters:

Note:

For reasons of legibility some of the examples below contain break lines and may not work when pasted into applications or command line tools.

In the parameter descriptions, the data type is listed first, followed by the Java type in parenthesis, for example, "Object (org.apache.axis2.context.MessageContext)".

Input Parameters

isVersion Submission **Boolean (Boolean)**. (Optional). The WS-Addressing version that should be used.

- If true, the WS-Addressing submission namespace will be used.

```
http://schemas.xmlsoap.org/ws/2004/08/addressing
```

- If false, the Final specification namespace will be used.

```
http://www.w3.org/2005/08/addressing
```

The default value is: `false`

To **String (String)**. (Optional). This value corresponds to the `/wsa:To` addressing header. You must specify a value that corresponds to the destination of the request message.

If this value is not specified, the default value depends on the *isVersionSubmission* property value. One of the following anonymous EPR values will be sent:

- If *isVersionSubmission* is set to `true`, the anonymous EPR value is:

```
http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
```

- If *isVersionSubmission* is set to `false`, the anonymous EPR is:

```
http://www.w3.org/2005/08/addressing/anonymous
```

From **String (String)**. (Optional). This value corresponds to the `/wsa:From` addressing header and refers to the source of the message.

The default value is: `""`

Action **String (String)**. (Optional). This value corresponds to the `/wsa:Action` addressing header. By default, this property has the same value as the operation on the virtual service being invoked (which will usually correspond to the same operation on the native service). But the user can specify a different value corresponding to the native service being called.

The default value is: URI identifying input operation corresponding to a WSDL port type being called on the virtual service

MessageContext **Object (org.apache.axis2.context.MessageContext)**. (Required). Mediator will place a *MessageContext* variable into the pipeline before executing the webMethods IS service call.

The default value is: `org.apache.axis2.context.MessageContext` instance

actor **String (String)**. (Optional). Indicates the value of the SOAP actor attribute if a new security header is being added to the SOAP request. If the request already has a security header with the actor specified in it, then this value will not overwrite it.

The default value is: ""

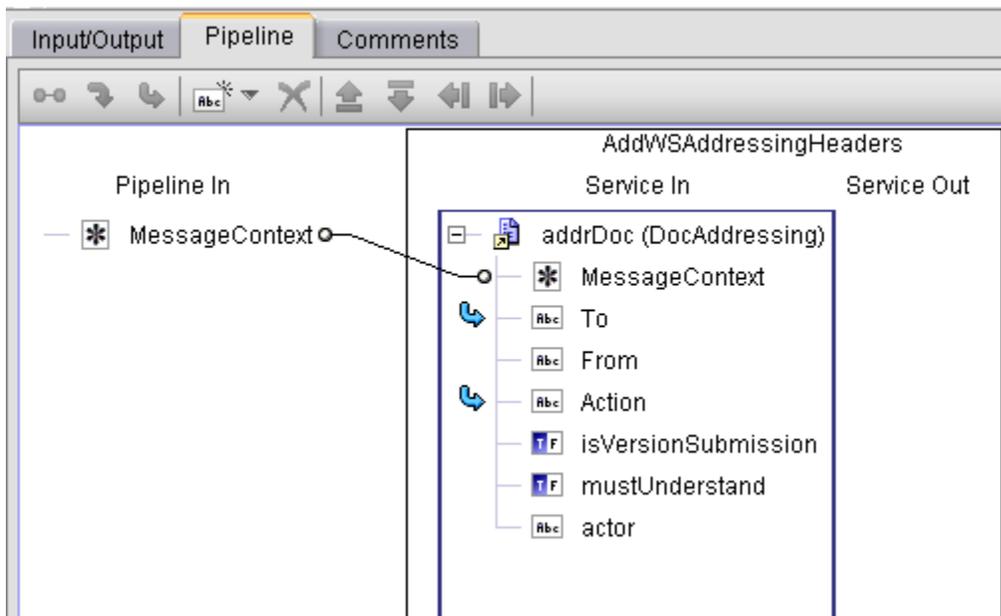
mustUnderstand

Boolean (Boolean). (Optional). Specifies whether the security header will have the mustUnderstand attribute set to 0 or 1 (false / true). If the security header already has this attribute set, this value will not overwrite it.

The default value is: false

Example of using AddWSAddressingHeaders

The sample service shown below is configured by providing the MessageContext parameter.



Context Variables in Virtual Services

Mediator provides predefined context variables and you can declare your own custom context variables. You can use both predefined and custom context variables when you configure various processing steps of a virtual service. Specifically, you can use them:

- In a webMethods IS service that you create and invoke during the **Request Processing** step or the **Response Processing** step.
- In a routing rule that you create in the **Context-Based Routing Protocols** step.

The Predefined Context Variables

You can use the predefined context variables listed below. Any context variable state defined during the inbound request processing steps will still be available during the outbound response processing steps.

Note:

Here are some general guidelines when using context variables:

- To set, get, or remove the predefined context variables, use [“The API for Context Variables” on page 1162](#) provided by Mediator.
- You do not need to declare the predefined context variables. If you attempt to declare an existing predefined context variable, an error message will be displayed.

Context Variable Display Name and Variable Name	Description
Average Response AVG_SUCCESS_TIME	The average amount of time it took the service to complete all invocations in the current interval. This is measured from the moment Mediator receives the request until the moment it returns the response to the caller. Note: By default, Average Response Time does not include metrics for failed invocations. You can include metrics for failed invocations by setting the <code>pg.PgMetricsFormatter.includeFaults</code> parameter to true. For details about advanced settings, see <i>Administering webMethods Mediator</i> .
Client IP Address INBOUND_IP	The IP address used to send the request to Mediator.
Consumer CONSUMER_APPLICATION	The name of the consumer application accessing the service, if known.
Dynamic Routing ROUTING_ENDPOINT	Mediator takes the ROUTING_ENDPOINT value from the message context and replaces the <code>#{sys:dyn-Endpoint}</code> variable in the Route Through field of dynamic routing policy configuration.
Fault Count INTERVAL_FAULT_COUNT	The number of service faults for the interval.
Inbound Content Type MESSAGE_TYPE	A Content-Type defined in axis2.xml for a message formatter. This value must be a key in the axis2 message formatters list, since it is used to control message serialization. (The valid choices are defined as attributes of <code><messageFormatters/></code> group in the Integration Server's axis2.xml configuration.)
Inbound HTTP Method INBOUND_HTTP_METHOD	The HTTP method used by the client to send the request (GET, POST, PUT, PATCH, DELETE, Use Context Variable).

Context Variable Display Name and Variable Name	Description
Inbound Protocol INBOUND_PROTOCOL	The protocol (HTTP or HTTPS) of the request.
Maximum Response SLOWEST_SUCCESS_INVOKE	Maximum Response Time. Note: By default, Maximum Response Time does not include metrics for failed invocations. You can include metrics for failed invocations by setting the <code>pg.PgMetricsFormatter.includeFaults</code> parameter to true. For details about advanced settings, see <i>Administering webMethods Mediator</i> .
Mediator Host Name MEDIATOR_HOSTNAME	Mediator host name.
Mediator IP Address MEDIATOR_IP	Mediator IP address.
Mediator Target Name TARGET_NAME	Mediator target name.
Minimum Response FASTEST_SUCCESS_INVOKE	Minimum Response Time. Note: By default, Minimum Response Time does not include metrics for failed invocations. You can include metrics for failed invocations by setting the <code>pg.PgMetricsFormatter.includeFaults</code> parameter to true. For details about advanced settings, see <i>Administering webMethods Mediator</i> .
Outbound HTTP Method ROUTING_METHOD	The HTTP method to be sent to the native service if the inbound HTTP method is custom. Otherwise, this value will be null. For more information, see “Important Considerations when Configuring Virtual REST Services” on page 1125.
Success Count INTERVAL_SUCCESS_COUNT	The number of success counts for a given service.
Total Count INTERVAL_TOTAL_COUNT	The total number of counts for a given service.

Context Variable Display Name and Variable Name	Description
Virtual Service Name SERVICE_NAME	Virtual service name.
N/A BUILDER_TYPE	A Content Type defined in axis2.xml for a message builder. This value must be a key in the axis2 message builders list, since it is used to control building of native service response messages. (The valid choices are defined as attributes of <messageBuilders/> group in the Integration Server's axis2.xml configuration.)
N/A, no display name INBOUND_REQUEST_URI	<p>A partial reference to a virtual service (for HTTP/HTTPS only). The protocol, host and port are not part of the value. For example, if the following virtual service is invoked:</p> <pre>http://mcusawco:5555/ws/TC1</pre> <p>then the expected value of this variable would be /ws/TC1.</p> <p>For a REST or XML service, the URL might also include query string parameters. For example, if the following virtual service is invoked:</p> <pre>http://mcusawco:5555/ws/cars?vin=1234</pre> <p>the expected value of this variable would be /ws/cars?vin1234.</p> <p>This is useful to know because by the time you are able to access the request inside of Mediator, the REST request would contain a top-level element that looks like this:</p> <pre><vin>1234</vin></pre> <p>So it is not obvious from an XSLT expression or a webMethods IS service callout what part of a REST request came in as a query parameter.</p> <p>Therefore, using this variable along with INBOUND_HTTP_METHOD and INBOUND_PROTOCOL, you can determine the exact entry point URI that was used when a virtual service was invoked.</p>
N/A, no display name	The reason returned by the native service provider in the case where it produced a SOAP fault. This will

Context Variable Display Name and Variable Name	Description
	not contain Mediator errors such as security policy enforcement errors. This variable will only contain the “reason” text wrapped in a SOAP fault.
	<p>Note: When you are using this variable in Conditional Error Processing message that you are specifying in the Response Processing Step, note the following: if a request is denied due to security policy enforcement, the fault handler variable \$ERROR_MESSAGE variable would contain a native service provider error message or other error messages that result from enforced security assertions. However, \$NATIVE_PROVIDER_ERROR will be null in this case.</p>
N/A, no display name OPERATION	The virtual service operation selected to execute a request.
N/A, no display name PROTOCOL_HEADERS	Contains a map of key-value pairs in the request, where the values are typed as strings.
N/A, no display name SOAP_HEADERS	(For use in webMethods IS services only.) Contains an array of the SOAP header elements in the request. To get/set this variable, see “The API for Context Variables” on page 1162 .
N/A, no display name USER	The value defined for the Integration Server session executing the request message. If the request is not authenticated, it will use a default unprivileged account. Otherwise, it will set the Integration Server session to the user credentials used for transport security. Also, if credentials were included for message based security (for example, an X509 token was included and this certificate was mapped to an Integration Server user), then this information would override any transport security (for example, basic authentication).

Note:
When you use predefined context variables in a Conditional Error Processing message in Response Processing step, note the following:

- To reference a predefined context variable in a Conditional Error Processing message, you need to prepend a \$ symbol to the context variable name to indicate that variable’s value

should be referenced. Think of this usage as being similar to the '&' address operation for C variables. A predefined context variable expression might look like this:

```
$USER="Administrator:"
```

- The \$ reference symbol may appear in the text as needed. (for example, as a currency symbol). There is no escape concept used with this operator. That is, no special meaning is attached to two occurrences of this symbol: "\$\$".
- If no value is defined for a valid context variable reference, the string is left unmodified for that context variable.

The API for Context Variables

Mediator provides an API that you can use to:

- Set, get, declare, and remove custom context variables.
- Set and get the predefined context variables. (It is not necessary (or even legal) to declare or remove the predefined context variables.)

Mediator provides the following Java services, which are defined in the class `ISMediatorRuntimeFacade.java`:

- `pub.mediator.ctxvar:getContextVariable`
- `pub.mediator.ctxvar:setContextVariable`
- `pub.mediator.ctxvar:declareContextVariable`
- `pub.mediator.ctxvar:removeContextVariable`

Sample flow services are described in this section as well.

- Sample Flow Service: Getting a Context Variable Value
- Sample Flow Service: Setting a Context Variable Value

pub.mediator.ctxvar:getContextVariable

Use this Java service to retrieve a context variable's value and assign it to a pipeline variable. All parameter names are case-sensitive.

Parameter	Pipeline Type	Data Type	Description	Examples
MessageContext	in	Object ref	This object is inserted into the pipeline by Mediator.	N/A
varName	in	String	Context variable name (predefined or custom).	PROTOCOL_HEADERS SOAP_HEADERS mx : CUSTOM_VAR

Parameter	Pipeline Data Type	Data Type	Description	Examples
serValue	out	Object ref	Java.io.Serializable value. (Usually a string).	

Notes on getting and setting the `PROTOCOL_HEADERS` and `SOAP_HEADERS` variables

All context variable values are typed as either `string` or `int` except for the predefined context variables `PROTOCOL_HEADERS` and `SOAP_HEADERS`, which are of the type `IData`. You can set or get values for `PROTOCOL_HEADERS` and `SOAP_HEADERS` in one of two ways:

■ set or get the entire structure.

To set the entire structure, you must:

- Set the `varName` parameter in `pub.mediator.ctxvar:setContextVariable` to `PROTOCOL_HEADERS` or `SOAP_HEADERS`.
- Use the method `ISMediatorRuntimeFacade.setContextVariableValue()`.

To get the entire structure, you must:

- Set the `varName` parameter in `pub.mediator.ctxvar:getContextVariable` to `PROTOCOL_HEADERS` or `SOAP_HEADERS`.
- Use the method `ISMediatorRuntimeFacade.getContextVariableValue()`.

If `varName` is set to `PROTOCOL_HEADERS`, you will get/set the entire `IData` structure containing all of the transport headers. The key is the transport header name (for example, `Content-Type`) and the value is a `String`. The `IData` object for `PROTOCOL_HEADERS` will contain a set of string values where each `IData` string key matches the header name in the transport headers map. The set of possible keys includes the HTTP v1.1 set of headers as well as any custom key-value pairs you might have defined.

If `varName` is set to `SOAP_HEADERS`, you will get/set the entire `IData` structure containing all of the SOAP headers in the SOAP envelope. The key is the array position starting with `0`, and the value is an `Axiom OMElement` containing that SOAP header block.

Alternatively, you can set the `varName` parameter to address a specific element in the array. For example, setting it to `PROTOCOL_HEADERS[Content-Type]` would apply to the `Content-Type` transport header. Similarly, setting it to `SOAP_HEADERS[0]` would return a `String` representation of the first SOAP header block (as opposed to an `Axiom OMElement`).

■ set or get a nested value.

Set a nested value in one of the following ways:

- Set the `varName` parameter in `pub.mediator.ctxvar:setContextVariable` to `PROTOCOL_HEADERS[arrayElement]`, where `[arrayElement]` refers to a specific element. For example, `PROTOCOL_HEADERS[Content-Type]` or `SOAP_HEADERS[0]` (to indicate the first array element in the set).

- Alternatively, use the method `ISMediatorRuntimeFacade.setContextVariableValue()`. You would use this method only if you are writing a Java service and you want to access it through the Java source code.

Get a nested value in one of the following ways:

- Set the `varName` parameter in `pub.mediator.ctxvar:getContextVariable` to `PROTOCOL_HEADERS[arrayElement]`, where `[arrayElement]` refers to a specific element. For example, `PROTOCOL_HEADERS[Content-Type]` or `SOAP_HEADERS[0]` (to indicate the first array element in the set).
- Alternatively, use the method `ISMediatorRuntimeFacade.getContextVariableValue()`. You would use this method only if you are writing a Java service and you want to access it through the Java source code.

You can set or get a nested value inside `PROTOCOL_HEADERS` and `SOAP_HEADERS` through an additional `keyName`. In this case, the object reference will *not* be an `IData` object.

- For `PROTOCOL_HEADERS`, the `keyName` must match the transport header name in a case-sensitive manner (for example, `PROTOCOL_HEADERS[Content-Type]` or `PROTOCOL_HEADERS[Authorization]`). In this case, the `Serializable` value will be a string.
- For `SOAP_HEADERS`, the `keyName` must match the 0-based array element. If a request has a SOAP security header element (that is, `</wsse:Security>`), then it would be addressed as `SOAP_HEADERS[0]`. In this case, the element will be in its string format.

pub.mediator.ctxvar:setContextVariable

Use this Java service to set a value on a context variable. The pipeline variable containing the context variable value should be an object reference that implements `java.io.Serializable`. All parameter names are case-sensitive.

Parameter	Pipeline Type	Data Type	Description	Examples
<code>MessageContext</code>	in	Object ref	This object is inserted into the pipeline by Mediator.	N/A
<code>varName</code>	in	String	Context variable name (predefined or custom).	<code>PROTOCOL_HEADERS</code> <code>SOAP_HEADERS</code> <code>mx:CUSTOM_VAR</code>
<code>serValue</code>	in	Object ref	<code>Java.io.Serializable</code> value. (Usually a string).	

pub.mediator.ctxvar:declareContextVariable

Use this Java service to declare a *custom* context variable. All custom-defined context variables must be declared in a custom namespace that is identified by using the prefix `mx` (for example, `mx:CUSTOM_VARIABLE`). All parameter names are case-sensitive.

Note:

It is not legal to use this service to declare the predefined context variables; you can only declare custom variables.

Parameter	Pipeline Type	Data Type	Description
ctxVar	in	Object ref	The document type defining the context variable object. Use the ctxVar Document Type provided in the Java service <code>pub.mediator.ctxvar:ctxVar</code> and map it to this input variable. Define the name (for example, <code>mx:CUSTOM_VARIABLE</code>), the <code>schema_type</code> (string or int), and <code>isReadOnly</code> (true or false).
ctxVar	out	Object ref	The set Context variable document type.
varNameQ	out	Object ref	<code>javax.xml.namespace.QName</code> value. The QName of the variable.

Note the following:

- After declaring the context variable, you can use the `setContext` variable to set a value on the context variable.
- You do *not* need to declare the following kinds of context variables:
 - The predefined context variables provided by Mediator. If you attempt to declare an existing predefined context variable, an error will occur.
 - Any custom context variable that you define in a routing rule that you create in the context-based routing step.
- Any custom context variables that you explicitly declare in source code using the API will have a declaration scope of `SESSION`.
- Any custom context variable's state that is defined during the inbound request processing steps will still be available during the outbound response processing steps.
- All context variable values are typed as either `string` or `int` (excluding the `SOAP_HEADERS` and `PROTOCOL_HEADERS` variables, which are of the type `IData`).

- Valid names should be upper case (by convention) and must be a valid Java Identifier. In general, use alpha-numeric, \$ or _ symbols to construct these context names. Names with punctuation, whitespace or other characters will be considered invalid and will fail deployment.
- All custom context variables must be declared in a custom namespace that is identified by using an `mx` prefix (for example, `mx:CUSTOM_VARIABLE`).
- To reference a custom context variable in a flat string, you need to prepend a \$ symbol to the context variable name to indicate that variable's value should be referenced. Think of this usage as being similar to the & address operation for C variables.

An expression that references a custom context variable might look like this:

```
$mx:TAXID=1234 OR $mx:ORDER_SYSTEM_NAME="Pluto"
```

Notice that the values of the data type "int" are not enclosed in quotation marks, while the values of the data type "string" are. The quotation marks are only needed if a context variable *expression* (as opposed to a reference) is defined.

- Referencing an undefined context variable does not result in an error.
- Once a variable has been declared it cannot be declared again.

pub.mediator.ctxvar:removeContextVariable

Use this Java service to remove a *custom* context variable from a request/response session. All parameter names are case-sensitive.

Note:

Keep the following points in mind:

- It is not legal to use this service to remove any predefined context variables; you can only remove custom variables.
- Attempting to remove a non-existent context variable will *not* result in an error.

Parameter	Pipeline Type	Data Type	Description	Examples
MessageContext	in	Object ref	This object is inserted into the pipeline by Mediator.	N/A
varName	in	String	Custom context variable name.	<code>mx:CUSTOM_VAR</code>

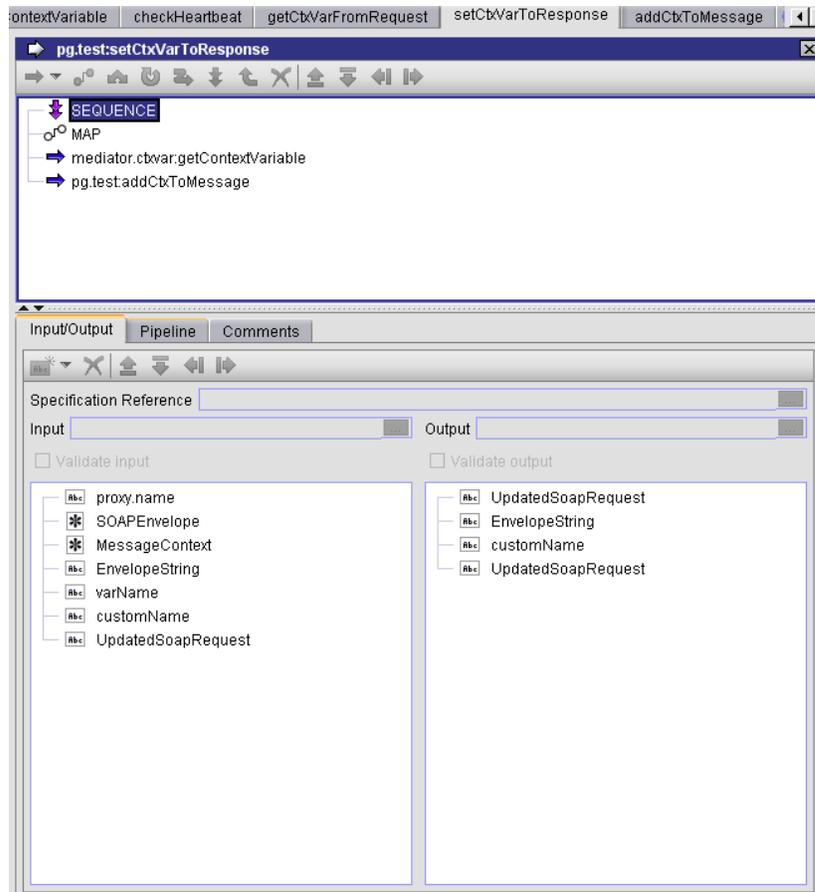
Sample Flow Service: Getting a Context Variable Value

This flow service sets the value of a custom context variable to be used in a response.

This flow service declares a pipeline variable named `customName`, which is set to the value `mx:COMP_TEST`.

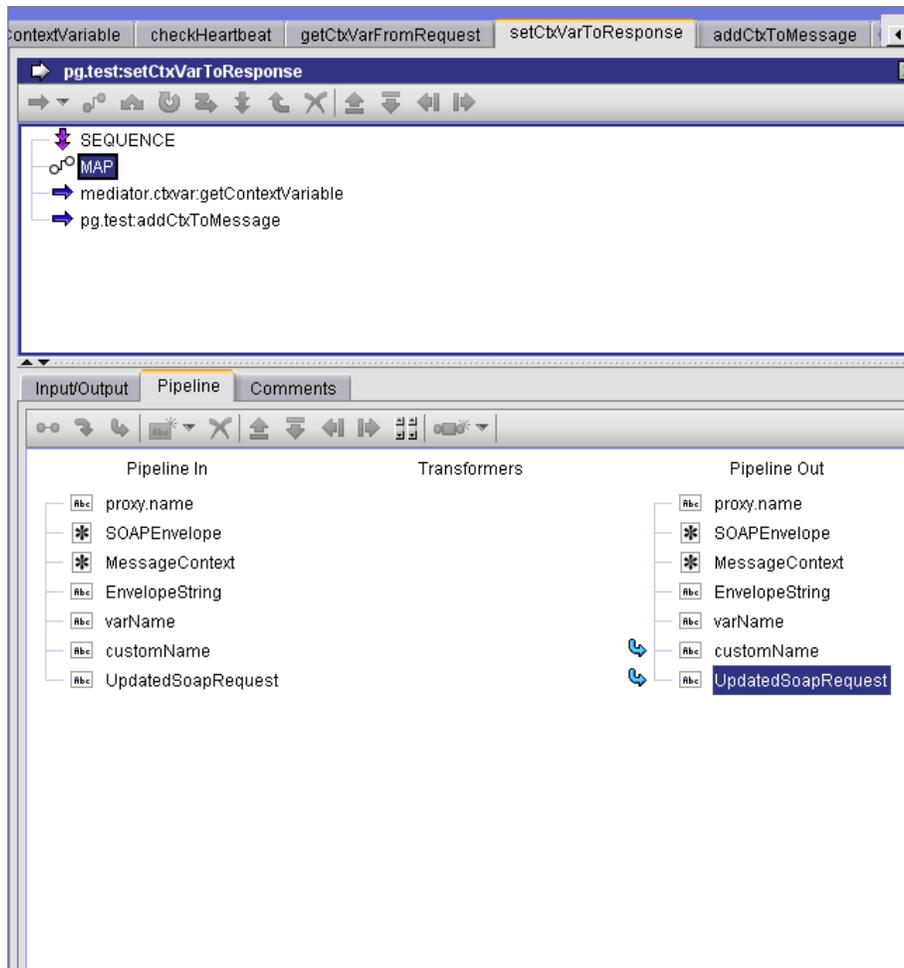
This flow service will retrieve the context variable for `customName` and create an element for its context variable value in the response message return to the consumer.

Step 1. Declaring `customName`



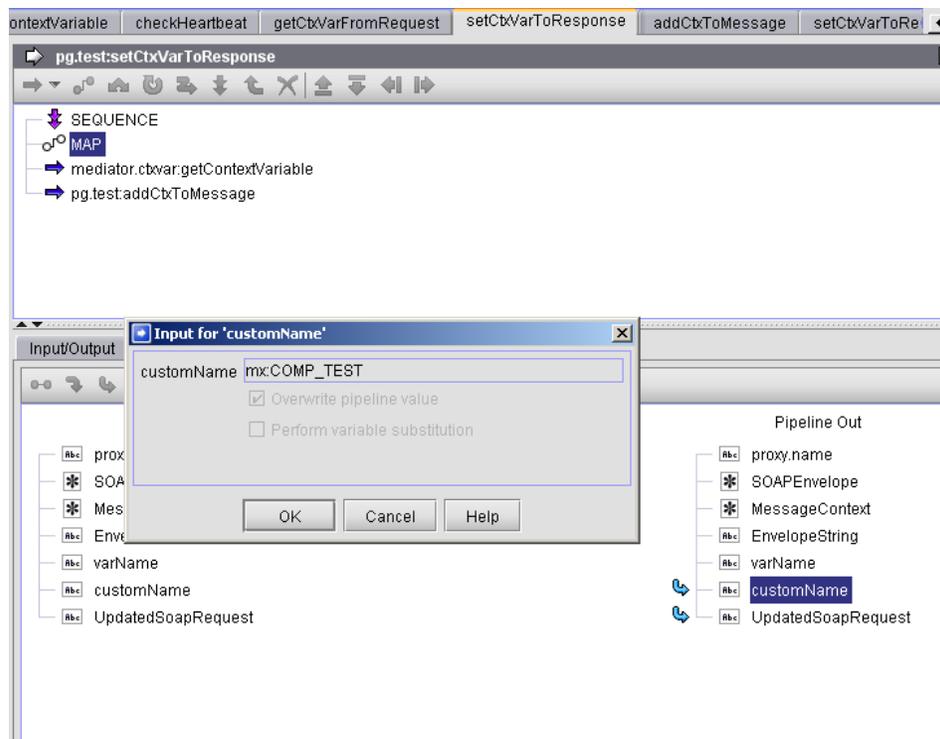
We define the `customName` variable value to be `mx:COMP_TEST` so we can use this variable to lookup the custom variable name that was seeded in the previous example.

Step 2. Setting `customName` to `mx:COMP_TEST`



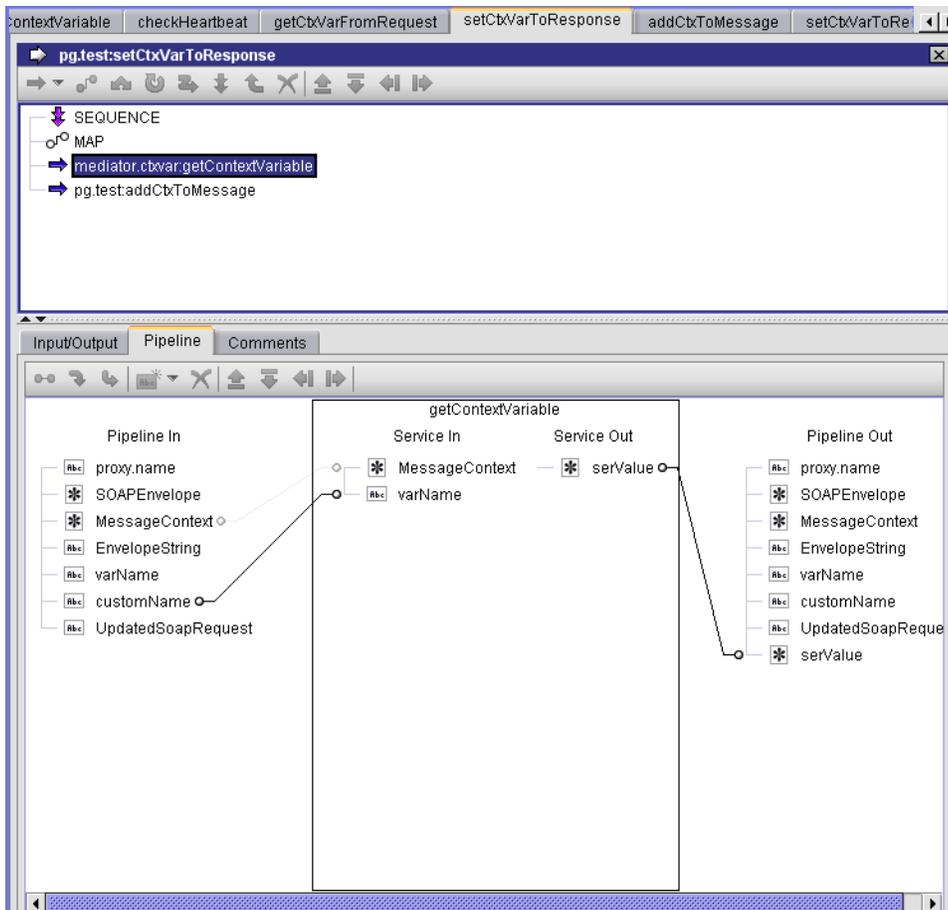
Clicking on the customName pipeline variable displays the name.

Step 3. Displaying the value of customName



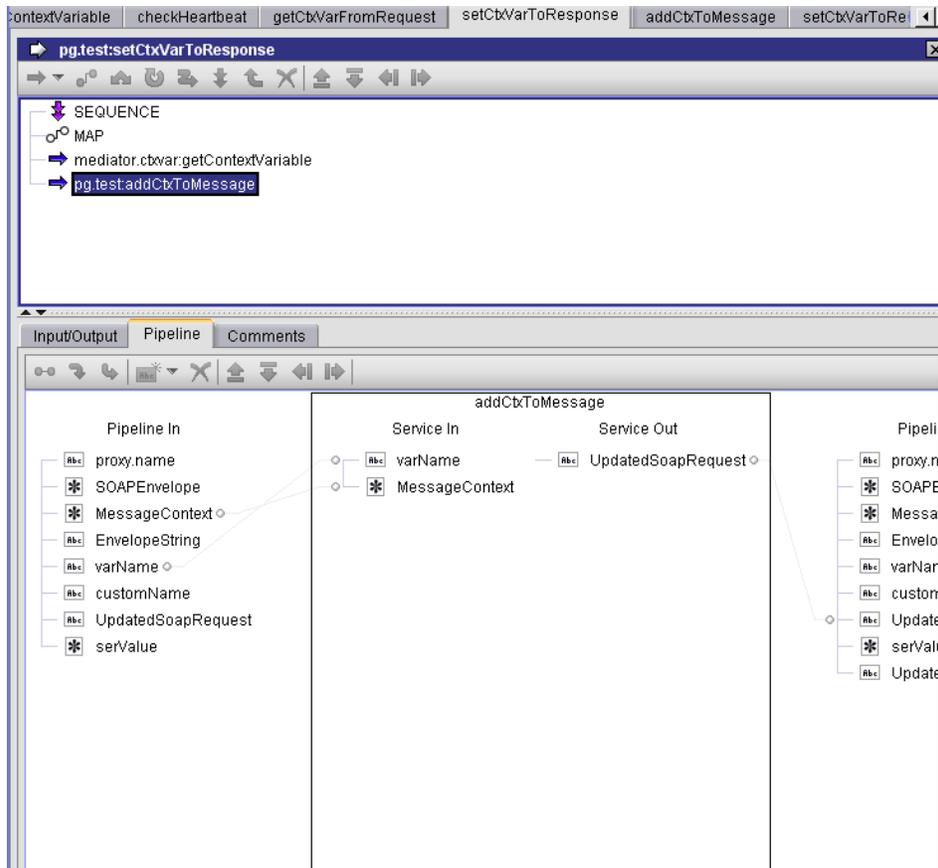
The call to `pub.mediator.ctxvar:getContextVariable` retrieves the value of the custom context variable from the context variable map.

Step 4. Calling `mediator.ctxvar:getContextVariable`



This is just a sample Java service that takes the context variable and creates a top-level element in the response message using the same name and value.

Step 5. Sample service using the context variable



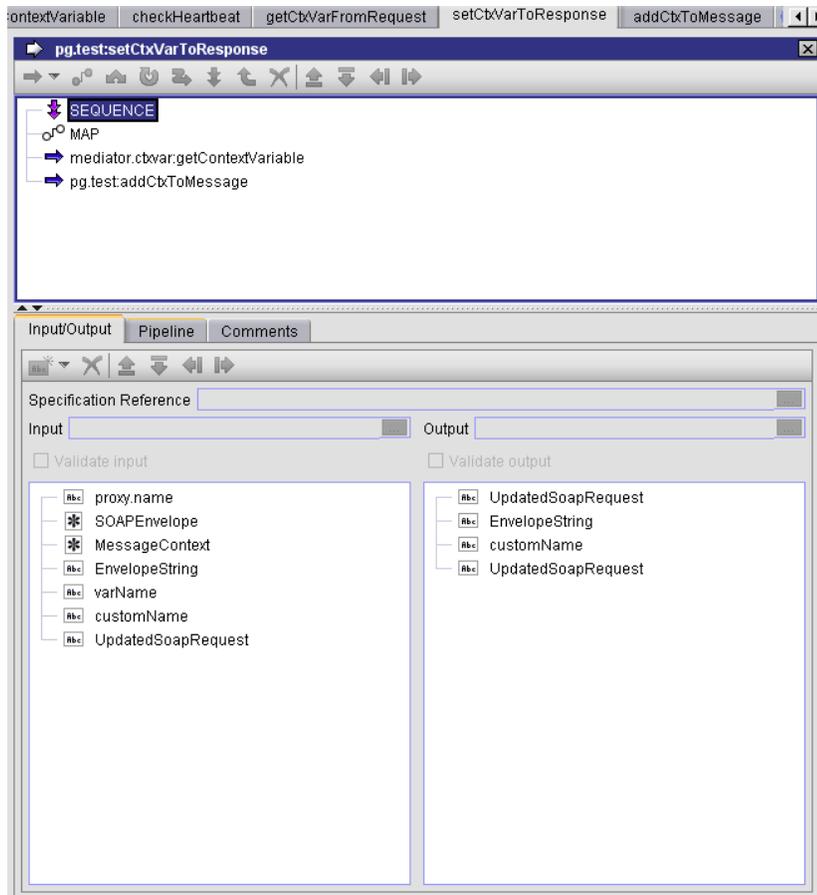
Sample Flow Service: Setting a Context Variable Value

This flow service sets the value of a custom context variable to be used in a response.

This flow service declares a pipeline variable named `customName`, which is set to the value `mx:COMP_TEST`.

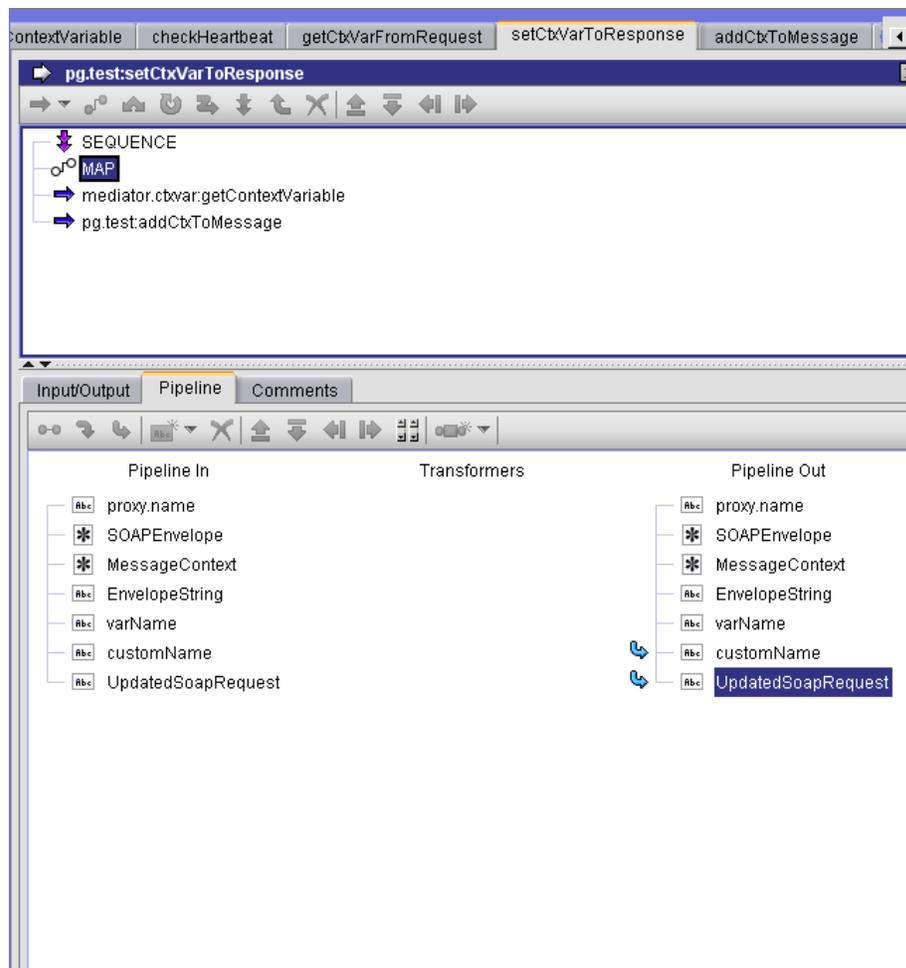
This flow service will retrieve the context variable for `customName` and create an element for its context variable value in the response message return to the consumer.

Step 1. Declaring `customName`



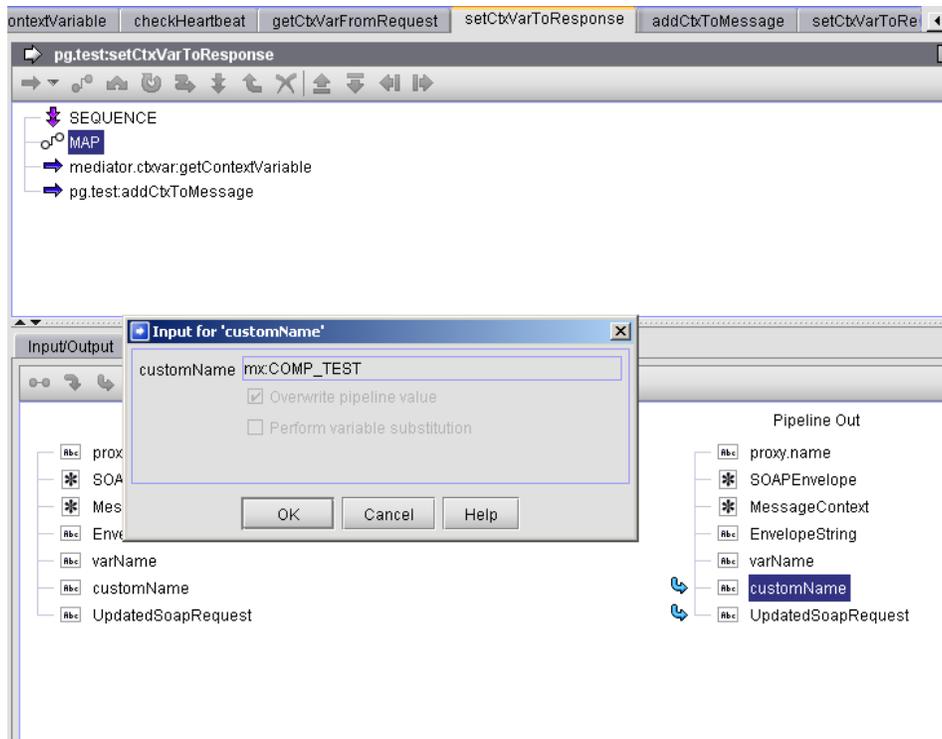
We define the `customName` variable value to be `mx:COMP_TEST` so we can use this variable to lookup the custom variable name that was seeded in the previous example.

Step 2. Setting `customName` to `mx:COMP_TEST`



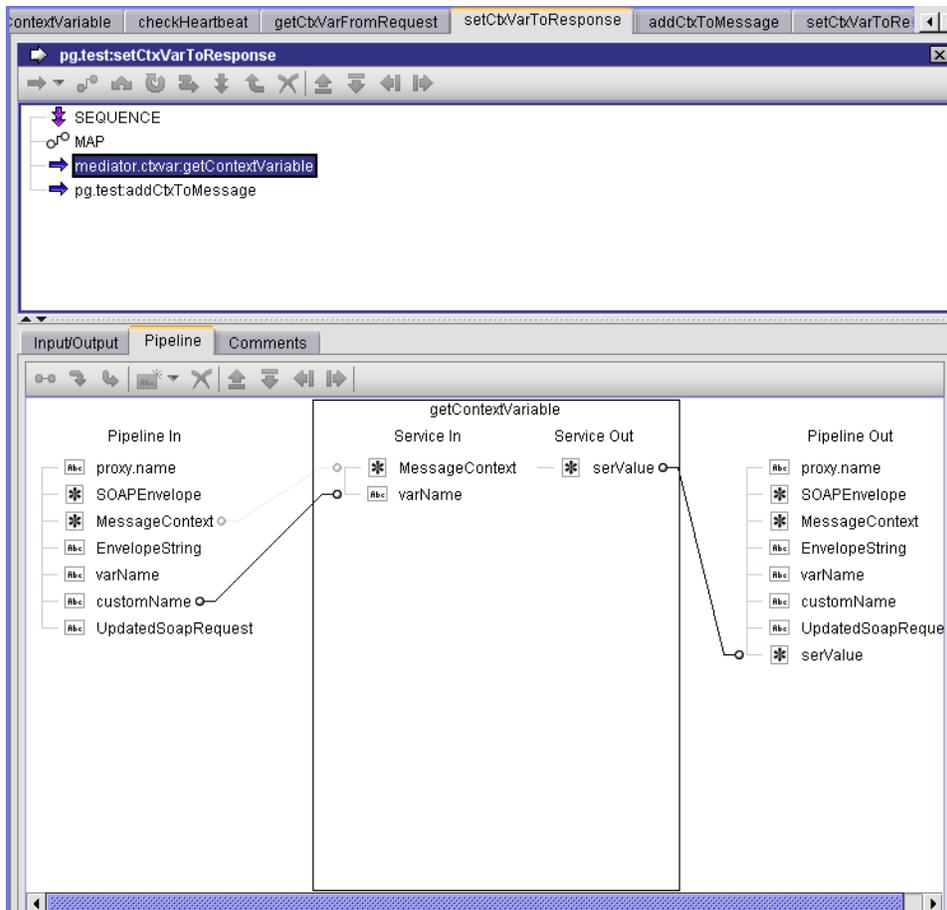
Clicking on the customName pipeline variable will display the name.

Step 3. Displaying the value of customName



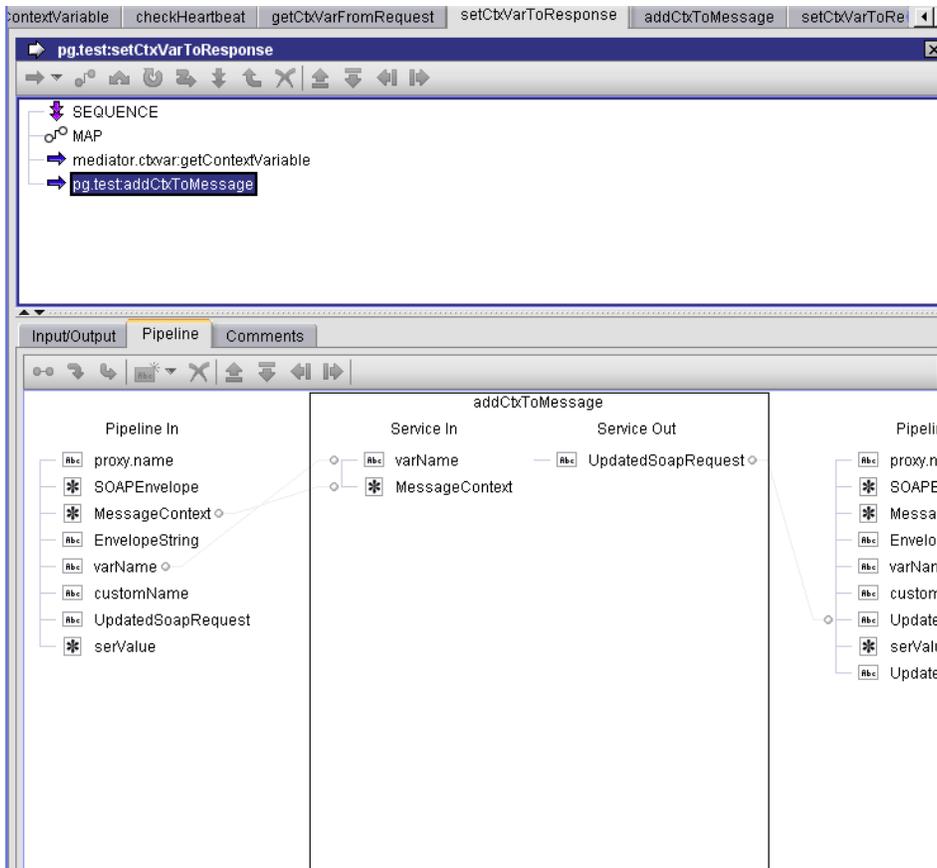
The call to `pub.mediator.ctxvar:getContextVariable` retrieves the value of the custom context variable from the context variable map.

Step 4. Calling `mediator.ctxvar:getContextVariable`



This is just a sample Java service that takes the context variable and creates a top-level element in the response message using the same name and value.

Step 5. Sample service using the context variable



Run-Time Policy Management

A run-time policy defines a sequence of actions that a policy-enforcement point (PEP), such as webMethods Mediator, will carry out when the PEP receives a request from a consumer application. The actions that a run-time policy can execute depend on the type of PEP you are using. If you are using webMethods Mediator, for example, you might create run-time policies to perform the following kinds of tasks:

- Verify that the requests submitted to a virtual service come from consumer applications that are authorized to use the virtual service.
- Validate request and response messages against the schema specified in the service's WSDL.
- Log the request and response messages to the CentraSite event log.
- Alert an administrator if the average response time for a service drops below a specified threshold.

Run-time policies enable service architects to separate the logic that relates to general functions, such as security, message validation, logging, and monitoring, from the business logic of the service. By using policies to carry out these general activities, an organization can easily modify and redeploy the policies as necessary without disrupting the native services that perform the business logic.

Like a design/change-time policy, a run-time policy consists of two major elements: an *action list* and a defined *scope*. The policy's action list specifies the sequence of actions that are to be executed by the PEP. The policy's scope determines to which virtual services the policy applies.

Note:

CentraSite does not support versioning for run-time policies.

Note:

The remaining sections assume that you are creating run-time policies that will be deployed with virtual services on webMethods Mediator. If you are using another PEP, the principles described in these sections will be similar, however, it will support its own unique set of policy actions. *The built-in policy actions that CentraSite provides out-of-the-box are to be used only with webMethods Mediator. You cannot use these actions with other PEPs.*

Run-Time Policy Actions

CentraSite is installed with a set of actions that you use to define run-time policies for virtual services deployed on webMethods Mediator. These actions fall into the following four categories:

Security actions, which you use to identify and authenticate the consuming application that submitted a request, to enforce the use of the SSL protocol (if required), to perform encryption and decryption of specified parts of the request and response messages, and to validate signatures of messages that are digitally signed.

Logging actions, which you use to log the request and/or response messages associated with a virtual service. Logging actions can send the logged messages to Mediator's local log, CentraSite's policy or audit log, to an SNMP server and/or email the messages to specified addresses.

Performance monitoring actions, which you use to monitor specified metrics (for example, service availability, average response time, fault count, request count) and log instances when these metrics violate specified thresholds. Violations can be reported in Mediator's local log, in CentraSite's policy or audit log, in an SNMP server and/or sent as an email message to specified addresses.

Data validation actions, which you use to validate the request and/or response message against the WSDL associated with the virtual service.

Run-Time Policy Scope

The scope of a run-time policy specifies for which virtual services the policy will be enforced. The scope of a run-time policy is determined by the following policy parameters:

- *The Target Type* parameter specifies the type of PEP on which the policy will be enforced. When you create run-time policies for webMethods Mediator, you set the Target Type to webMethods Integration Server.
- *The Organization* parameter specifies to which organization's virtual services the policy applies. Like design-time policies, a run-time policy can be system-wide or organization-specific. System-wide policies apply to virtual services in any organization. Organization-specific policies apply only to virtual services that belong to a specified organization.

Note:

Once you create a policy, its organizational scope is fixed and cannot be changed. That is, if you create a policy whose scope is specific to organization ABC, you cannot change its scope to make it system-wide or switch it to another organization. You must create a new policy and set its organizational scope as needed.

- *The Asset Type* parameter specifies to which types of assets the policy applies. When you create run-time policies for webMethods Mediator, you set the **Asset Type** parameter to Virtual Services. When you are creating a run-time policy for another type of PEP, you will set this parameter to Web Services.

Refining the Run-Time Policy's Scope with Additional Selection Criteria

Although it is possible to create a run-time policy that applies to all virtual services, this is not the common case. More frequently, you will create run-time policies that apply to a particular set of virtual services (or possibly even one specific virtual service). For example, you might define a run-time policy that monitors the response time for all virtual services that are considered to be “critical”.

To target a policy for a particular set of virtual services, you refine the policy's scope by specifying additional selection criteria based on the virtual service's Name, Description or Classification properties. For example, if you wanted to apply a particular run-time policy to critical services as described above, you would classify virtual services according to their criticality and create a policy that targets virtual services that are classified as critical.

In cases where you need to apply a run-time policy to one specific virtual service, you can use the selection criteria to identify the virtual service by name.

Note:

While you are creating a run-time policy, you can refer to the policy's **Service** profile to see exactly which set of virtual services are currently within the policy's scope.

To use run-time policies effectively, you need to think about what selection criteria your policies will use to identify the set of virtual services with which they are to be used. You must also ensure that the virtual services in your registry adhere to the Name, Description and/or Classification conventions needed to support your selection scheme. For example, if you intend to enforce different logging policies for different classes of virtual services, you must define the taxonomy by which the virtual services will be classified (for logging purposes) and ensure that virtual services are classified according to this taxonomy before they are deployed.

Run-Time Policy Deployment

When you deploy a virtual service to webMethods Mediator, CentraSite combines the actions from all of the run-time policies that apply to the virtual service and generates what is called the *effective policy* for the virtual service. For example, let's say your virtual service is within the scope of two run-time policies: one policy that performs a logging action and another policy that performs a security action. When you deploy the virtual service, CentraSite automatically combines the two policies into one effective policy. The effective policy, which contains both the logging action and the security action, is the policy that CentraSite actually deploys to the Mediator with the virtual service.

When CentraSite generates the effective policy, it validates the resulting action list to ensure that it contains no conflicting or incompatible actions. If the list contains conflicts or inconsistencies, CentraSite resolves them according to policy resolution rules.

For example, an action list can include only one Identify Consumer action. If the resulting action list contains multiple Identify Consumer actions, CentraSite resolves the conflict by including only one of the actions (selected according to a set of internal rules) in the effective policy and omitting the others.

The effective policy that CentraSite produces for a virtual service is contained in an object called a *virtual service definition* (VSD). The VSD is given to Mediator when you deploy the virtual service. After you deploy a virtual service, you can view its VSD (and, thus, examine the effective policy that CentraSite generated for it) from CentraSite Control or from the Mediator user interface.

Creating and Testing Policies

In a two-stage deployment of CentraSite, you will create and test run-time policies on the creation CentraSite and then promote them to the consumption CentraSite when they are considered ready for use.

Distributing the Development of Policies

Because run-time policies involve several different aspects of operational behavior, there are often several different types of architects or experts involved in their creation. For example,

- Security-related policies might be defined and managed by a security architect, who is responsible for ensuring that the policies adhere to corporate security standards.
- Performance-monitoring policies might be defined and managed by an administrator or analyst from the Operations organization.
- Application-related policies (that is, policies that involve logging or validating the data associated with the service itself) might be defined and managed by an SOA Application Architect.

Because CentraSite combines applicable run-time policies at deployment time, it is possible to distribute the policy-development responsibilities to the appropriate experts in your organization as described above. Each expert defines and manages the policies for his or her area of expertise. Then, at deployment time, CentraSite combines the applicable policies produced by each of these experts into one effective policy for the virtual service.

Lifecycle Model for Policies in a Multi-Stage Deployment of CentraSite

CentraSite provides a lifecycle model for policies that applies to both design/change-time policies and run-time policies. The lifecycle model has associated policies that CentraSite uses to validate, activate and deactivate policies. Because of the complex nature of this lifecycle model, we recommend that you use the lifecycle model as installed. Do not attempt to customize this lifecycle model. (You can associate additional policies with its state changes, however.)

If you are operating in a multi-stage deployment, this means that you will use the *same lifecycle model* on all stages in your environment. You will not define a multi-stage lifecycle model for policies as you would for an asset.

When you develop a run-time policy on the creation CentraSite, that policy will enter the New state. It will transition between the Productive and Suspended states as it undergoes development and testing. When the run-time policy is considered to be ready for production, you will promote it to the consumption CentraSite. Here it will also enter the lifecycle in the New state. When you are ready to activate the policy in the consumption CentraSite, you switch it to the Productive state.

Tip:

To indicate that a policy has been promoted to the production environment, consider adding a comment such as Promoted or Moved to Production to the policy's **Description** property on the creation CentraSite.

Actions that Run-Time Policies Can Execute

A run-time action is a single task that is included in a run-time policy and is evaluated by webMethods Mediator. Actions in run-time policies perform tasks such as identifying and validating consumers, traffic management and logging transaction activity. You specify actions when you define the policy.

CentraSite provides run-time action templates. A run-time action template is a definition of an action that can be used in a run-time policy. Most action templates specify a set of parameters associated with a particular policy action. For example, when you configure the Evaluate WSS Username Token action you use an identifier (for example, a WSS username token) to identify and validate the consumers who are trying to access the APIs. You can include multiple actions in a single policy.

Built-in Actions

CentraSite includes many built-in actions that you can use to compose run-time policies.

CentraSite Business UI

Built-in run-time actions are provided in the following categories:

Category	Description
Logging and Monitoring actions	Actions that monitor and collect information about the number of messages that were processed successfully or failed, the average execution time of message processing, and the number of alerts associated with an API.
Security actions	Actions that enforce identification and validation of the consumers who are trying the access the API.

Category	Description
Traffic management actions	Actions that limit the number of service invocations allowed during a specified time interval, and send alerts to a specified destination when the performance conditions are violated.
Validation	Actions that validate all XML request and response messages against an XML schema referenced in the WSDL.

CentraSite Control

Built-in run-time actions are provided in the following categories:

Category	Description
WS-SecurityPolicy 1.2 actions	<p>Mediator provides two kinds of actions that support WS-SecurityPolicy 1.2: authentication actions and XML security actions.</p> <ul style="list-style-type: none"> ■ You use the authentication actions to verify that the service consumer has the proper credentials to access a virtual service. You can authenticate consumers by their WSS X.509 certificates, WSS Username tokens, or WSS SAML tokens. ■ You use the XML security actions to provide confidentiality (through encryption) and integrity (through signatures) for request and response messages.
Monitoring actions	<p>Mediator provides the following run-time monitoring actions:</p> <ul style="list-style-type: none"> ■ The Monitor Service Performance action that monitors a user-specified set of run-time performance conditions for a virtual service and sends alerts to a specified destination when these conditions are violated. ■ The Monitor Service Level Agreement action that provides the same functionality as Monitor Service Performance, but this action is different because it enables you to monitor a virtual service's run-time performance for particular consumers. You configure this action to define a <i>Service Level Agreement (SLA)</i> that is set of conditions that defines the level of performance that a specified consumer should expect from a service. ■ The Throttling Traffic Optimization action (not available in Mediator versions below 9.0) that limits the number of service invocations allowed during a specified time interval and sends alerts to a specified destination when

Category	Description
Additional actions	<p data-bbox="573 260 1284 394">the performance conditions are violated. You can use this action to avoid overloading the back-end services and their infrastructure, limit specific consumers in terms of resource usage, and so on.</p> <p data-bbox="526 422 1284 489">Mediator provides the following actions, which you can use in conjunction with the actions above:</p> <ul data-bbox="526 516 1284 1465" style="list-style-type: none"><li data-bbox="526 516 1284 720">■ Identify Consumer, used in conjunction with an authentication action (Require WSS Username Token, Require WSS X.509 Token, or Require HTTP Basic Authentication). Alternatively, you can use this action alone to identify consumers only by host name or IP address.<li data-bbox="526 747 1284 882">■ Require HTTP Basic Authentication that uses HTTP basic authentication to verify the consumer's authentication credentials contained in the request's Authorization header against the Integration Server.<li data-bbox="526 909 1284 1113">■ Authorize User that authorizes consumers against a list of users and a list of groups registered in the Integration Server on which Mediator is running. You use this action in conjunction with an authentication action (Require WSS Username Token, Require WSS SAML Token, or Require HTTP Basic Authentication).<li data-bbox="526 1140 1284 1245">■ Authorize Against Registered Consumers that authorizes consumer applications against all Application assets that are registered in CentraSite as consumers for the service.<li data-bbox="526 1272 1284 1339">■ Log Invocation that logs request or response payloads to a destination you specify.<li data-bbox="526 1367 1284 1465">■ Validate Schema that validates all XML request and response messages against an XML schema referenced in the WSDL.

Custom Actions

If you need to execute a task that is not provided by a built-in action, you can create a custom action to perform the work. CentraSite offers the functionality to implement custom computed actions with your own algorithms using the GWT framework.

Supported Run-Time Actions and Asset Type Combinations

Not all virtual asset types support the full set of actions. Some actions execute only with a certain type of virtual assets. For example, a Require Encryption action executes only on Virtual Service

asset type. If you create a run-time policy whose scope applies to Virtual REST Service asset type, that policy action definition does not include the Require Encryption action.

The following table identifies the actions that each virtual asset type supports:

Action	Description	Applicable for..		
		Web Service	REST Service	OData Service
Require JMS	Specify the JMS protocol for the API to accept and process the requests.	✓		
Require HTTP / HTTPS	Specify the HTTP and/or HTTPS protocol and the SOAP format (for a SOAP-based API) for the API to accept and process the requests.	✓	✓	✓
Request Transformation	Invoke an XSL transformation in the incoming request before it is submitted to the an API.	✓	✓	
Invoke webMethods Integration Server	Invoke a webMethods Integration Server service to pre-process the request before it is submitted to the an API.	✓	✓	✓
Enable REST Support	Enables REST support for an existing SOAP based API by exposing the API both as a REST based API as well as a SOAP API.	✓		
Set Media Type	Specifies the content type for a REST request received from a client if the content type header is not specified.	✓	✓	
HTTP Basic Authentication	Identify and validate the consumer's authentication credentials contained in the request's Authorization header using HTTP basic	✓	✓	✓

Action	Description	Applicable for...		
		Web Service	REST Service	OData Service
	authentication mechanism.			
NTLM Authentication	Identify and validate the consumer's authentication credentials contained in the request's Authorization header using NTLM authentication mechanism.	✓	✓	✓
OAuth2 Authentication	Identify and validate the consumer's authentication credentials contained in the request's Authorization header using OAuth 2.0 authentication mechanism.	✓	✓	
JMS Routing Rule	Specify a JMS queue to which the Mediator is to submit the request, and the destination to which the an API is to return the response.	✓		
Set Message Properties	Specify JMS message properties to authenticate client requests before submitting to the an APIs.	✓		
Set JMS Headers	Specify JMS headers to authenticate client requests before submitting to the an APIs.	✓		
Log Invocation	Log request/response payloads to a destination you specify.	✓	✓	✓
Monitor Service Performance	Monitor the run-time performance for a specific consumer, and defines the level of performance that the specified consumer	✓	✓	✓

Action	Description	Applicable for...		
		Web Service	REST Service	OData Service
	should expect from the API.			
Monitor Service Level Agreement	Monitor a user-specified set of run-time performance conditions for an API, and sends alerts to a specified destination when these performance conditions are violated.	✓	✓	✓
Straight Through Routing	Route requests directly to a native endpoint that you specify.	✓	✓	✓
Content Based Routing	Route requests to different endpoints based on specific values that appear in the request message.	✓	✓	
Dynamic Routing	Enables Mediator to support dynamic routing of virtual aliases based on policy configuration.	✓	✓	✓
Load Balancing and Failover Routing	Routes the requests across multiple endpoints.	✓	✓	✓
Context Based Routing	Route requests to different endpoints based on specific values that appear in the request message.	✓	✓	✓
Set Custom Headers	Specify the HTTP headers to process the requests.	✓	✓	✓
Require SSL	Mandate that requests be sent via SSL client certificates, and can be used by both SOAP and REST APIs.	✓		
Require Signing	Mandate that a request's XML element (which is represented by an XPath expression) be signed.	✓		

Action	Description	Applicable for...		
		Web Service	REST Service	OData Service
Require Encryption	Mandate that a request's XML element (which is represented by an XPath expression) be encrypted.	✓		
Require Timestamps	Mandate that timestamps be included in the request header.	✓		
Require WSS SAML Token	Uses a WSS Security Assertion Markup Language (SAML) assertion token to validate API consumers.	✓		
Evaluate HTTP Basic Authentication	<ul style="list-style-type: none"> ■ Identify the consumer against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator). ■ Validate the client's authentication credentials contained in the request's Authorization header against the list of users in the Integration Server on which Mediator is running. 	✓	✓	✓
Evaluate Hostname	<ul style="list-style-type: none"> ■ Identify the consumer against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator). 	✓	✓	✓

Action	Description	Applicable for...		
		Web Service	REST Service	OData Service
	<ul style="list-style-type: none"> Validate the client's IP address against the list of users in the Integration Server on which Mediator is running. 			
Evaluate IP Address	<ul style="list-style-type: none"> Identify the consumer against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator). Validate the client's IP address against the list of users in the Integration Server on which Mediator is running. 	✓	✓	✓
Evaluate WSS Username Token	<ul style="list-style-type: none"> Identify the consumer against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator). Validate the client's WSS username token against the list of users in the Integration Server on which Mediator is running. 	✓		
Evaluate WSS X.509 Certificate	<ul style="list-style-type: none"> Identify the consumer against either the Registered Consumers list (the list of 	✓		

Action	Description	Applicable for...		
		Web Service	REST Service	OData Service
	<p>registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator).</p> <ul style="list-style-type: none"> ■ Validate the client's WSS X.509 token against the list of users in the Integration Server on which Mediator is running. 			
Evaluate XPath Expression	<ul style="list-style-type: none"> ■ Identify the consumer against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator). ■ Validate the client's XPath expression against the list of users in the Integration Server on which Mediator is running. 	✓	✓	
Evaluate OAuth2 Token	<ul style="list-style-type: none"> ■ Identify the consumer against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator). ■ Validate the client's IP address against the list of users in the Integration Server on 	✓	✓	

Action	Description	Applicable for...		
		Web Service	REST Service	OData Service
	which Mediator is running.			
Evaluate Client Certificate for SSL Connectivity	<ul style="list-style-type: none"> Identify the consumer against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available users in Mediator). Validate the client's certificate against the list of users in the Integration Server on which Mediator is running. 	✓	✓	✓
Throttling Traffic Optimization	<ul style="list-style-type: none"> Limit the number of API invocations during a specified time interval, and sends alerts to a specified destination when the performance conditions are violated. Avoid overloading the back-end services and their infrastructure, to limit specific consumers in terms of resource usage, and so on. 	✓	✓	✓
Service Result Cache	Enables caching of the results of SOAP and REST API invocations.	✓	✓	✓
Validate Schema	Validate all XML request and/or response messages against an XML schema referenced in the WSDL.	✓		

Action	Description	Applicable for...		
		Web Service	REST Service	OData Service
Response Transformation	Invoke an XSL transformation in the SOAP response payloads from XML format to the format required by the consumer.	✓	✓	
Invoke webMethods Integration Server	Invoke a webMethods Integration Server service to process the response from the an API before it is returned to the consumer.	✓	✓	✓
Set Media Type	Specifies the content type for a REST response.	✓	✓	
Conditional Error Processing	Return a custom error message (and/or the native provider's service fault content) to the consumer when the native provider returns a service fault.	✓	✓	✓

Managing Run-Time Policies through CentraSite Business UI

Creation and configuration of run-time policies from CentraSite are disabled by default. However, you can enable it by modifying the `centrasite.xml` file. For the procedure, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

If the default configuration is overridden, then you create run-time policies and apply them to proxy APIs in order to govern the APIs' run-time execution.

A run-time policy is a sequence of actions that are carried out by a policy-enforcement point (PEP) gateway when a consumer requests a particular API through the gateway. The actions in a run-time policy perform activities such as identifying and validating consumers, validating signatures, and capturing performance metrics.

This section describes how to create run-time policies using CentraSite Business UI and store them in the CentraSite registry or repository.

When you create a run-time policy in CentraSite Business UI, you:

- Define the virtual types on which you want to enforce the policy. CentraSite provides predefined virtual types.

- Add run-time actions to the policy and configure their parameters. CentraSite provides a set of built-in run-time actions.
- Specify the gateway (for example, webMethods Mediator) to which you publish the policy and its virtual types.
- Activate the policy.

Creating a Run-Time Policy

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

Pre-requisites:

To create a new run-time policy in CentraSite, you must have one of the following permissions:

- To create policies for a specific organization, you must have the Manage Run-Time Policies permission for that organization. By default, users in the CentraSite Administrator, Organization Administrator, or Policy Administrator role have this permission.
- To create system-wide policies (that is, policies that apply to all organization within an instance of CentraSite), you must have the Manage System-Wide Run-Time Policies permission. By default, users in the CentraSite Administrator role and Operations Administrator role have this permission.

> To create a run-time policy

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. On the actions bar of the Search Results page, click **Add Policy**.
3. On the **Create Run-Time Policy (Step 1 of 3)** page, provide the required information for each of the displayed data fields.
 - a. Specify the basic information for the runtime policy.

In this field...

Do the following...

Name

Type a name for the new policy. A policy name can contain any character (including spaces). A policy name does not need to be unique within the registry. However, to reduce ambiguity, you should avoid giving multiple policies the same name. As a best practice, Software AG recommends that you adopt appropriate naming conventions to ensure that policies are distinctly named within your organization.

In this field... **Do the following...**

Description (Optional). Type a description for the new policy. This description appears when a user displays a list of policies in the user interface.

Version (Optional). Specify a version identifier for the new policy.

Note:
The version identifier does not need to be numeric.

Examples:

0.0a

1.0.0 (beta)

Pre-release 001

V1-2007.04.30

The version identifier you type here is the policy's public, user-assigned version identifier.

Apply policy to all organizations Enable the **Apply policy to all organizations** checkbox if you want to apply the policy to the specified services in all organizations.

Note:
The **Apply policy to all organizations** checkbox appears if you belong to the CentraSite Administrator role.

Apply Policy to Organization Alternatively, select an organization to which the policy applies.

Note:
The **Apply Policy to Organization** list contains the names of all organizations if you belong to the API Runtime Provider role.

- b. In the **Filters** panel, specify the scope of the runtime policy.
 - a. In the **Applicable Types** section, select the virtual types of services to which this policy applies. Select one of the following:
 - Virtual Service
 - Virtual REST Service
 - Virtual XML Service
 - Virtual OData Service

- b. In the **Filter Criteria** section, specify additional selection criteria to narrow the set of services to which this runtime policy applies.
 - c. Click **Next**.
 4. On the **Create Run-Time Policy (Step 2 of 3)** page, perform the following:
 - a. Select the actions that you want CentraSite to execute when it applies this policy.

If necessary, you can click **Previous** to return to **Create Run-Time Policy (Step 1 of 3)** and change your scope.
 - b. Click **Save**.
 - c. Configure the parameters for each action on the **Message Flow** area.
 - d. Click **Next**.
 5. On the **Create Run-Time Policy (Step 3 of 3)** page, perform the following:
 - a. In the **Available Gateways** list, mark the checkbox next to the name of each gateway (for example, webMethods Mediator) you want to publish the policy. (You can select multiple gateways.)
 - b. Review the virtual services that are in the scope of this policy and already published to the selected gateways.

Click **Previous**, if required, to return to **Create Run-Time Policy (Step 2 of 3)** and modify the action parameters.
 6. When you are ready to put the policy into effect, activate and publish the policy.

Activating a Run-Time Policy

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

Pre-requisites:

To activate a run-time policy in CentraSite, you must have one of the following permissions:

- To activate a organization-specific policy, you must have the Manage Run-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to activate that policy unless you have permission to manage run-time policies for organization ABC.

- To activate a system-wide policy, you must have the Manage System-Wide Run-Time Policies permission.

A run-time policy is not eligible for deployment unless it is active. In other words, when CentraSite deploys a virtual service, it uses only active run-time policies to produce the effective policy for the virtual service. Whether a run-time policy is active or inactive is determined by its lifecycle state.

Run-time policies are governed by the same lifecycle model as design/change-time policies. The predefined lifecycle model for policies that is installed with CentraSite is made up of four states: New, Productive, Suspended and Retired. Under this lifecycle model, all policies enter the New state when they are initially created.

While a run-time policy is in the New state, it is inactive. To activate a policy, you change the policy's lifecycle state to the Productive state. This state change executes CentraSite's *Automatic Policy Activation* policy.

Note:

The *Automatic Policy Activation* policy is a hidden system policy. You cannot edit or delete this policy.

If a virtual service is already deployed to webMethods Mediator and is in the scope of an activated run-time policy, you must redeploy the virtual service to Mediator so that the actions are activated.

Note:

We recommend that you use the lifecycle model that CentraSite provides for policies “as-is”, even for a multi-stage deployment.

CentraSite does not begin enforcing a run-time policy until you *activate* it.

When you activate a run-time policy, keep the following points in mind:

- You are not allowed to activate the policy unless all the required parameters have been set.
- Some organizations require an approval to activate the policy. If your organization has an approval action associated with the activation of a policy, CentraSite does not activate the policy until the required approvals are obtained.
- If an earlier version of the policy is already active, CentraSite deactivates the old version before it activates the new one. When a policy becomes active, CentraSite begins enforcing it immediately.
- When a policy becomes active, CentraSite begins enforcing it immediately. You can suspend enforcement of a policy by switching it to the Suspended state.
- To activate a policy, you must have permission to change the policy to the Productive state. To successfully change a policy to the Productive state, you must also have the Modify permission on all virtual type services to which the policy is applied.
- Upon activation of a policy, CentraSite applies the active policy to all of the virtual APIs that are in the scope of this policy.

- You should be aware that a run-time policy is automatically published whenever a virtual API associated to it is published to the Mediator gateway.

To determine whether a policy is active or inactive, examine the policy's decoration indicator on the Search Results for Governance Rules page. The decoration indicates the policy's activation state as follows:

Decoration	Description
	Policy is active.
	Policy is inactive.

➤ To activate a run-time policy

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. To filter the list to see just a list of the available run-time policies, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, do the following:
 - a. Locate **Applicable Scopes**.
 - b. Choose **Runtime Policy** from the drop-down list.
 - c. Click the plus button next to the drop-down box or press Enter to add the scope **Runtime Policy** to the search recipe.

CentraSite displays the list of runtime policies that are available to you in the Search Results page.
3. Click the policy you want to activate.

This opens the Run-Time Policy Details page.
4. Examine the information on **Run-Time Policy Details (Step 2 of 3)** and verify that all of the actions on the **Message Flow** area are set properly.
5. In **Run-Time Policy Details (Step 3 of 3)**, do one of the following:
 - Click **Activate** to activate the policy. (If you do not see the **Activate** button, it is probably because you do not have permission to change the lifecycle state of a policy.)
 - Click **Activate and Publish** to activate and publish the policy to webMethods Mediator in a single step. (If you do not see the **Activate and Publish** button, it is probably because you do not have permission to change the lifecycle state of a policy or publish the policy.)

6. Examine the policy's decoration indicator on the **Search Results for Governance Rules** page to verify that the policy's state has been changed.

If this state change requires approval, the policy's decoration indicator indicates that the policy is in the pending mode. CentraSite automatically switches the policy to the requested state (and activate the policy) after all the necessary approvals have been obtained.

Alternatively, in the **Search Results for Governance Rules** page, you can change the policy's lifecycle state to the Productive state, by toggling the policy's decoration indicator on and off.

Note:

While the policy is in pending mode, it cannot be edited.

Deactivating a Run-Time Policy

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

Pre-requisites:

To deactivate a run-time policy in CentraSite, you must have one of the following permissions:

- To deactivate a organization-specific policy, you must have the Manage Run-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to deactivate that policy unless you have permission to manage run-time policies for organization ABC.
- To deactivate a system-wide policy, you must have the Manage System-Wide Run-Time Policies permission.

Deactivating a run-time policy causes CentraSite to suppress enforcement of the policy. You usually deactivate a policy for the following reasons:

- To suspend enforcement of a particular policy (temporarily or permanently).
- To edit a policy (for example, to modify the scope of a policy or change its action list).

To deactivate a policy, you change the policy to the Suspended state. Switching the policy to this state triggers the *Automatic Policy Deactivation* policy, which deactivates the policy. Switching the policy to the Retired state also deactivates the policy, but you do not want to switch a policy to this state unless you intend to deactivate it permanently. After you place a policy in the Retired state, you cannot reactivate it.

When you deactivate a policy, CentraSite does not deactivate a policy if it is in the process of being executed. If you attempt to deactivate a policy while it is executing, your state change request fails. If this occurs, wait for a period time and then try to deactivate the policy again.

> To deactivate a run-time policy

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. To filter the list to see just a list of the available policies, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, do the following:
 - a. Locate **Applicable Scopes**.
 - b. Choose **Runtime Policy** from the drop-down list.
 - c. Click the plus button next to the drop-down box or press Enter to add the scope **Runtime Policy** to the search recipe.

CentraSite displays the list of runtime policies that are available to you in the Search Results page.

3. Click the policy you want to deactivate.

This opens the Run-Time Policy Details page.

4. Toggle the policy's decoration indicator to **Suspended** state to deactivate it temporarily or the Retired state to deactivate it permanently. (If you do not see the decoration indicator, it is probably because you do not have permission to change the lifecycle state of a policy.)
5. Examine the policy's decoration indicator on the Search Results for Governance Rules page to verify that the policy's state has been changed.

If this state change requires approval, the policy's decoration indicator indicates that the policy is in the pending mode. CentraSite automatically switches the policy to the requested state (and deactivate the policy) after all the necessary approvals have been obtained.

Viewing Run-Time Policy List

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

The Governance Rules page displays the run-time policies for all organizations, not just your own. It also includes system-wide policies.

By default, all users have View permissions on the run-time policies in the registry.

➤ To view the run-time policy list

1. In the CentraSite Business UI activity bar, click **Governance Rules**.

2. To filter the list to see just a list of the available policies, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, do the following:
 - a. Locate **Applicable Scopes**.
 - b. Choose **Runtime Policy** from the drop-down list.
 - c. Click the plus button next to the drop-down box or press Enter to add the scope **Runtime Policy** to the search recipe.

CentraSite displays the list of runtime policies that are available to you in the Search Results for Governance Rules page.
 - c. To further filter the list of runtime policies:

To see...**Do this...**

A subset of the available policies

Type a partial string in the **Keyword** text field. Click the plus button next to the drop-down list or press Enter to add the keyword to the search recipe.

The list of policies whose scope applies for a particular virtual type

Click **Choose a type** button. In the **Runtime Virtual Types** dialog box, select **Everything** or one of the following virtual types:

- **Everything**: Displays the list of policies whose scope applies to any of the virtual types: Virtual Service, Virtual REST Service, and Virtual XML Service.
- **Virtual Service**: Displays the list of policies whose scope applies to the Virtual Service type.
- **Virtual REST Service**: Displays the list of policies whose scope applies to the Virtual REST Service type.
- **Virtual XML Service**: Displays the list of policies whose scope applies to the Virtual XML Service type.
- **Virtual XML Service**: Displays the list of policies whose scope applies to the Virtual XML Service type.
- **Virtual OData Service**: Displays the list of policies whose scope applies to the Virtual OData Service type.

To see...	Do this...
A list of the active run-time policies	Enable the Show only active policies option.
The list of policies whose scope applies for a particular organization	<ol style="list-style-type: none"> In the Applicable Organizations section, select an organization from the drop-down list. Click the plus button next to the drop-down list or press Enter to add the selected organization to the search recipe.
The list of policies whose scope applies for all organizations	<ol style="list-style-type: none"> In the Applicable Organizations section, select All. Click the plus button next to the drop-down list or press Enter to add the selected organization to the search recipe.

3. The Search Results page provides the following information about each policy:

Column	Description						
Name	The name assigned to the policy.						
Description	Additional comments or descriptive information about the policy.						
Created Date	The date on which the policy was added to the registry. CentraSite automatically sets this attribute when a user adds the policy to the registry. Once it is set, it cannot be modified.						
Last Updated Date	The date on which the policy was last updated. CentraSite automatically updates this attribute when a user modifies any of the policy's attributes.						
State	The policy's current lifecycle state.						
Owner	The user to which the policy belongs.						
Organization	The organization to which the policy applies.						
	<table border="1"> <thead> <tr> <th><u>This value...</u></th> <th><u>Indicates that...</u></th> </tr> </thead> <tbody> <tr> <td>All</td> <td>The policy is system-wide and applies to all organizations.</td> </tr> <tr> <td><i>OrgName</i></td> <td>The policy applies to the specified organization.</td> </tr> </tbody> </table>	<u>This value...</u>	<u>Indicates that...</u>	All	The policy is system-wide and applies to all organizations.	<i>OrgName</i>	The policy applies to the specified organization.
<u>This value...</u>	<u>Indicates that...</u>						
All	The policy is system-wide and applies to all organizations.						
<i>OrgName</i>	The policy applies to the specified organization.						
Active	The policy's current enforcement state.						
	<table border="1"> <thead> <tr> <th><u>Icon</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td></td> <td>The policy is active. CentraSite enforces this policy when events within the scope of the policy occur.</td> </tr> </tbody> </table>	<u>Icon</u>	<u>Description</u>		The policy is active. CentraSite enforces this policy when events within the scope of the policy occur.		
<u>Icon</u>	<u>Description</u>						
	The policy is active. CentraSite enforces this policy when events within the scope of the policy occur.						

Column	Description
	 The policy is inactive. Inactive policies exist in the registry, but they are not enforced.
Priority	The priority value assigned to the policy.
Version	The user-assigned version identifier for the policy.

Modifying Run-Time Policy Details

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

Pre-requisites:

To examine and modify the properties of a run-time policy in CentraSite, you must have one of the following permissions:

- To examine and modify the properties of a organization-specific policy, you must have the Manage Run-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to examine and modify the properties of that policy unless you have permission to manage run-time policies for organization ABC.
- To examine and modify the properties of a system-wide policy, you must have the Manage System-Wide Run-Time Policies permission.

Important:

If you belong to a role that includes the Manage System-Wide Run-Time Policies permission, you have the ability to modify CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be modified except under the direction of a technical representative from Software AG.

You cannot modify a policy while it is in the Productive state. To make changes to a policy, you can do any of the following:

- Create a new version of the policy, make the necessary changes to the new version and switch the new version to the Productive state when you are ready to put it into effect. Switching the new version to the Productive state will immediately put the previous version in the Retired state. (The Retired state is an end state. After you place a policy in this state, you can no longer reactivate it.)
- Create a completely new policy that includes the required changes. When you are ready to put the new policy into effect, switch the old policy to the Suspended state and switch the new policy to the Productive state. When you are certain that you will no longer need to revert to the original policy, switch it to the Retired state.

- Switch the existing policy to the Suspended state, make the necessary changes to the policy and then switch it back to the Productive state. While the policy is in the Suspended state, it will not be enforced. (Because suspending the policy results in an enforcement gap, one usually does not use this approach in a production environment.)

➤ **To view or modify the properties of a run-time policy**

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. To filter the list to see just a list of the available policies, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, do the following:
 - a. Locate **Applicable Scopes**.
 - b. Choose **Runtime Policy** from the drop-down list.
 - c. Click the plus button next to the drop-down box or press Enter to add the scope **Runtime Policy** to the search recipe.

CentraSite displays the list of runtime policies that are available to you in the Search Results for Governance Rules page.

3. Click the policy you want to examine and modify the attributes.

This opens the Run-Time Policy Details page.

4. If the policy is active, deactivate it.

You cannot modify the details of an active policy.

5. On the Run-Time Policy Details page, examine or modify the properties as required.

Field	Description
Name	The name of the run-time policy. A policy name can contain any character (including spaces). A policy name does not need to be unique within the registry. However, to reduce ambiguity, you should avoid giving multiple policies the same name. As a best practice, we recommend that organizations adopt appropriate naming conventions to ensure the assignment of distinct policy names.
Description	(Optional). Additional comments or descriptive information about the policy.

Field	Description
Version	<p>The user-assigned version ID assigned to this policy. You may use any versioning scheme you select for identifying different versions of a policy. The identifier does not need to be numeric.</p> <p>Examples:</p> <p>0.0a</p> <p>1.0.0 (beta)</p> <p>Pre-release 001</p> <p>V1-2007.04.30</p>
Apply policy to all organizations (option)	This property determines if the policy is system-wide (global).
Apply Policy to Organization	This property determines if the policy belongs to a specific organization.
Filters	The settings on this section determine the virtual types to which the policy is applied.
Policy Actions Message Flow	The settings on this area specify the actions that the gateway executes when the policy is enforced.
Gateways	One or more gateways to which the policy is published.
Published Virtual APIs	Displays the list of virtual APIs/virtual REST APIs/virtual XML APIs to which the policy applies.

6. Click **Save**.

7. When you are ready to put the policy into effect, activate and publish the policy.

Scope of a Run-Time Policy

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

Scope refers to the set of properties that determine when a policy is enforced. For a run-time policy, scope is determined by the policy's **Organization**, **Applicable Virtual Types**, **Published APIs** and **Gateways** properties, which are described below.

Property	Description
----------	-------------

Organization	Determines whether the policy belongs to a specific organization or is system-wide.
---------------------	---

Important:

You cannot change the value of the **Apply Policy to Organization** property if the policy is system-wide.

Applicable Virtual Types	The list of virtual types to which this policy applies.
---------------------------------	---

- Virtual Service
- Virtual REST Service
- Virtual ODATA Service
- Virtual XML Service

You can optionally restrict a policy to specific instances of the selected virtual types by specifying additional filter criteria.

Applicable Gateways	The list of gateways to which the policy applies.
----------------------------	---

- API Gateway
- Mediator
- Insight Server

System-wide and Organization-specific Policy Enforcement

The **Organization** property specifies the organization to which the policy applies. When the **Organization** property is set to `ALL`, it indicates that the policy is *system-wide*. When the **Organization** property specifies a particular organization, it indicates that the policy is *organization-specific*.

Note:

By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#). This section is not applicable, if the CentraSite run-time aspects are not enabled.

Organization-specific Policies

An organization-specific policy is enforced on objects that belong to the same organization as the organization to which the policy applies. For example, if you have a policy that executes when user objects are updated and its **Organization** property specifies organization ABC, CentraSite only execute that policy when user objects *in organization ABC* are updated.

Points to keep in mind when working with organization-specific policies:

- You can create organization-specific policies for any organization on which you have Manage Design/Change-Time Policies permission. For example, if you have Manage Design/Change-Time Policies permission for organization ABC and XYZ, you can create organization-specific policies for either organization.
- At enforcement time, CentraSite selects policies based on the organization to which the object belongs, *not* the organization to which the requestor belongs. For example, if a user from organization XYZ edits an asset in organization ABC, CentraSite applies organization ABC's policies (the organization to which the asset belongs), not organization XYZ's policies (the organization to which the requestor belongs).

System-wide Policies

A system-wide policy is enforced for all organizations. For example, if you create a system-wide policy that executes when an asset is created, CentraSite enforces the policy whenever *any* user in *any* organization adds an asset to the catalog.

To create a system-wide policy, you must belong to a role that has Manage System-Wide Design/Change-Time Policies permission. In a standard CentraSite configuration, only users in the CentraSite Administrator role and the Policy Administrator role have this permission.

System-wide policies are useful for managing many types of objects. For example, they are often used to assign users to certain server-wide groups or to enforce server-wide naming conventions on objects. However, organization-specific policies are often better choices for asset-related policies, because they enable an organization to tailor its policies to its own development processes and methodologies.

Modifying Scope of a Run-Time Policy

Note:

By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#). This section is not applicable, if the CentraSite run-time aspects are not enabled.

Pre-requisites:

To modify the scope of a run-time policy in CentraSite, you must have one of the following permissions:

- To modify the scope of an organization-specific policy, you must have the Manage Run-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to modify the scope of that policy unless you have permission to manage run-time policies for organization ABC.
- To modify the scope of a system-wide policy, you must have the Manage System-Wide Run-Time Policies permission.

You use the **Apply policy to all organizations** property on the Run-Time Policy Details page to specify a policy's scope.

Note:

After a policy has been created, its **Apply policy to all organizations** property cannot be changed.

> To modify the scope of a run-time policy

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. To filter the list to see just a list of the available policies, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, do the following:
 - a. Locate **Applicable Scopes**.
 - b. Choose **Runtime Policy** from the drop-down list.
 - c. Click the plus button next to the drop-down box or press Enter to add the scope **Runtime Policy** to the search recipe.

CentraSite displays the list of runtime policies that are available to you in the Search Results page.
3. Select the policy whose scope you want to modify.

This opens the Run-Time Policy Details page.
4. If the policy is active, deactivate it.

You cannot modify the scope of an active policy.
5. On the **Run-Time Policy Details** page, do the following:
 - Enable the **Apply policy to all organizations** property if the policy applies to all organizations. Else, in the **Apply Policy to Organization** property, select the organization to which the policy applies.
 - In the **Applicable Types** lists, select the virtual types to which the policy applies.
 - Optional. In the **Filter Criteria** section, specify additional selection criteria to narrow the set of objects to which this policy is applied.
 - In the **Available Gateways** lists, select the gateways to which the policy is applied.
6. Click **Save** to save the updated policy.
7. When you are ready to put the policy into effect, activate and publish the policy.

Refining the Policy Scope

Note:

By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#). This section is not applicable, if the CentraSite run-time aspects are not enabled.

To further restrict the set of virtual APIs to which the policy is applied, you can specify additional selection criteria in the **Filter criteria** section of the **Run-Time Policy Details** page. Using this section, you can filter objects by Name, Description, and Classification attributes. If you specify no filter criteria, the policy applies to all virtual APIs.

■ Filtering by Name and Description

You can filter policies based on their Name and Description attributes using any of the following comparison operators:

Comparison Operator	Description
Equals	Selects APIs whose Name or Description value matches a given string of characters. For example, you would use this operator if you wanted to apply a policy only to APIs with the Name or Description value <code>Mobile App Store</code> .
Contains	Selects APIs whose Name or Description value includes a given string of characters anywhere within the property's value. For example, you would use this operator if you wanted to apply a policy to APIs that had the word <code>Mobile</code> anywhere in their Name or Description property.
StartsWith	Selects APIs whose Name or Description value begins with a given string. For example, you would use this operator if you wanted to apply a policy only to APIs whose Name or Description begins with the characters <code>Mobile</code> .

When specifying match strings for the comparison operators described above, keep the following points in mind:

- Match strings *are not* case-sensitive. If you define a filter for names that start with `ABC` it selects names starting `abc` and `Abc`.
- Wildcard characters are not supported. That is, you cannot use characters such as `*` or `%` to represent *any sequence of characters*. These characters, if present in the match string, are simply treated as literal characters that are to be matched.

■ Filtering by Classification Attribute

You can also filter APIs based on the way in which they are classified. When you filter APIs in this way, CentraSite applies the policy to APIs that have at least one classification attribute

whose value matches a specified taxonomy category. For example, you could use a classification filter to apply a policy to those APIs that are classified with Production sandbox.

When you filter APIs by classification, CentraSite inspects all of an APIs classification attributes. If any of those attributes contain the exact category specified by the selection criteria, the API is listed in **Run-Time Policy Details (Step 3 of 3)** of the policy.

Note:

To satisfy the selection criteria, the attribute value in the API must match the category specified in the selection criteria *exactly*. Sub-categories of the specified category *are not* considered to be matches. For example, say you have a taxonomy category called Project ABC, and that category has the subcategories Project ABC Design, Project ABC Development, and Project ABC Deployment. If you filter for category Project ABC, CentraSite applies the policy to objects that are classified by the specific category Project ABC, but not objects that are classified by that category's sub-categories.

You can refine the policy scope by specifying additional criteria for selecting APIs to which you want the policy applied.

➤ **To refine the scope of a run-time policy**

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. To filter the list to see just a list of the available policies, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, do the following:
 - a. Locate **Applicable Scopes**.
 - b. Choose **Runtime Policy** from the drop-down list.
 - c. Click the plus button next to the drop-down box or press Enter to add the scope **Runtime Policy** to the search recipe.

CentraSite displays the list of runtime policies that are available to you in the Search Results for Governance Rules page.

3. Select the policy whose scope you want to refine.

This opens the Run-Time Policy Details page.

4. In **Run-Time Policy Details (Step 1 of 3)**, locate the **Filter Criteria** section.
5. If you want to filter by Name or Description, do the following:
 - a. Select **Name** or **Description** in the first field.

- b. Select the comparison operator (for example, Equals, Contains, StartsWith) in the second field.
 - c. Specify the match string in the third field.
6. If you want to filter by the asset's classification attribute, do the following:
 - a. Select **Classification** in the first field.
 - b. Click **Browse** and select the category by which you want to filter APIs.
7. If you want to specify additional criteria, click the plus button and repeat steps 4 and 5.

Important:

If you specify multiple filters, the policy is applied only if the API matches *all the selection criteria* (that is, the selection criteria is combined using an AND operator, not an OR).

This displays the list of services in the **Run-Time Policy Details (Step 3 of 3)** page.

Assigning Actions to a Run-Time Policy

Note:

By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#). This section is not applicable, if the CentraSite run-time aspects are not enabled.

The **Message Flow** area on **Run-Time Policy Details (Step 1 of 3)** specifies the list of actions that you want CentraSite to execute when it enforces the run-time policy. CentraSite executes actions in the order in which they appear in the list.

The action list can include any built-in or custom actions that are compatible with the policy's scope.

Modifying Policy Action List

Note:

By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#). This section is not applicable, if the CentraSite run-time aspects are not enabled.

Pre-requisites:

To modify the action list of a run-time policy in CentraSite, you must have one of the following permissions:

- To modify the action list of a organization-specific policy, you must have the Manage Run-Time Policies permission for the organization specified in the scope of the original policy. For

example, if the original policy is scoped for organization ABC, you are not permitted to modify the action list of that policy unless you have permission to manage run-time policies for organization ABC.

- To modify the action list of a system-wide policy, you must have the Manage System-Wide Run-Time Policies permission.

Important:

If you belong to a role that includes the Manage System-Wide Run-Time Policies permission, you have the ability to modify the action list of CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be modified except under the direction of a technical representative from Software AG.

You modify the action list of a run-time policy using the Run-Time Policy Details page in CentraSite Business UI.

➤ **To modify the action list for a run-time policy**

1. In CentraSite Business UI activity bar, click **Governance Rules**.
2. To filter the list to see just a list of the available policies, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, do the following:
 - a. Locate **Applicable Scopes**.
 - b. Choose **Runtime Policy** from the drop-down list.
 - c. Click the plus button next to the drop-down box or press Enter to add the scope **Runtime Policy** to the search recipe.

CentraSite displays the list of runtime policies that are available to you in the Search Results for Governance Rules page.

3. Select the policy whose action list you want to modify.

This opens the Run-Time Policy Details page.
4. If the policy is active, deactivate it.

You cannot modify the action list of an active policy.
5. To add actions to the Run-Time Policy Details page, you simply drag and drop the individual actions from the **Policy Actions** area to the **Message Flow** area.
6. To remove an existing action from the Run-Time Policy Details page, proceed as follows:
 - a. Hover over the action you want to remove.

This causes a **Delete** icon to appear, that you can use for removing the action.

- b. Click **Delete**.
 - c. Repeat the above steps for each action that you want to remove.
7. Configure the parameter values for any new actions that you might have added to the list, or to make any necessary updates to the parameter values for existing actions.
 8. Click **Save**.
 9. When you are ready to put the policy into effect, activate and publish the policy.

Configuring Policy Action Parameters

Note:

By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#). This section is not applicable, if the CentraSite run-time aspects are not enabled.

To configure the action parameters for a run-time policy in CentraSite, you must have one of the following permissions:

- To configure the action parameters for a organization-specific policy, you must have the Manage Run-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to export that policy unless you have permission to manage run-time policies for organization ABC.
- To configure the action parameters for a system-wide policy, you must have the Manage System-Wide Run-Time Policies permission.

Policy actions have parameters that you must set to configure the behavior of the action at enforcement time.

Important:

Until the required parameters are set for all actions, you can not activate the policy or save the policy.

> To configure action parameters for a run-time policy

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. To filter the list to see just a list of the available policies, do the following:
 - a. Go to the advanced search panel.

- b. In the **Narrow Your Results** section, do the following:
 - a. Locate **Applicable Scopes**.
 - b. Choose **Runtime Policy** from the drop-down list.
 - c. Click the plus button next to the drop-down box or press Enter to add the scope **Runtime Policy** to the search recipe.

CentraSite displays the list of runtime policies that are available to you in the Search Results page.

3. Select the policy whose action parameters you want to configure.

This opens the Run-Time Policy Details page.

4. If the policy is active, deactivate it.

You cannot configure the action parameters of an active policy.

5. In the **Message Flow** area, do the following for each action in the list:

- a. Hover over an action name whose parameters you want to examine or configure.
- b. Click the **Configure** (⚙️) icon that is displayed to the right of the action name.
- c. In the **<action_name>** dialog box, set the parameters as necessary.

Note:

If you fail to specify all of the required parameters, CentraSite issues a red error icon. Hover over the error icon shows a hint with the error description.

- d. Click **Save** to save the parameter settings.

6. Click **Save**.

Note:

If you fail to specify all of the required parameters, CentraSite issues an error message with the description.

7. When you are ready to put the policy into effect, activate and publish the policy.

Viewing the List of Virtual Services to Which a Run-Time Policy Applies

Note:

By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time](#)

[Aspects](#) on page 951. This section is not applicable, if the CentraSite run-time aspects are not enabled.

You can view the list of services to which run-time policy is applicable by using the Run-Time Policy Details page in CentraSite Business UI.

To view the list of services for a run-time policy you must have the instance-level View permission for that particular policy. By default, all users have View permissions on the run-time policies in the registry.

Important:

The list only includes virtual services that are published to the specified gateways in the policy definition. Virtual services that are within the scope of the policy, but have not yet been published to the specified gateways, do not appear in this list.

➤ **To view the list of services applied to a run-time policy**

1. In CentraSite Business UI activity bar, click **Governance Rules**.
2. To filter the list to see just a list of the available policies, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, do the following:
 - a. Locate **Applicable Scopes**.
 - b. Choose **Runtime Policy** from the drop-down list.
 - c. Click the plus button next to the drop-down box or press Enter to add the scope **Runtime Policy** to the search recipe.

CentraSite displays the list of runtime policies that are available to you in the Search Results page.

3. Select the policy whose list of services you want to view.

This opens the Run-Time Policy Details page.

4. Navigate to **Run-Time Policy Details (Step 3 of 3)** page.

This displays the list of services that was generated based on the criteria you specified in the **Filters** section of **Run-Time Policy Details (Step 1 of 3)** page.

Deleting Run-Time Policies

Note:

By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time](#)

Aspects on page 951. This section is not applicable, if the CentraSite run-time aspects are not enabled.

To delete a run-time policy in CentraSite, you must have one of the following permissions:

- To delete a organization-specific policy, you must have the Manage Run-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to delete that policy unless you have permission to manage run-time policies for organization ABC.
- To delete a system-wide policy, you must have the Manage System-Wide Run-Time Policies permission.

You delete a policy to remove it from CentraSite permanently.

CentraSite is installed with a system-wide policy called *Check State Validation Policy for Policy*. This policy does not allow you to delete a policy unless the policy is in the New or Retired state.

In addition to being in the New or Retired state, the following conditions must also be met in order to delete a policy:

- The policy must not be in-progress.
- The policy must be inactive.
- You must have Full permission on the policy.

➤ To delete run-time policies

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. To filter the list to see just a list of the available policies, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, do the following:
 - a. Locate **Applicable Scopes**.
 - b. Choose **Runtime Policy** from the drop-down list.
 - c. Click the plus button next to the drop-down box or press Enter to add the scope **Runtime Policy** to the search recipe.

CentraSite displays the list of runtime policies that are available to you in the Search Results page.

3. If the policy you want to delete is active, deactivate it.

You cannot delete an active policy.

4. Select the check box for a single policy, or select the check boxes for multiple policies you want to delete.
5. On the actions bar of the Search Results page, click **Delete**.

You can also delete a single runtime policy from the actions bar of its details page.

Important:

If you have selected several policies where one or more of them are system-wide policies, you can use the **Delete** button to delete the policies. However, if you do not have the required permission for all of the selected policies, only policies for which you have permission for are deleted.

6. When you are prompted to confirm the delete operation, click **Yes**.

The selected policies are permanently removed from the CentraSite Registry Repository.

Managing Run-Time Policies through CentraSite Control

You create run-time policies and apply them to virtual services in order to govern the virtual services' run-time execution.

A run-time policy is a sequence of actions that are carried out by a policy-enforcement point (PEP) when a consumer requests a particular service through the PEP. The actions in a run-time policy perform activities such as identifying or authenticating consumers, validating digital signatures, and capturing performance metrics. You create run-time policies using CentraSite Control and store them in the CentraSite registry or repository.

When you create a run-time policy in CentraSite Control, you:

- Specify the PEP target type (for example, webMethods Mediator) to which you deploy the virtual services and their policies.
- Add run-time actions to the policy and configure their parameters. CentraSite provides a set of built-in run-time actions.
- Apply the policy to the desired virtual services.
- Activate the policy.

Creating a Run-Time Policy

To create a new run-time policy, you must have one of the following permissions in CentraSite:

- To create policies for a specific organization, you must have the Manage Run-Time Policies permission for that organization. By default, users in the CentraSite Administrator, Organization Administrator, or Policy Administrator role have this permission.
- To create system-wide policies (that is, policies that apply to all organization within an instance of CentraSite), you must have the Manage System-Wide Run-Time Policies permission. By

default, users in the CentraSite Administrator role and Operations Administrator role have this permission.

➤ **To create a run-time policy**

1. In CentraSite Control, go to **Policies > Run-Time**.

This displays a list of defined run-time policies in the Run-Time Policies page.

2. Click **Add Policy**.

3. In the **Policy Information** panel, specify the following fields:

In this field...	Specify...
Name	Type a name for the new policy. A policy name can contain any character (including spaces). A policy name does not need to be unique within the registry. However, to reduce ambiguity, you should avoid giving multiple policies the same name. As a best practice, Software AG recommends that you adopt appropriate naming conventions to ensure that policies are distinctly named within your organization.
Description	<i>Optional.</i> Type a description for the new policy. This description appears when a user displays a list of policies in the user interface.
Version	<i>Optional.</i> Specify a version identifier for the new policy.

Note:

The version identifier does not need to be numeric.

Examples:

```
0.0a
1.0.0 (beta)
Pre-release 001
V1-2007.04.30
```

The version identifier you type here is the policy's public, user-assigned version identifier. CentraSite also maintains an internal, system-assigned version number for the policy.

4. In the **Scope** panel, specify the required fields.

Scope refers to the set of properties that determine the target type, organization, and asset type to which the policy applies.

In this field...	Specify...
Target Type	The target type to which the policy is deployed. Select webMethods Integration Server (that is, the webMethods Mediator gateway type).

In this field... **Specify...**

Organization The organization to which the policy applies. Select **All** if you want to apply the policy to the specified services in all organizations.

Important:

Once you create a policy, its organizational scope is fixed and cannot be changed. That is, if you create a policy whose scope is specific to organization ABC, you cannot change its scope to make it system-wide or switch it to another organization. You must create a new policy and set its organizational scope as needed.

Asset Types The type of asset to which this policy applies. Select one of the following:

- Service
- XML Service
- REST Service
- Virtual Service
- Virtual XML Service
- Virtual REST Service

Note: CentraSite does not provide out-of-the-box policy-enforcement for web services.

5. In the **Apply Policy to Services that Meet the Following Criteria** panel, specify criteria that identify the virtual services to which the policy applies.

To target a policy for a particular set of virtual services, you refine the policy's scope by specifying additional selection criteria based on the virtual service's Name, Description, or Classification properties.

- a. select an attribute (Name, Description, or Classification) that identifies the services to which the policy applies.
- b. select an operator for the attribute (if applicable).
- c. Specify a value for the attribute (if applicable). Values are case-sensitive.
- d. If you need to specify multiple values or attributes, use the plus button to add multiple rows. For example, for the Classification attribute you might select multiple Taxonomy names. If you specify multiple criteria, they are connected by the AND operator.

After you save the policy, you see the generated list of services is displayed on the Policy Detail page's **Services** profile.

Note:

Keep the following in mind:

- If you specify no criteria, the policy applies to all virtual services.
- You can specify only *one* Name Equals <value> condition. However, you can specify multiple **Name Contains <value>** or **Name Starts With <value>** conditions.

CAUTION: CentraSite checks for policy conflicts when you deploy a virtual service to Mediator. If the service has only one policy applied to it (the policy you are applying here), that policy is deployed to Mediator, and Mediator executes the policy's run-time actions in the order in which they appear in the policy. However, if the service already has additional policies applied to it, a policy conflict might occur, which might cause unintended consequences. CentraSite informs you of policy conflicts.

6. Click **Next**.
7. In the **Available Actions** dialog, select the built-in actions that you want to include in the policy.

Keep the following points in mind when you select the actions for the policy:

- If you are using webMethods Mediator as your PEP, you must include the Identify Consumer built-in action (and optionally other identification actions) in order to identify or authenticate consumers.
- Ensure that the actions in the **Selected Actions** list appear in the order in which you want them to run when the policy is enforced. If necessary, use the control buttons on the right side of the list to place them in the correct order.

8. Click **Finish** to save the new (as yet incomplete) policy.

The Runtime Policy Detail page is displayed, showing details of the policy you just created.

9. Specify parameter values for each of the policy's actions as follows:
 - a. In the **Actions** profile, select the action whose parameters you want to set.
 - b. In the **Edit Action Parameters** page, set the parameters as necessary and click **Save**.
 - c. Click **Save** and then **Close**.

Icon	Description
	The action has required input parameters that have not yet been set.
	All of the action's required input parameters have been set.

Note:

This icon automatically appears for actions that have no input parameters.

The icons next to the actions in the **Parameters Set** column indicates whether the action parameters have been set.

10. If you want to allow other users to view, edit, or delete this policy, go to the **Policy Detail** page, select the **Permissions** profile, and assign permissions to those users. You do not see this profile unless you belong to a role that has the Manage Runtime Policies permission.
11. Activate the policy.

Activating a Run-Time Policy

To activate a run-time policy in CentraSite, you must have one of the following permissions:

- To activate a organization-specific policy, you must have the Manage Run-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to activate that policy unless you have permission to manage run-time policies for organization ABC.
- To activate a system-wide policy, you must have the Manage System-Wide Run-Time Policies permission.

A run-time policy is not eligible for deployment unless it is active. In other words, when CentraSite deploys a virtual service, it uses only active run-time policies to produce the effective policy for the virtual service. Whether a run-time policy is active or inactive is determined by its lifecycle state.

Run-time policies are governed by the same lifecycle model as design/change-time policies. The predefined lifecycle model for policies that is installed with CentraSite is made up of four states: New, Productive, Suspended and Retired. Under this lifecycle model, all policies enter the New state when they are initially created.

While a run-time policy is in the New state, it is inactive. To activate a policy, you change the policy's lifecycle state to the Productive state. This state change executes CentraSite's *Automatic Policy Activation* policy.

Note:

The *Automatic Policy Activation* policy is a hidden system policy. You cannot edit or delete this policy.

If a virtual service is already deployed to webMethods Mediator and is in the scope of an activated run-time policy, you must redeploy the virtual service to Mediator so that the actions are activated.

Note:

We recommend that you use the lifecycle model that CentraSite provides for policies “as-is”, even for a multi-stage deployment.

CentraSite does not begin enforcing a run-time policy until you *activate* it.

When you activate a run-time policy, keep the following points in mind:

- You are not allowed to activate the policy unless all of its parameters have been set.

- Some organizations require an approval to activate the policy. If your organization has an approval action associated with the activation of a policy, CentraSite does not activate the policy until the required approvals are obtained.
- If an earlier version of the policy is already active, CentraSite deactivates the old version before it activates the new one. When a policy becomes active, CentraSite begins enforcing it immediately.
- When a policy becomes active, CentraSite begins enforcing it immediately. You can suspend enforcement of a policy by switching it to the Suspended state.
- To activate a policy, you must have permission to change the policy to the Productive state. To successfully change a policy to the Productive state, you must also have the Modify permission on all virtual type services to which the policy is applied.
- Upon activation of a policy, CentraSite applies the active policy to all of the virtual APIs that are in the scope of this policy.
- You should be aware that a run-time policy is automatically published whenever a virtual API associated to it is published to the Mediator gateway.

The icon in the **Active** column on the **Policies > Run-Time** page indicates the policy's activation state as follows:

Icon	Description
	Policy is active.
	Policy is inactive.

The activation state of a policy is also reported next to the **State** field in the Run-Time Policy Details page.

1. In CentraSite Control, go to **Policies > Run-Time**.
This displays a list of defined run-time policies in the Run-Time Policies page.
2. Locate the policy you want to activate and select **Details** from its context menu.
This opens the Run-Time Policy Details page.
3. Examine the **Actions** profile and verify that all of the actions on this profile display the green checkmark icon in the **Parameters Set** column. If any of the actions display the red circle icon in this column, set their parameters before you continue.
4. In the **Policy Information** panel, click **Change State**. (If you do not see **Change State**, it is probably because you do not have permission to change the lifecycle state of a policy.)
5. In the **Change Lifecycle State** dialog box, select the **Productive** lifecycle state and click **OK**.

6. Examine the **State** field in the **Policy Information** panel to verify that the policy's state has been changed.

If this state change requires approval, the **State** field indicates that the policy is in the pending mode. CentraSite automatically switches the policy to the requested state (and activate the policy) after all the necessary approvals have been obtained.

Note:

While the policy is in pending mode, it cannot be edited.

7. After you activate the policy, users with the proper permissions can deploy the services to your PEP. At that time, CentraSite automatically validates the service's policies (for example, check for policy conflicts or other violations).

Deactivating a Run-Time Policy

You can deactivate a run-time policy by using the Run-Time Policy Details page. Alternatively, you can also deactivate a run-time policy by using the Run-Time Policies page in CentraSite Control.

To deactivate a run-time policy in CentraSite, you must have one of the following permissions:

- To deactivate a organization-specific policy, you must have the Manage Run-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to deactivate that policy unless you have permission to manage run-time policies for organization ABC.
- To deactivate a system-wide policy, you must have the Manage System-Wide Run-Time Policies permission.

Deactivating a run-time policy causes CentraSite to suppress enforcement of the policy. You usually deactivate a policy for the following reasons:

- To suspend enforcement of a particular policy (temporarily or permanently).
- To edit a policy (for example, to modify the scope of a policy or change its action list).

To deactivate a policy, you change the policy to the Suspended state. Switching the policy to this state triggers the *Automatic Policy Deactivation* policy, which deactivates the policy. Switching the policy to the Retired state also deactivates the policy, but you do not want to switch a policy to this state unless you intend to deactivate it permanently. After you place a policy in the Retired state, you cannot reactivate it.

When you deactivate a policy, CentraSite does not deactivate a policy if it is in the process of being executed. If you attempt to deactivate a policy while it is executing, your state change request fails. If this occurs, wait for a period time and then try to deactivate the policy again.

> To deactivate a run-time policy

1. In CentraSite Control, go to **Policies > Run-Time**.

This displays a list of defined run-time policies in the Run-Time Policies page.

2. Locate the policy you want to deactivate and select **Details** from its context menu.

This opens the Run-Time Policy Details page.

3. In the **Policy Information** panel, click **Change State**.
4. In the **Change Lifecycle State** dialog box, select the **Suspended** state (to deactivate it temporarily) or the **Retired** state (to deactivate it permanently).
5. Click **OK**.

Viewing Run-Time Policy List

By default, all users have View permissions on the run-time policies in the registry.

The Run-Time Policies page displays the run-time policies for all organizations, not just your own. It also includes system-wide policies.

> To view the policy list

1. In CentraSite Control, go to **Policies > Run-Time**.
2. To filter the list to view a subset of the available policies, type a partial string in the **Search** field.

CentraSite applies the filter to the **Name** column. The **Search** field is a type-ahead field, so as soon as you type any characters, the display is updated to show only those policies whose name contains the specified characters. The wildcard character % is supported.

3. To specify the sorting order, select either **Web Service**, **REST Service**, **XML Service**, **Virtual Service**, **Virtual REST Service** or **Virtual XML Service** from the **Browse by** list.

The Run-Time Policies page provides the following information about each policy:

Note:

Only the first six columns described below are displayed in this list by default. To display the additional columns, click **Select Columns**.

Column	Description
Name	Name of the policy.
Description	Description of the policy.
System Version	The system-assigned version identifier for the policy.
Organization	The organization to which the policy applies.

Column	Description						
	<table border="1"> <thead> <tr> <th>This value...</th> <th>Indicates that...</th> </tr> </thead> <tbody> <tr> <td>All</td> <td>The policy is system-wide and applies to all organizations.</td> </tr> <tr> <td><i>OrgName</i></td> <td>The policy applies to the specified organization.</td> </tr> </tbody> </table>	This value...	Indicates that...	All	The policy is system-wide and applies to all organizations.	<i>OrgName</i>	The policy applies to the specified organization.
This value...	Indicates that...						
All	The policy is system-wide and applies to all organizations.						
<i>OrgName</i>	The policy applies to the specified organization.						
State	The policy's current lifecycle state.						
Active	The policy's current enforcement state.						
	<table border="1"> <thead> <tr> <th>Icon</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>The policy is active. CentraSite enforces this policy when events within the scope of the policy occur.</td> </tr> <tr> <td></td> <td>The policy is inactive. Inactive policies exist in the registry, but they are not enforced.</td> </tr> </tbody> </table>	Icon	Description		The policy is active. CentraSite enforces this policy when events within the scope of the policy occur.		The policy is inactive. Inactive policies exist in the registry, but they are not enforced.
Icon	Description						
	The policy is active. CentraSite enforces this policy when events within the scope of the policy occur.						
	The policy is inactive. Inactive policies exist in the registry, but they are not enforced.						
Version	The user-assigned version identifier for the policy.						
Owner	The user to which the policy belongs.						

Viewing or Modifying a Run-Time Policy

To examine and modify the properties of a run-time policy in CentraSite, you must have one of the following permissions:

- To examine and modify the properties of a organization-specific policy, you must have the Manage Run-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to examine and modify the properties of that policy unless you have permission to manage run-time policies for organization ABC.
- To examine and modify the properties of a system-wide policy, you must have the Manage System-Wide Run-Time Policies permission.

Important:

If you belong to a role that includes the Manage System-Wide Run-Time Policies permission, you have the ability to modify CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be modified except under the direction of a technical representative from Software AG.

You cannot modify a policy while it is in the Productive state. To make changes to a policy, you can do any of the following:

- Create a new version of the policy, make the necessary changes to the new version and switch the new version to the Productive state when you are ready to put it into effect. Switching the new version to the Productive state will immediately put the previous version in the Retired

state. (The Retired state is an end state. After you place a policy in this state, you can no longer reactivate it.)

- Create a completely new policy that includes the required changes. When you are ready to put the new policy into effect, switch the old policy to the Suspended state and switch the new policy to the Productive state. When you are certain that you will no longer need to revert to the original policy, switch it to the Retired state.
- Switch the existing policy to the Suspended state, make the necessary changes to the policy and then switch it back to the Productive state. While the policy is in the Suspended state, it will not be enforced. (Because suspending the policy results in an enforcement gap, one usually does not use this approach in a production environment.)

➤ To view or modify the properties of a run-time policy

1. In CentraSite Control, go to **Policies > Run-Time**.

This displays a list of defined run-time policies in the Run-Time Policies page.

2. Locate the policy whose details you want to view or edit and select **Details** from its context menu.

This opens the Run-Time Policy Details page.

3. If the policy is active, deactivate it.

You cannot modify the details of an active policy.

4. On the Policy details page, examine or modify the properties as required.

Field or Profile	Description						
Name	The name of the policy. A policy name can contain any character (including spaces).						
Description	<i>Optional.</i> Additional comments about the policy.						
Version	The user-defined version number to be assigned to the new version. We recommend that you update the version number anytime you make significant modifications to a policy.						
State	The policy's current lifecycle state (for example, New, Productive, Suspended, Retired). This field also displays an icon that indicates the activation state of the policy:						
	<table border="1"> <thead> <tr> <th>Icon</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>The policy is active (that is, ready to be deployed to a PEP).</td> </tr> <tr> <td></td> <td>The policy is inactive.</td> </tr> </tbody> </table>	Icon	Description		The policy is active (that is, ready to be deployed to a PEP).		The policy is inactive.
Icon	Description						
	The policy is active (that is, ready to be deployed to a PEP).						
	The policy is inactive.						

Field or Profile	Description
Organization	The organization to which the policy belongs.
Owner	The user who created the policy.
System Version	The automatically-generated system version identifier for the policy.
Actions profile	The settings in this profile specify the actions that the PEP will execute when the policy is enforced.
Scope profile	The settings in this profile determine the services to which the policy is applied.
Services profile	Displays the list of web services and virtual services to which the policy applies.
Permissions profile	The settings in this profile identify which users can view, edit and delete the policy.

5. Click **Save** to save the updated policy.
6. When you are ready to put the policy into effect, activate the policy.

Setting Permissions on a Run-Time Policy

To set instance-level permissions on a run-time policy in CentraSite, you must have one of the following permissions:

- To set permissions on a organization-specific policy, you must belong to a role that has the Manage Run-Time Policies for the organization to which the policy belongs or have the Full instance-level permission on the policy itself.
- To set permissions on a system-wide policy, you must belong to a role that has the Manage System-Wide Run-Time Policies or have the Full instance-level permission on the policy itself.

Important:

If you belong to a role that includes the Manage System-Wide Run-Time Policies permission, you have the ability to modify permissions of CentraSite's predefined policies. However, you should not do this. These policies perform critical functions within the registry and must not be modified except under the direction of a technical representative from Software AG.

By default, all users have View permissions on the run-time policies in the registry.

Users who belong to a role that includes the Manage Run-Time Policies permission for an organization have Full permission on the policies that belong to the organization. Users who belong to a role that includes the Manage System-Wide Run-Time Policies permission, have Full permission on all system-wide policies. To enable other users to modify and delete policies, you must modify the policy's instance-level permission settings.

You can modify the instance-level permissions for a policy by executing a run-time policy or by specifying the permissions manually on the **Permissions** tab in CentraSite Control.

When setting permissions on policies, keep the following points in mind:

- You can assign permissions to any individual user or group defined in CentraSite.

Note:

If you give a user permission to view, edit or delete a policy, and you want that user to be able to perform these operations using CentraSite Control, ensure that the user belongs to a role that also has the Use the Policy UI permission.

Permission	Description
View	Enables users to see the policy in their policy list and view details for the policy.
Modify	Enables users to view and modify the properties of a policy (including the policy's scope and action list).
Full	Enables users to view, modify or delete the policy.

- The groups to which you can assign permissions include the following system-defined groups:

Group Name	Description
Users	All users within a specified organization.
Members	All users within a specified organization and its child organizations.
Everyone	All users of CentraSite <i>including guest users</i> (if your CentraSite permits access by guests).

- If a user is affected by multiple permission assignments, the user receive the union of all the assignments. For example, if group ABC has Modify permission on a policy and group XYZ has Full permission on the same policy, users that belong to both groups will, in effect, receive Full permission on the policy.

➤ To assign instance-level permissions to a run-time policy

1. In CentraSite Control, go to **Policies > Run-Time**.

This displays a list of defined run-time policies in the Run-Time Policies page.

2. Locate the policy whose permissions you want to modify and select **Details** from its context menu.

This opens the Run-Time Policy Details page.

3. If the policy is active, deactivate it.

You cannot modify the permission settings of an active policy.

4. On the Policy details page, click the **Permissions** tab.
5. To add users or groups to the **Users / Groups** list, do the following:
 - a. Click **Add Users / Groups**.
 - b. Select the users and groups to which you want to assign permissions.

If you want to filter the list, type a partial string in the **Search** field. CentraSite applies the filter to the **Users/Groups** column.

Examples

String	Description
b	Displays names that contain b
bar	Displays names that contain bar
%	Displays all users and groups

- c. Click **OK**.
6. To remove a user or group from the **Users / Groups** list, select the check box beside the group name or user ID and click **Delete**.
7. Use the **View**, **Modify** and **Full** check boxes to assign specific permissions to each user and group in the **Users / Groups** list as follows:

Permission	Allows the selected user or group to...
------------	---

View	View the policy.
-------------	------------------

Note:

Disabling this permission does not prevent a user from accessing the policy. CentraSite implicitly grants users View permission on all design/change-time policies within an instance of CentraSite. This implicit permission that CentraSite grants to a user cannot be not revoked by disabling the **View** permission on this tab.

Modify	View and edit the policy.
---------------	---------------------------

Full	View, edit, and delete the policy. This permission also allows the selected user or group to assign instance-level permissions to the policy.
-------------	---

8. Click **Save** to save the new permission settings.
9. When you are ready to put the policy into effect, activate the policy.

Viewing the List of Services to Which a Run-Time Policy Applies

To view the list of services for a run-time policy you must have the instance-level View permission for that particular policy. By default, all users have View permissions on the run-time policies in the registry.

Important:

The list only includes services that are in the deployable state (that is, services whose deployment state has been enabled by the Change Deployment State action in a design-time policy). Services that are within the scope of the policy, but have not yet been made deployable, do not appear in this list.

> To view the list of services

1. In CentraSite Control, go to **Policies > Run-Time**.

This displays a list of defined run-time policies in the Run-Time Policies page.

2. Locate the policy whose services you want to view and select **Details** from its context menu.

This opens the Run-Time Policy Details page.

3. Select the **Services** profile.

This displays the list of virtual services and web services that was generated based on the criteria you specified in the **Scope** profile.

4. To view details of a service, click its hyperlinked name.

Note:

To add or remove services to or from the list, return to the **Scope** profile and change the criteria in the **Apply Policy to Services that meet the following Criteria** panel.

Deleting Run-Time Policies

To delete a run-time policy in CentraSite, you must have one of the following permissions:

- To delete a organization-specific policy, you must have the Manage Run-Time Policies permission for the organization specified in the scope of the original policy. For example, if the original policy is scoped for organization ABC, you are not permitted to delete that policy unless you have permission to manage run-time policies for organization ABC.
- To delete a system-wide policy, you must have the Manage System-Wide Run-Time Policies permission.

You delete a policy to remove it from CentraSite permanently.

CentraSite is installed with a system-wide policy called *Check State Validation Policy for Policy*. This policy does not allow you to delete a policy unless the policy is in the New or Retired state.

In addition to being in the New or Retired state, the following conditions must also be met in order to delete a policy:

- The policy must not be in-progress.
- The policy must be inactive.
- You must have Full permission on the policy.

> To delete run-time policies

1. In CentraSite Control, go to **Policies > Run-Time**.

This displays a list of defined run-time policies in the Run-Time Policies page.

2. If the policy is active, deactivate it.

You cannot delete an active policy.

3. Right-click a policy you want to delete, and click **Delete**, or select the check boxes for multiple policies, click the **Actions** menu, and then click **Delete**.

4. Click **OK** in the confirmation dialog box.

Note:

When you delete a policy that is an intermediate version, CentraSite also deletes all previous versions of the policy.

System-Assigned and User-Assigned Version Identifiers

CentraSite maintains two version identifiers for a policy: a *system-assigned identifier* and a *user-assigned identifier*.

- The system-assigned identifier is a version number that CentraSite maintains for its own internal use. CentraSite automatically assigns this identifier to a policy when the policy is created. You cannot delete it or modify it. A policy's system-assigned identifier is numeric.

A policy's system-assigned version number is shown in the **System Version** column on the Run-Time Policies page and in the **System Version** field of the policy's detail page.

- The user-assigned identifier is an optional identifier that you can assign to distinguish a specific version of a policy. This identifier does not need to be numeric. For example, you may use a value such as V2.a (beta) to identify a version.

A policy's user-assigned version number is shown in the **Version** column on the Run-Time Policies page and in the **Version** field of the policy's detail page.

Built-In Run-Time Actions Reference (CentraSite Business UI)

This section describes the built-in run-time actions that you can include in run-time policies for Virtual Services. You use these actions only when you are using CentraSite Business UI to create run-time policies for Virtual Services.

Summary of Actions in the Request Handling Category

The following action templates are available in the Request Handling category:

Request Handling is the process of receiving and transforming the incoming message from a client into the custom format as expected by the native API.

Protocol > Require JMS	Specifies the JMS protocol to be used for the API to accept and process the requests.
Protocol > Require HTTP / HTTPS	Specifies the protocol (HTTP or HTTPS) and SOAP format (for a SOAP-based API) to be used to accept and process the requests.
Request Transformation	Invokes an XSL transformation in the SOAP request before it is submitted to the native API.
Invoke webMethods Integration Server	Invokes a webMethods Integration Server service to pre-process the request before it is submitted to the native API.
Enable REST Support	Enables REST support for an existing SOAP based API by exposing the API both as a SOAP based API and a REST based API.
Set Media Type	Specifies the content type for a REST request received from a client if the content type header is not specified.

Summary of Actions in the Policy Enforcement Category

Policy Enforcement is the process of enforcing the adherence to real-time policy compliance identifying or authenticating, monitoring, auditing, and measuring and collecting result statistics for an API.

CentraSite provides various categories of policy enforcement actions.

JMS Routing Actions

JMS Routing actions route the incoming message to an API over JMS. For example, to a JMS queue where an API can then retrieve the message asynchronously.

- JMS Routing Rule** Specifies a JMS queue to which the Mediator is to submit the request, and the destination to which the native API is to return the response.
- Set Message Properties** Specifies JMS message properties to authenticate client requests before submitting to the native APIs.
- Set JMS Headers** Specifies JMS headers to authenticate client requests before submitting to the native APIs.

Logging and Monitoring Actions

Logging and Monitoring actions monitor and collect information about the number of messages that were processed successfully or failed, the average execution time of message processing, and the number of alerts associated with an API.

- Log Invocation** Logs request or response payloads to a destination you specify.
- Monitor Service Performance** This action provides the same functionality as Monitor Service Level Agreement but this action is different because it enables you to monitor the API's run-time performance for all clients. This action monitors a user-specified set of run-time performance conditions for an API, and sends alerts to a specified destination when these performance conditions are violated.
- Monitor Service Level Agreement** Specifies a Service Level Agreement (SLA) including a set of conditions to define the level of performance that a specified client should expect from an API.

Routing Actions

Routing actions route the incoming message (for example, directly to the API, or routed according to the routing rules, or routed to a pool of servers for the purpose of load balancing and failover handling).

- Straight Through Routing** Routes the requests directly to a native endpoint that you specify.
- Content Based Routing** Route requests to different endpoints based on specific criteria that you specify.
- Load Balancing and Failover Routing** Routes the requests across multiple endpoints.
- Set Custom Headers** Specifies the HTTP headers for the outgoing message to the native service.

Context Based Routing Route requests to different endpoints based on specific values that appear in the request message.

Dynamic Routing Route requests to a dynamic URL based on specific criteria that you specify.

Security Actions

Security actions provide client validation (through WSS X.509 certificates, WSS username tokens, and so on), confidentiality (through encryption) and integrity (through signatures) for request and response messages.

For the client validation, Mediator maintains a list of consumer applications specified in CentraSite that are authorized to access the API published to Mediator. Mediator synchronizes this list of consumer applications through a manual process initiated from CentraSite.

Generally speaking there are two different lists of consumers in the Mediator:

- **List of Registered Consumers**

List of users and consumer applications (represented as Application assets) who are registered as consumers for the API in CentraSite, and available in the Mediator.

- **List of Global Consumers**

List of all users and consumer applications (represented as consumers) available in the Mediator.

Mediator provides Evaluate actions that you can include in a message flow to identify and validate clients, and then configure their parameters to suit your needs. You use these Evaluate actions to perform the following actions:

- Identify the clients who are trying to access the APIs (through IP address or hostname).
- Validate the client's credentials.

Evaluate Client Certificate for SSL Connectivity Mediator validates the client's certificate that the client submits to the API in CentraSite. The client certificate that is used to identify the client is supplied by the client to the Mediator during the SSL handshake over the transport layer.

Evaluate Hostname

- Mediator tries to identify the client against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available consumers in Mediator).
- Mediator tries to validate the client's hostname against the specified list of consumers in the Integration Server on which Mediator is running.

Evaluate HTTP Basic Authentication

- Mediator tries to identify the client against either the Registered Consumers list (the list of registered consumers in Mediator)

or the Global Consumers list (the list of available consumers in Mediator).

- Mediator tries to validate the client's authentication credentials contained in the request's Authorization header against the specified list of consumers in the Integration Server on which Mediator is running.

Evaluate IP Address

- Mediator tries to identify the client against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available consumers in Mediator).
- Mediator tries to validate the client's IP address against the specified list of consumers in the Integration Server on which Mediator is running.

Evaluate KerberosToken

Mediator tries to authenticate the client based on the Kerberos token and the authenticated client principal name is verified with the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available consumers in Mediator).

Evaluate OAuth2 Token

- Mediator tries to identify the client against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available consumers in Mediator).
- Mediator tries to validate the client's OAuth access token against the specified list of consumers in the Integration Server on which Mediator is running.

Evaluate WSS Username Token

Applicable only for SOAP APIs.

- Mediator tries to identify the client against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available consumers in Mediator).
- Mediator tries to validate the client's WSS username token against the specified list of consumers in the Integration Server on which Mediator is running.

Evaluate WSS X.509 Certificate

Applicable only for SOAP APIs.

- Mediator tries to identify the client against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available consumers in Mediator).

- Mediator tries to validate the client's WSS X.509 token against the specified list of consumers in the Integration Server on which Mediator is running.
- Evaluate XPath Expression**
 - Mediator tries to identify the client against either the Registered Consumers list (the list of registered consumers in Mediator) or the Global Consumers list (the list of available consumers in Mediator).
 - Mediator tries to validate the client's XPath expression against the specified list of consumers in the Integration Server on which Mediator is running.
- Require Encryption**

Applicable only for SOAP APIs.

This policy requires that a request's XML element, which is represented by an XPath expression, or parts of SOAP request such as SOAP body or SOAP headers to be encrypted.
- Require Signing**

Applicable only for SOAP APIs.

This policy requires that a request's XML element, which is represented by an XPath expression, or parts of SOAP request such as SOAP body or SOAP headers to be signed.
- Require SSL**

Applicable only for SOAP APIs.

This policy requires that requests be sent through SSL client certificates.
- Require Timestamps**

Applicable only for SOAP APIs.

This policy requires that timestamps be included in the request header. Mediator checks the timestamp value against the current time to ensure that the request is not an old message. This serves to protect your system against attempts at message tampering, such as replay attacks.
- Require WSS SAML Token**

Applicable only for SOAP APIs.

This policy uses a WSS Security Assertion Markup Language (SAML) assertion token to validate API clients.
- Validate SAML Audience URIs**

This policy validates the Audience Restriction in the Conditions section of the SAML assertion. It verifies if any valid Audience URI within a valid Condition element in the SAML assertion matches with any of the configured URIs.
- Allow Anonymous Usage**

This policy allows all incoming requests to access the API without a restriction.

Traffic Management Action

Throttling Traffic Optimization	Limits the number of service invocations during a specified time interval, and sends alerts to a specified destination when the performance conditions are violated. You can use this action to avoid overloading the back-end services and their infrastructure, to limit specific clients in terms of resource usage, and so on.
Service Result Cache	Enables caching of the results of SOAP and REST API invocations.

Summary of Actions in the Validation Category

Validate Schema	Validates all XML request and response messages against an XML schema referenced in the WSDL.
------------------------	---

Summary of Actions in the Outbound Authentication Category

Outbound authentication actions are used to set the client credentials to access the native API.

HTTP Basic Authentication	Used when native API enforces basic authentication. Based on the modes selected, Mediator either uses configured basic authentication credentials to invoke a native service or it uses credentials from the authorization header of the incoming request to access the native API.
NTLM Authentication	Used when native API enforces NTLM authentication. Based on the modes selected, Mediator either uses configured authentication credentials to obtain the NTLM token to invoke the native service or it uses credentials from the authorization header of the incoming request to obtain the NTLM token to access the native API.
OAuth2 Authentication	Used when native API enforces OAuth authorization. Based on the modes selected, Mediator either uses configured OAuth token to invoke the native service or it uses the OAuth token of the incoming request to access native service.
Kerberos Authentication	Used when a service provider wants a web service client that does not have the ability to generate the Kerberos token to access a service enforced with the Kerberos policy. It is also used when service provider wants a web service client to access a service enforced with the Kerberos policy.
SAML Authentication	Used when a native API enforces SAML authentication. Based on the modes selected, Mediator either uses the WSS Username

mode or the Kerberos Over Transport mode to get the SAML assertion token and access the native API.

Summary of Actions in the Response Handling Category

Response Handling is the process of transforming the response message coming from the native API into the custom format as expected by the client.

Response Transformation	Invokes an XSL transformation in the response payloads from XML format to the format required by the client.
Invoke webMethods Integration Server	Invokes a webMethods Integration Server service to process the response from the native API before it is returned to the client.
Set Media Type	Specifies the content type for a REST response to the client if the content type header is not specified.

Summary of Actions in the Error Handling Category

Error Handling is the process of passing an exception message which has been issued as a result of a run-time error to take any necessary actions.

Conditional Error Processing	Returns a custom error message (and the native provider's service fault content) to the client when the native provider returns a service fault.
-------------------------------------	--

Effective Policies

When you publish an API to Mediator, CentraSite automatically validates the API's policy enforcement workflow to ensure that:

CentraSite informs you of any violation, and you have to correct the violations before publishing the API.

When you publish an API to Mediator, CentraSite combines the actions specified within the proxy API's enforcement definition, and generates what is called the effective policy for the API. For example, suppose your API is configured with two run-time actions: one that performs a logging action and another that performs a security action. When you publish the API, CentraSite automatically combines the two actions into one effective policy. The effective policy, which contains both the logging action and the security action, is the policy that CentraSite actually publishes to Mediator with the API.

When CentraSite generates the effective policy, it validates the resulting action list to ensure that:

- Any action that appears in a single message flow multiple times is allowed to appear multiple times.

For those actions that can appear in a message flow only once (for example, Evaluate IP Address), Mediator selects only one, which might cause problems or unintended results.

- All action dependencies are properly met. That is, some actions must be used in conjunction with another particular action.

If the list contains conflicts or inconsistencies, CentraSite resolves them according to Policy Resolution Rules.

The effective policy that CentraSite produces for an API is contained in an object called a virtual service definition (VSD). The VSD is given to Mediator when you publish the API. After you publish an API to Mediator, you can view its VSD (and thus examine the effective policy that CentraSite generated for it) from the Mediator user interface.

The following table shows:

- Action is WS-Security Policy 1.2 compliant.
- Action dependencies, that is, whether an action must be used in conjunction with another particular action.
- Action exclusives, that is, whether an action cannot be used in conjunction with another particular action.
- Action occurrences, that is, whether an action can occur once or multiple times within a message flow stage. An action can occur multiple times in a policy if the selection criteria is combined using an AND operator (not an OR operator).

Action

Require HTTP / HTTPS	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	Require JMS
	Once or multiple in a policy?	Once

heading-no-quotes	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	Require HTTP / HTTPS
	Once or multiple in a policy?	Once

Request Transformation	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Multiple

Action

Invoke webMethods Integration Server	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Multiple
Enable REST Support	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	WS-Security based actions
	Once or multiple in a policy?	Once
Set Media Type	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Once
Require SSL	WS-Security Policy Compliant	Yes
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Once
Require WSS SAML Token	WS-Security Policy Compliant	Yes
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Once
Require Signing	WS-Security Policy Compliant	Yes
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Once
Require Encryption	WS-Security Policy Compliant	Yes
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Once

Action

Require Timestamps	WS-Security Policy Compliant	Yes
	Dependency Requirement	At least one of the following actions: <ul style="list-style-type: none"> ■ Evaluate WSS Username Token ■ Evaluate WSS X.509 Certificate ■ Require Signing ■ Require Encryption
	Mutually Exclusive	None
	Once or multiple in a policy?	Once
Evaluate Kerberos Token	WS-Security Policy Compliant	If you Evaluate Kerberos Token at: <ul style="list-style-type: none"> ■ Message Level: Yes ■ Transport Level: No
	Dependency Requirement	None
	Mutually Exclusive	No
	Once or multiple in a policy?	Once
Evaluate OAuth2 Token	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Once
HTTP Basic Authentication	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	<ul style="list-style-type: none"> ■ Evaluate OAuth2 Authentication ■ OAuth2 Authentication ■ NTLM Authentication
	Once or multiple in a policy?	Once

Action

Evaluate WSS Username Token	WS-Security Policy Compliant	Yes
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Once
Evaluate WSS X.509 Certificate	WS-Security Policy Compliant	Yes
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Once
Evaluate IP Address	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Once
Evaluate XPath Expression	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Once
Evaluate Hostname	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Once
Evaluate Client Certificate for SSL Connectivity	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Once
Log Invocation	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Once

Action

Monitor Service Level Agreement	WS-Security Policy Compliant	No
	Dependency Requirement	At least one Evaluate action, or the Require WSS SAML Token.
	Mutually Exclusive	None
	Once or multiple in a policy?	Multiple
Monitor Service Performance	WS-Security Policy Compliant	No
	Dependency Requirement	At least one Evaluate action, or the Require WSS SAML Token.
	Mutually Exclusive	None
	Once or multiple in a policy?	Multiple
Throttling Traffic Optimization	WS-Security Policy Compliant	No
	Dependency Requirement	At least one of the Evaluate actions, or the Require WSS SAML Token, provided the Alert for Consumer Applications value is specified.
	Mutually Exclusive	None
	Once or multiple in a policy?	Multiple
Validate Schema	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Once
Validate SAML Audience URIs	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Once
HTTP Basic Authentication	WS-Security Policy Compliant	No
	Dependency Requirement	At least one Routing based action.
	Mutually Exclusive	<ul style="list-style-type: none"> ■ NTLM Authentication ■ OAuth2 Authentication

Action

		<ul style="list-style-type: none"> ■ JMS Routing Rule ■ Evaluate OAuth2 Authentication
	Once or multiple in a policy?	Once
NTLM Authentication	WS-Security Policy Compliant	No
	Dependency Requirement	At least one Routing based action.
	Mutually Exclusive	<ul style="list-style-type: none"> ■ HTTP Basic Authentication ■ OAuth2 Authentication ■ JMS Routing Rule ■ Evaluate HTTP Basic Authentication ■ Evaluate OAuth2 Authentication
	Once or multiple in a policy?	Once
OAuth2 Authentication	WS-Security Policy Compliant	No
	Dependency Requirement	At least one of the Routing actions.
	Mutually Exclusive	<ul style="list-style-type: none"> ■ HTTP Basic Authentication ■ NTLM Authentication ■ JMS Routing Rule ■ Evaluate HTTP Basic Authentication
	Once or multiple in a policy?	Once
Kerberos Authentication (Outbound Scenarios)	WS-Security Policy Compliant	If you Evaluate Kerberos Token at: <ul style="list-style-type: none"> ■ Message Level: Yes ■ Transport Level: No
	Dependency Requirement	Only the Evaluate HTTP Basic Authentication policy is enforced and the <code>Authenticate User</code> option is selected.

Action

	Mutually Exclusive	None
	Once or multiple in a policy?	Once
JMS Routing Rule	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	Routing based actions
	Once or multiple in a policy?	Once
SAML Authentication	WS-Security Policy Compliant	Yes
	Dependency Requirement	None
	Mutually Exclusive	<ul style="list-style-type: none"> ■ HTTP Basic Authentication ■ NTLM Authentication ■ JMS Routing Rule ■ Evaluate HTTP Basic Authentication
	Once or multiple in a policy?	Once
Set Message Properties	WS-Security Policy Compliant	No
	Dependency Requirement	JMS Routing Rule
	Mutually Exclusive	Routing based actions
	Once or multiple in a policy?	Once
Set JMS Headers	WS-Security Policy Compliant	No
	Dependency Requirement	JMS Routing Rule
	Mutually Exclusive	Routing based actions
	Once or multiple in a policy?	Once
Straight Through Routing	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	Routing based actions
	Once or multiple in a policy?	Once
Content Based Routing	WS-Security Policy Compliant	No
	Dependency Requirement	None

Action

	Mutually Exclusive	Routing based actions
	Once or multiple in a policy?	Once
Load Balancing and Failover Routing	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	Routing based actions
	Once or multiple in a policy?	Once
Context Based Routing	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	Routing based actions
	Once or multiple in a policy?	Once
Set Custom Headers	WS-Security Policy Compliant	No
	Dependency Requirement	At least one Routing based action.
	Mutually Exclusive	None
	Once or multiple in a policy?	Once
Response Transformation	WS-Security Policy Compliant	No
	Dependency Requirement	None
	Mutually Exclusive	None
	Once or multiple in a policy?	Multiple
Dynamic Routing	WS-Security Policy Compliant	No
	Dependency Requirement	If Context Variable is selected as the Route using option then 'Invoke webMethods Integration Server' must be configured as part of Request Handling to set the Context Variable for ROUTING_ENDPOINT.
	Mutually Exclusive	<ul style="list-style-type: none"> ■ Straight Through Routing ■ Content Based Routing ■ Load Balancing and Failover Routing

Action

	■ Context Based Routing
	Once or multiple in a policy? Once
Invoke webMethods Integration Server	WS-Security Policy Compliant No
	Dependency Requirement None
	Mutually Exclusive None
	Once or multiple in a policy? Multiple
Conditional Error Processing	WS-Security Policy Compliant No
	Dependency Requirement None
	Mutually Exclusive None
	Once or multiple in a policy? Once
Validate SAML Audience URIs	WS-Security Policy Compliant No
	Dependency Requirement None
	Mutually Exclusive None
	Once or multiple in a policy? Once

Usage Cases for Identifying or Authenticating Consumers

When deciding which type of identifier to use to identify a consumer application, consider the following points:

- Whatever identifier you select to identify a consumer application, it must be unique to the application. Identifiers that represent user names are often not suitable because the identified users might submit requests for multiple applications.
- Identifying applications by IP address or host name is often a suitable choice, however, it does create a dependency on the network infrastructure. If a consumer application moves to a new machine, or its IP address changes, you must update the identifiers in the application asset.
- Using X.509 certificates or a custom token that is extracted from the SOAP message itself (using an XPATH expression), is often the most trouble-free way to identify a consumer application.

Following are some common combinations of actions used to authenticate or identify consumers:

- **Scenario 1: Identify consumers by IP address or host name**
 - The simplest way to identify consumers is to use the Identify Consumer action and set its Identify User Using parameter to specify either a host name or an IP address (or a range of IP addresses).

■ Scenario 2: Authenticate consumers by HTTP authentication token

Use the following actions:

- Identify Consumer action and set its `Identify User Using` parameter to HTTP Authentication Token (to identify consumers using the token derived from the HTTP header).
- Require HTTP Basic Authentication.
- Additionally, you can use one or both of the following:
 - Authorize User action (to authorize a list of users and groups registered in the Integration Server on which Mediator is running).
 - Authorize Against Registered Consumers action (to authorize consumer applications against all Application assets registered as consumers for a service in CentraSite).

■ Scenario 3: Authenticate consumers by WS-Security authentication token

Use the following actions:

- Identify Consumer action, and set its `Identify User Using` parameter to WS-Security Authentication Token (to identify consumers using the token derived from the WSS Header).
- Require WSS Username Token action.
- Additionally, you can use one or both of the following:
 - Authorize User action (to authorize a list of users and groups registered in the Integration Server on which Mediator is running).
 - Authorize Against Registered Consumers action (to authorize consumer applications against all Application assets registered as consumers for a service in CentraSite).

■ Scenario 4: Authenticate consumers by WSS X.509 token

- Identify Consumer action, and set its `Identify User Using` parameter to Consumer Certificate (to identify consumers using the WSS X.509 token).
- Require WSS X.509 Token action.
- Require SSL action.

Built-in Actions for Run-Time Policies (CentraSite Business UI)

This section provides an alphabetic list of the built-in run-time actions you can include in the run-time governance rules for APIs.

Allow Anonymous Usage

When an API is not configured with any of the **Security** policy actions, Mediator or API Gateway routes the incoming requests directly to the native API, irrespective of the client credentials passed in the request.

When an API is configured with a **Security** policy action, Mediator or API Gateway routes the incoming requests to the native API that are successfully identified using the client credentials passed in the request. When this API is also configured with the Allow Anonymous Usage policy action, Mediator or API Gateway routes all of the incoming requests to the native API. However, the successfully identified requests are grouped under the respective identified consumer, and all unidentified requests are grouped under a common consumer named as unknown.

With this action, you can perform the following list of consumer-specific actions, while also allowing the requests to pass through to the native API:

- View the runtime events for a particular consumer.
- Monitor the Service Level Agreement for a few consumers and send an alert email based on specific criteria's, for example, request count, availability, and so on.
- Throttle requests from a particular consumer, and restrict requests from that consumer if the number of requests to the API reaches the configured hard limit during a specified period of time.

Input Parameters

Allow Anonymous Usage (Boolean). Specifies whether to allow all the incoming requests to access the API, without restriction.

Value	Description
True	(Default). Allows all of the incoming requests to access an API even if the request is not identified using the Security policies that are configured for that API.
False	Allows only the incoming requests that are identified using the configured Security policies to access the API.

Content Based Routing

If you have a native API that is hosted at two or more endpoints, you can use the Content Based Routing to route specific types of messages to specific endpoints.

You can route messages to different endpoints based on specific values that appear in the request message.

When this action is configured for a proxy API, the requests are routed according to the routing rules you create. That is, they are routed based on the successful evaluation of one or more XPath expressions that are constructed utilizing the content of the request payload. For example, a routing rule might allow requests for half of the methods of a particular service to be routed to Endpoint A, and the remaining methods to be routed to Endpoint B.

Input Parameters

Default Route To (URI). Type the URL of the native API endpoint to route the request to in case all routing rules evaluate to False. For example:

```
http://mycontainer/creditCheckService
```

Click the **Configure Endpoint Properties** icon (next to the **Default Route To** field) to configure a set of properties for the specified endpoint.

Alternatively, Mediator offers Local Optimization capability if the native endpoint is hosted on the same Integration Server as Mediator. With local optimization, API invocation happens in-memory and not through a network hop.

Specify the native API in either of the following forms:

```
local://<Service-full-path>
```

OR

```
local://<server>:<port>/ws/<Service-full-path>
```

For example:

```
local://MyAPIFolder:MyLocalAPI
```

which points to the endpoint API `MyLocalAPI` which is present under the folder `MyAPIFolder` in Integration Server.

Note:

Local Optimization is not applicable to REST APIs.

Add Routing Rule (button) Click the **Add Routing Rule** button and complete the **Routing Rule** dialog box as follows.

Field	Description
XPath Expression	An argument to evaluate the XPath expression contained in the request.
Namespace	The namespace declaration indicated by the Prefix and URI reference text boxes.
HTTP Methods	The HTTP operations (CUSTOM, DELETE, GET, PATCH, POST, PUT) to perform on the resource. (The HTTP Methods list displays the list of supported HTTP methods.)

	Route To	Type the URL of the Native Service endpoint to route the request to, if the above rule criteria are met.								
	Configure Endpoint Properties (icon)	Configure the following set of properties for the specified endpoint individually.								
Configure Endpoint Properties (icon) 	(Optional). This icon displays the Endpoint Properties dialog box that enables you to configure a set of properties for the Mediator to route incoming requests to the native API as follows:									
	SOAP Optimization Method	(Only for SOAP-based APIs). Mediator can use the following optimization methods to parse SOAP requests to the native API:								
	<table border="1"> <thead> <tr> <th data-bbox="609 672 803 724">Value</th> <th data-bbox="803 672 1352 724">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="609 724 803 850">MTOM</td> <td data-bbox="803 724 1352 850">Mediator uses the Message Transmission Optimization Mechanism (MTOM) to parse SOAP requests to the API.</td> </tr> <tr> <td data-bbox="609 850 803 976">SwA</td> <td data-bbox="803 850 1352 976">Mediator uses the SOAP with Attachment (SwA) technique to parse SOAP requests to the API.</td> </tr> <tr> <td data-bbox="609 976 803 1102">None</td> <td data-bbox="803 976 1352 1102"><i>Default.</i> Mediator does not use any optimization method to parse the SOAP requests to the API.</td> </tr> </tbody> </table>	Value	Description	MTOM	Mediator uses the Message Transmission Optimization Mechanism (MTOM) to parse SOAP requests to the API.	SwA	Mediator uses the SOAP with Attachment (SwA) technique to parse SOAP requests to the API.	None	<i>Default.</i> Mediator does not use any optimization method to parse the SOAP requests to the API.	
Value	Description									
MTOM	Mediator uses the Message Transmission Optimization Mechanism (MTOM) to parse SOAP requests to the API.									
SwA	Mediator uses the SOAP with Attachment (SwA) technique to parse SOAP requests to the API.									
None	<i>Default.</i> Mediator does not use any optimization method to parse the SOAP requests to the API.									
		<p>Note: Keep the following points in mind:</p> <ul style="list-style-type: none"> ■ Bridging between SwA and MTOM is not supported. If a client sends a SwA request, Mediator can only forward SwA to the native API. The same is true for MTOM, and applies to responses received from the native API. That is, a SwA or MTOM response received by Mediator from a native API is forwarded to the client using the same format it received. ■ When sending SOAP requests that do not contain a MTOM or SWA attachment to a native API that returns an MTOM or SWA response, the request 'Accept' header must be set to multipart/related. This is necessary so Mediator knows how to parse the response properly. 								
	HTTP Connection Timeout	(Optional). (Number). The time interval (in seconds) after which a connection attempt timeouts. If a value 0 is specified (or if the value is not specified), Mediator uses the value specified in the <code>Connection Timeout</code> field (in the								

Integration Server Administrator, go to **Settings > Extended**). Default: 30 seconds.

Read Timeout (Optional). (Number). The time interval (in seconds) after which a socket read attempt will timeout.

The precedence of the `Read Timeout` configuration is as follows:

1. If a value is specified for the `Read Timeout` field in the routing endpoint alias, Mediator will use the value specified in the **Runtime Alias > Endpoint Alias > Endpoint Properties > Read Timeout** field. The read timeout value defined at an alias level takes precedence over the timeout values defined at an API level and the global configuration.
2. If a value 0 is specified (or if the value is not specified) for the `Read Timeout` field in the routing endpoint alias, then Mediator will use the value specified in the `Read Timeout` field of this routing action. The read timeout value defined at an API level takes precedence over the global configuration.
3. If a value 0 is specified (or if the value is not specified) for the `Read Timeout` field in this routing action (at an API level), then Mediator will use the value of the global property `pg.endpoint.readTimeout` located in the file `Integration Server_directory\packages\WmMediator\config\resources\pg-config.properties` (in the Mediator Administration console, go to **Settings > Extended Settings > pg.endpoint.readTimeout** property.).

Note:

If a value for the `Read Timeout` configuration is not specified in any of the above configuration parameters, then Mediator will use the default 30 seconds.

SSL Configuration (Optional). To enable SSL client authentication that Mediator uses to authenticate incoming requests for the native API, you must specify values for both the `Client Certificate Alias` field and the `IS Keystore Alias` field. If you specify a value for only one of these fields, a deployment error occurs.

Note:

SSL client authentication is optional; you may leave both fields blank.

Prerequisites: You must set up the key alias and keystore properties in the Integration Server. For the procedure, see *webMethods Integration Server Administrator's Guide*.

You use these properties to specify the following fields:

Value	Description
Client Certificate Alias	The client's private key to be used for performing SSL client authentication.
IS Keystore Alias	The keystore alias of the instance of Integration Server on which Mediator is running. This value (along with the value of Client Certificate Alias) is used for performing SSL client authentication.
WS Security Header	(Only for SOAP-based APIs). Indicates whether Mediator should pass the WS-Security headers of the incoming requests to the native API.

Value	Description
Remove processed security headers	(Default). Removes the security header if it is processed by Mediator (that is, if Mediator processes the header according to the API's security run-time action). Mediator does not remove the security header if both of the following conditions are true: 1) Mediator did not process the security header, and 2) the <code>mustUnderstand</code> attribute of the security header is 0 or false).
Pass all security headers	Passes the security header, even if it is processed by Mediator (that is, even if Mediator processes the header according to the API's security action).

Context Based Routing

If you have a native API that is hosted at two or more endpoints, you can use the Context Based Routing to route specific types of messages to specific endpoints.

When this action is configured for a proxy API, the requests are routed according to the routing rules you create. A routing rule specifies where requests should be routed, and the criteria by which they should be routed there. For example, requests can be routed according to certain clients, certain dates and times, or according to requests that exceed or fall below a specified metric (Total Count, Success Count, Fault Count, and so on). You can create one or more rules.

Input Parameters

Default Route To (URI). Type the URL of the native API endpoint to route the request to in case all routing rules evaluate to False. For example:

```
http://mycontainer/creditCheckService
```

Click the **Configure Endpoint Properties** icon (next to the **Default Route To** field) to configure a set of properties for the specified endpoint.

Alternatively, Mediator offers Local Optimization capability if the native endpoint is hosted on the same Integration Server as Mediator. With local optimization, API invocation happens in-memory and not through a network hop.

Specify the native API in either of the following forms:

```
local://<Service-full-path>
```

OR

```
local://<server>:<port>/ws/<Service-full-path>
```

For example:

```
local://MyAPIFolder:MyLocalAPI
```

which points to the endpoint API `MyLocalAPI` which is present under the folder `MyAPIFolder` in Integration Server.

Note:

Local Optimization is not applicable to REST APIs.

Add Routing Rule (button) Click the **Add Routing Rule** button and complete the **Routing Rule** dialog box as follows.

Field	Description
Name	Name of the routing rule.
HTTP Methods	The HTTP operations (CUSTOM, DELETE, GET, PATCH, POST, PUT) to perform on the resource. (The HTTP Methods list displays the list of supported HTTP methods.)
Condition	The context variables for processing client requests. The Variable list displays the list of supported variables - Consumer, Custom Context Variable, Date, IP Address, IPv6

Address, Predefined Context Variable, and Time.

- **Consumer** - Type the name of the consumer application in the text box.
- **Custom Context Variable**
 - Select a data type - **String** or **Integer**.
 - Select an operator - **Greater Than, Less Than, Not Equal To, or Equal To**.
 - Type a custom variable name in the **Custom Context** text box.
 - Type a value in the **Variable Value** text box.
- **Date** - Select an operator **Before** or **After**. Type a date value in the text box.
- **Time** - Select an operator **Before** or **After**. Type a time value in the text box.
- **IP Address / IPv6 Address** - Type an IP address range in the **From** and **To** text boxes.
- **Predefined Context Variable**
 - Select a data type - **String** or **Integer**.
 - Select an operator - **Greater Than, Less Than, Not Equal To, or Equal To**.
 - Select a predefined context variable from the **Predefined Context** list.
 - Type a value in the **Variable Value** text box.

Note:

Keep the following points in mind:

- For a list of the predefined context variables, see [“Context Variables in Virtual Services”](#) on page 1157.

		<ul style="list-style-type: none"> ■ The predefined context variable <code>PROTOCOL_HEADER</code> is not available in the drop-down list; to include <code>PROTOCOL_HEADER</code> in the rule, define the variable as Custom Context Variable. ■ If you define a custom context variable in the routing rule, you must write a <i>webMethods IS service</i> and invoke it in the API's Context Based Routing action. In this Integration Server service, use the API to get or set the custom context variable. 								
	Route To	Type the URL of the Native Service endpoint to route the request to, if the above rule criteria are met.								
	Configure Endpoint Properties (icon)	Configure the following set of properties for the specified endpoint individually.								
Configure Endpoint Properties (icon) 	<i>Optional.</i> This icon displays the Endpoint Properties dialog box that enables you to configure a set of properties for the Mediator to route incoming requests to the native API as follows:									
	SOAP Optimization Method	<i>Only for SOAP-Based APIs.</i> Mediator can use the following optimization methods to parse SOAP requests to the native API:								
		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Value</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;">MTOM</td> <td>Mediator uses the Message Transmission Optimization Mechanism (MTOM) to parse SOAP requests to the API.</td> </tr> <tr> <td style="vertical-align: top;">SwA</td> <td>Mediator uses the SOAP with Attachment (SwA) technique to parse SOAP requests to the API.</td> </tr> <tr> <td style="vertical-align: top;">None</td> <td><i>Default.</i> Mediator does not use any optimization method to parse the SOAP requests to the API.</td> </tr> </tbody> </table>	Value	Description	MTOM	Mediator uses the Message Transmission Optimization Mechanism (MTOM) to parse SOAP requests to the API.	SwA	Mediator uses the SOAP with Attachment (SwA) technique to parse SOAP requests to the API.	None	<i>Default.</i> Mediator does not use any optimization method to parse the SOAP requests to the API.
Value	Description									
MTOM	Mediator uses the Message Transmission Optimization Mechanism (MTOM) to parse SOAP requests to the API.									
SwA	Mediator uses the SOAP with Attachment (SwA) technique to parse SOAP requests to the API.									
None	<i>Default.</i> Mediator does not use any optimization method to parse the SOAP requests to the API.									
		<p>Note: Keep the following points in mind:</p> <ul style="list-style-type: none"> ■ Bridging between SwA and MTOM is not supported. If a client sends a SwA request, Mediator can only forward SwA to the native API. The same is true for MTOM, and applies to responses received from the native API. That is, a SwA or MTOM response 								

received by Mediator from a native API is forwarded to the client using the same format it received.

- When sending SOAP requests that do not contain a MTOM or SWA attachment to a native API that returns an MTOM or SWA response, the request 'Accept' header must be set to multipart/related. This is necessary so Mediator knows how to parse the response properly.

HTTP Connection Timeout (Optional). (Number). The time interval (in seconds) after which a connection attempt times out. If a value 0 is specified (or if the value is not specified), Mediator uses the value specified in the `Connection Timeout` field (in the Integration Server Administrator, go to **Settings > Extended**). Default: 30 seconds.

Read Timeout (Optional). (Number). The time interval (in seconds) after which a socket read attempt will timeout.

The precedence of the `Read Timeout` configuration is as follows:

1. If a value is specified for the `Read Timeout` field in the routing endpoint alias, Mediator will use the value specified in the **Runtime Alias > Endpoint Alias > Endpoint Properties > Read Timeout** field. The read timeout value defined at an alias level takes precedence over the timeout values defined at an API level and the global configuration.
2. If a value 0 is specified (or if the value is not specified) for the `Read Timeout` field in the routing endpoint alias, then Mediator will use the value specified in the `Read Timeout` field of this routing action. The read timeout value defined at an API level takes precedence over the global configuration.
3. If a value 0 is specified (or if the value is not specified) for the `Read Timeout` field in this routing action (at an API level), then Mediator will use the value of the global property `pg.endpoint.readTimeout` located in the file `Integration Server_directory\packages\WmMediator\config\resources\pg-config.properties` (in the Mediator Administration console, go to **> Settings > Extended Settings > pg.endpoint.readTimeout** property.).

Note:

If a value for the Read Timeout configuration is not specified in any of the above configuration parameters, then Mediator will use the default 30 seconds.

SSL Configuration

(Optional). To enable SSL client authentication that Mediator uses to authenticate incoming requests for the native API, you must specify values for both the Client Certificate Alias field and the IS Keystore Alias field. If you specify a value for only one of these fields, a deployment error occurs.

Note:

SSL client authentication is optional; you may leave both fields blank.

Prerequisite: You must set up the key alias and keystore properties in the Integration Server. For the procedure, see *webMethods Integration Server Administrator's Guide*.

You uses these properties to specify the following fields:

Value	Description
Client Certificate Alias	The client's private key to be used for performing SSL client authentication.
IS Keystore Alias	The keystore alias of the instance of Integration Server on which Mediator is running. This value (along with the value of Client Certificate Alias) is used for performing SSL client authentication.

WS Security Header

(Only for SOAP-Based APIs). Indicates whether Mediator should pass the WS-Security headers of the incoming requests to the native API.

Value	Description
Remove processed security headers	(Default). Removes the security header if it is processed by Mediator (that is, if Mediator processes the header according to the API's security run-time action). Mediator does not remove the security header if both of the following conditions are true: 1) Mediator did not process the security header, and 2) the mustUnderstand attribute of the security header is 0 or false).

Pass all security headers	Passes the security header, even if it is processed by Mediator (that is, even if Mediator processes the header according to the API's security action).
---------------------------	--

Conditional Error Processing

This action returns a custom error response (and the native provider's service fault content) to the client when the native provider returns a service fault. You can configure conditional error processing and use variables to create custom error messages. This action also allows you to define error messages based on content types: text, XML, and JSON.

Alternatively, you can configure global error responses for all APIs, using Mediator's Service Fault Configuration page (see *Administering webMethods Mediator*).

Note:

To enable support for conditional error processing in CentraSite, add an entry in the **centrasite.xml** customization file as follows:

```
<Action name="ConditionalErrorProcessing" id="uddi:61849291-3bad-4612-819b-5fe5a6b4e958" occurrence="*"></Action>
```

You can find this file on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\custom\conf`.

Input Parameters

Error Conditions Specifies the error conditions and how each of these error conditions are to be processed. You can define a maximum of three conditions. You can configure only one error condition per type. For example, you cannot define two status code conditions. To specify multiple conditions, use the plus button to add rows.

Type	Value	Description
	Status Code	(Default). Specify the error status code. For example: 400, 500.
	HTTP Header	Specify the details of the custom HTTP header(s) included in the client requests. The parameters are: <ul style="list-style-type: none"> ■ HTTP Name <i>String</i>. Specifies the name of the HTTP header. ■ HTTP Value <i>String</i>. Specifies the value of the HTTP header.

XPath Expression	<p>Specifies the details of the XPath expression in the API request. The parameters are:</p> <ul style="list-style-type: none"> ■ XPath Expression ■ Namespaces: Specifies the namespace of the XPath expression. To specify multiple XPath expressions, use the plus button to add rows. <ul style="list-style-type: none"> ■ Prefix: The prefix for the namespace. For example, <code>soapenv</code> or <code>axis</code> ■ URI: The namespace URI - For example, <code>http://schemas.xmlsoap.org/soap/envelope/</code> or <code>http://ws.apache.org/axis</code> ■ Value: Specifies the value of the XPath expression. 						
Pre-Processing	<p>Specifies how the native service error response is to be processed before sending the same to Mediator. You can configure multiple ESB and XSLT processing steps. You can either log the error message sent by the native service without any changes or you can remove any critical information that you do not want Mediator to log or send to the client. The processing of these steps is taken up in the order in which they are configured.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; border-bottom: 1px solid black;">Value</th> <th style="text-align: left; border-bottom: 1px solid black;">Description</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">XSLT</td> <td style="padding: 5px;">(Optional). (String). Specify the XSLT file that you want to use to transform the service error response. Use the select button to browse to and select a file.</td> </tr> <tr> <td style="padding: 5px;">ESB</td> <td style="padding: 5px;">(Optional). (String). (Default). Specify the webMethods IS Service. Invokes one or more webMethods IS services to manipulate the response message from the native API before it is returned to the consuming application. The IS service have access to the response message context (the axis2 MessageContext instance) before it is updated with the custom error message. For example, you might want to send emails or perform custom alerts based on the response payload.</td> </tr> </tbody> </table>	Value	Description	XSLT	(Optional). (String). Specify the XSLT file that you want to use to transform the service error response. Use the select button to browse to and select a file.	ESB	(Optional). (String). (Default). Specify the webMethods IS Service . Invokes one or more webMethods IS services to manipulate the response message from the native API before it is returned to the consuming application. The IS service have access to the response message context (the axis2 MessageContext instance) before it is updated with the custom error message. For example, you might want to send emails or perform custom alerts based on the response payload.
Value	Description						
XSLT	(Optional). (String). Specify the XSLT file that you want to use to transform the service error response. Use the select button to browse to and select a file.						
ESB	(Optional). (String). (Default). Specify the webMethods IS Service . Invokes one or more webMethods IS services to manipulate the response message from the native API before it is returned to the consuming application. The IS service have access to the response message context (the axis2 MessageContext instance) before it is updated with the custom error message. For example, you might want to send emails or perform custom alerts based on the response payload.						
Failure Message	<p><i>String</i>. Specify the custom failure message that Mediator should send to the client. You can configure an error template each for the content types:</p>						

Text, XML, and JSON. For addition details, see the fault handler variables listed in the table below. For example, `$CONSUMER_APPLICATION $OPERATION :$ERROR_MESSAGE`.

Value	Description
Content-Type	<p>Specifies the content type for which the failure message is defined. You can select the following content types:</p> <ul style="list-style-type: none"> ■ text ■ xml ■ json <p>Note: You can define only one error template per content type.</p>
Error Template	Specifies the content of the error template to use. You can use predefined fault handler variables to create the error template. A list of the predefined variables is available in the table below.
Use as Default	Specifies the error template to use as the default.

Custom Error Variables

Specify the error variables to be used in the custom error message. To specify multiple error variables, use the plus button to add rows.

Payload Type	Select the payload type.
Value	Description
Request	Selects the request payload type.
Response	Selects the response payload type.
Name	Specifies the name of the payload.
XPath Expression	Specifies the details of the XPath expression in the API request.
Namespaces	Specifies the namespace of the XPath expression. To specify multiple namespaces, use the plus button to add rows.
Prefix	The prefix for the namespace.
URI	The namespace URI.

Post-Processing You can configure multiple ESB and XSLT processing steps. You can either log the error message sent by the native service without any changes or you can remove any critical information that you do not want Mediator to log or send to the client. The processing of these steps is taken up in the order in which they are configured.

Value	Description
XSLT	(Optional). (String). Specify the XSLT file that you want to use to transform the service error response. Use the select button to browse to and select a file.
ESB	(Optional). (String). (Default). Specify the webMethods IS Service . Invokes one or more webMethods IS services to manipulate the API fault after the Conditional Error Processing action is invoked. The IS service has access to the entire API fault and the Conditional Error Processing message. You can make further changes to the fault message structure, if needed.

Send Native Provider Fault Message When the parameter is enabled, Mediator sends the native SOAP or REST failure message to the client. When you enable this parameter, the `Failure Message` is ignored when a fault is returned by the native API provider. (Faults returned by internal Mediator exceptions are handled by the `Failure Message`.)

Failure Messages

The failure message is returned in both of the following cases:

- When a failure is returned by the native API provider.

In this case, the `$ERROR_MESSAGE` variable in the failure message contains the message produced by the provider's exception that caused the error. This is equivalent to the `getMessage` call on the Java Exception.

- When a failure is returned by internal Mediator exceptions (such as policy violation errors, timeouts, and so on).

In this case, `$ERROR_MESSAGE` contains the error message generated by Mediator.

Alternatively, you can configure global failure messages for *all* APIs, using Mediator's `Service Fault Configuration` page, as described in *Administering webMethods Mediator*.

Mediator returns the following failure message to the consuming application:

```
Mediator encountered an error:$ERROR_MESSAGE while executing
```

```
operation:$OPERATION service:$SERVICE at time:$TIME on date:$DATE.
The client ip was:$CLIENT_IP. The current user:$USER.
The consumer application:$CONSUMER_APPLICATION".
```

The precedence of the failure message configurations is as follows:

- If you configure a Conditional Error Processing action for an API, the failure message configurations take precedence over any settings on the global Service Fault Configuration page.
- If you do not configure a Conditional Error Processing action for an API, the settings on the Service Fault Configuration page take precedence.

The default failure message contains predefined fault handler variables (\$ERROR_MESSAGE, \$OPERATION, and so on.).

You can customize the default failure message using the following substitution variables, where Mediator replaces the variable reference with the real content at run time:

- The predefined context variables listed in [“Context Variables in Virtual Services” on page 1157](#).
- Custom context variables that you declare using Mediator's API .

Note:

If you want to reference a custom context variable that you have already defined in a virtual API, you must add the prefix \$mx to the variable name in order to reference the variable. For example, if you defined the variable TAXID, you would reference it as \$mx:TAXID.

The fault handler variables are described below.

Note:

If no value is defined for a fault handler variable, then the returned value is the literal string null. For example, \$CONSUMER_APPLICATION is always null if the service's policy does not contain at least one of the Evaluate actions.

Fault Handler Variable	Description
\$ERROR_MESSAGE	The error message produced by the exception that is causing the error. This is equivalent to the getMessage call on the Java Exception. This maps to the faultString element for SOAP 1.1 or the Reason element for SOAP 1.2 catch.
\$OPERATION	The operation that was invoked when this error occurred.
\$SERVICE	The service that was invoked when this error occurred.
\$TIME	The time (as determined on the Container side) at which the error occurred.
\$DATE	The date (as determined on the Container side) at which the error occurred.

Fault Handler Variable	Description
\$CLIENT_IP	The IP address of the client invoking the service. This might be available for only certain invoking protocols, such as HTTP(S).
\$USER	The currently authenticated user. The user presents only if the Transport/SOAP Message have user credentials.
\$CONSUMER_APPLICATION	The currently identified consumer application (client).
\$NATIVE_STATUS_CODE	The HTTP error status code that is returned by the native service.

Dynamic Routing

This action enables Mediator to support dynamic routing of virtual aliases based on policy configuration. The policies configured are enforced on the request sent to a service and these requests are forwarded to the dynamic endpoint.

You can define the routing decisions based on one of the following routing options:

- **Header:** You can define the dynamic URL based on the HTTP header value sent by the client. This header name is configured by the API provider and is used to decide the routing decisions at the virtual service. The request message must be routed to the dynamic URL generated from the HTTP header value.
- **Context Variable:** You can define the dynamic URL based on the context variable value. The API providers must provide IS service in the policy action, Invoke webMethods Integration Server. IS service would perform custom manipulations and set the value for the Context Variable ROUTING_ENDPOINT. Mediator takes this ROUTING_ENDPOINT value as the Native endpoint value and performs the Routing.

Input Parameters

The table lists the input parameters to be configured based on the routing options:

Route Using: Header

Value	Description
Header name	<i>String.</i> Mediator uses the HTTP header name that you specify in the Header name field to invoke the virtual service. The name specified in this field is not case-sensitive.
Route Through	<i>URI. Mandatory.</i> Configure the complete or partial dynamic routing endpoint with <code>\${sys:dyn-Endpoint}</code> variable. For example, <code>http://HOSTNAME:5555/rest/cm/softwareag/dynamicURI/validateDynamicURI/\${sys:dyn-Endpoint}</code> .

At run-time the system-defined alias "*sys:dyn-Endpoint*" is replaced with the value of the header name in the incoming request. For example,

```
http://HOSTNAME:5555/rest/com/softwareag/dynamicURI/validateDynamicURI/test.
```

Click the **Configure Endpoint Properties** icon (next to the **Route Through** field) if you want to configure a set of properties for the specified endpoint. Alternatively, Mediator offers local endpoint where the native endpoint is hosted on the same Integration Server as Mediator.

Default Route- To *URI. Mandatory.* Enter the URL of the native API endpoint to route the request to in case the configured header key is not available in the incoming request. For example:

```
http://mycontainer/creditCheckService
```

Click the **Configure Endpoint Properties** icon (next to the **Default Route-To** field) if you want to configure a set of properties for the specified endpoint. Alternatively, Mediator offers local endpoint where the native endpoint is hosted on the same Integration Server as Mediator.

Route Using: Context Variable

Value	Description
Route Through	<p><i>URI. Mandatory.</i> Configure the complete or partial dynamic routing endpoint with <i>sys:dyn-Endpoint</i> variable. For example,</p> <pre>http://HOSTNAME:5555/rest/com/softwareag/dynamicURI/validateDynamicURI/<i>sys:dyn-Endpoint</i>.</pre> <p>At run-time the system-defined alias "<i>sys:dyn-Endpoint</i>" is replaced with the value of the key ROUTING_ENDPOINT in the message context. For example,</p> <pre>http://HOSTNAME:5555/rest/com/softwareag/dynamicURI/validateDynamicURI/test.</pre> <p>Click the Configure Endpoint Properties icon (next to the Route Through field) if you want to configure a set of properties for the specified endpoint. Alternatively, Mediator offers local endpoint where the native endpoint is hosted on the same Integration Server as Mediator.</p>
Default Route- To	<p><i>URI. Mandatory.</i> Enter the URL of the native API endpoint to route the request to, in case the key ROUTING_ENDPOINT is not set in the message context. For example:</p> <pre>http://mycontainer/creditCheckService</pre> <p>Click the Configure Endpoint Properties icon (next to the Default Route-To field) if you want to configure a set of properties for the specified endpoint. Alternatively, Mediator offers local endpoint where the native endpoint is hosted on the same Integration Server as Mediator.</p>

Configure Endpoint Properties  (icon) *Optional.* This icon displays the **Endpoint Properties** dialog box that enables you to configure a set of properties for the Mediator to route incoming requests to the native API as follows:

SOAP Optimization Method This is applicable only for a SOAP API.

Specify the optimization method that Mediator will use to parse the SOAP requests to native API:

Value	Description
MTOM	Mediator will use the Message Transmission Optimization Mechanism (MTOM) to parse SOAP requests to the API.
SwA	Mediator will use the SOAP with Attachment (SwA) technique to parse SOAP requests to the API.
None	<i>Default.</i> Mediator will not use any optimization method to parse the SOAP requests to the API.

Note:

Keep the following points in mind:

- Bridging between SwA and MTOM is not supported. If a client sends a SwA request, Mediator can only forward SwA to the native API. The same is true for MTOM, and applies to responses received from the native API. That is, a SwA or MTOM response received by Mediator from a native API will be forwarded to the client using the same format it received.
- When sending SOAP requests that do not contain a MTOM or SWA attachment to a native API that returns an MTOM or SWA response, the request 'Accept' header must be set to 'multipart/related'. This is necessary so Mediator knows how to parse the response properly.

HTTP Connection Timeout

Number. Optional. The time interval (in seconds) after which a connection attempt will timeout. If a value 0 is specified (or if the value is not specified), Mediator will use the value of the global property `pg.endpoint.connectionTimeout` located in the file `Integration Server_directory/packages\Wmsgiator\config/resources\pg-config.properties` (in the

Mediator Administration console, go to > **Settings > Extended Settings > pg.endpoint.connectionTimeout** property.). Default: 30 seconds.

Read Timeout

Number. Optional. The time interval (in seconds) after which a socket read attempt will timeout.

The precedence of the Read Timeout configuration is as follows:

1. If a value is specified for the Read Timeout field in the routing endpoint alias, Mediator will use the value specified in the **Runtime Alias > Endpoint Alias > Endpoint Properties > Read Timeout** field. The read timeout value defined at an alias level takes precedence over the timeout values defined at an API level and the global configuration.
2. If a value 0 is specified (or if the value is not specified) for the Read Timeout field in the routing endpoint alias, then Mediator will use the value specified in the Read Timeout field of this routing action. The read timeout value defined at an API level takes precedence over the global configuration.
3. If a value 0 is specified (or if the value is not specified) for the Read Timeout field in this routing action (at an API level), then Mediator will use the value of the global property `pg.endpoint.readTimeout` located in the file `Integration Server_directory\packages\Wmsmgator\config\resources\pg-config.properties` (in the Mediator Administration console, go to > **Settings > Extended Settings > pg.endpoint.readTimeout** property.).

Note:

If a value for the Read Timeout configuration is not specified in any of the above configuration parameters, then Mediator will use the default 30 seconds.

SSL Configuration

Optional. To enable SSL client authentication that Mediator will use to authenticate incoming requests for the native API, you must specify values for both the Client Certificate Alias field and the IS Keystore

Alias field. If you specify a value for only one of these fields, a deployment error will occur.

Note:

SSL client authentication is optional; you may leave both fields blank.

Prerequisite: You must set up the key alias and keystore properties in the Integration Server. For the procedure, see *webMethods Integration Server Administrator's Guide*.

You will use these properties to specify the following fields:

Value	Description
Client Certificate Alias	<i>Mandatory.</i> The client's private key to be used for performing SSL client authentication.
IS Keystore Alias	<i>Mandatory.</i> The keystore alias of the instance of Integration Server on which Mediator is running. This value (along with the value of Client Certificate Alias) will be used for performing SSL client authentication.

WS Security Header This is applicable only for a SOAP API.

Indicates whether Mediator should pass the WS-Security headers of the incoming requests to the native API.

Value	Description
Remove processed security headers	<i>Default.</i> Removes the security header if it is processed by Mediator (that is, if Mediator processes the header according to the API's security run-time action). Note that Mediator will not remove the security header if both of the following conditions are true: 1) Mediator did not process the security header, and 2) the <code>mustUnderstand</code> attribute of the security header is 0/false).

Pass all security headers

Passes the security header, even if it is processed by Mediator (that is, even if Mediator processes the header according to the API's security action).

Enable REST Support

This action enables REST support for an existing SOAP based API. You would need to configure the **Enable REST Support** action to:

- Expose a SOAP API both as a SOAP based API and a REST based API using Mediator. Clients who can only send REST requests can now invoke a REST-enabled SOAP API using both a SOAP request and a REST request in Mediator.
- Expose a SOAP API as a REST based API using API Portal. Clients can now invoke a REST-enabled SOAP API using a REST request in API Portal.

Important:

This action is applicable only for a SOAP API.

Note:

This action is set by default in the **Receive** step for all SOAP APIs. To disable the REST support for a SOAP API, delete the Enable REST Support action in the **Receive** step for the API.

Input Parameters

None.

Evaluate Client Certificate for SSL Connectivity

If you have a native API that requires to authenticate a client to the Integration Server using the Secure Sockets Layer (SSL) client authentication, you can use the Evaluate Client Certificate action to extract the client's identity certificate, and verify the client's identity (certificate-based authentication).

This form of authentication does not occur at the message layer using a user ID and password or tokens. This authentication occurs during the connection handshake using SSL certificates.

This action extracts the client identity certificate supplied by the client to the Mediator during the SSL handshake over the Transport layer. For example, when you have configured this action for a proxy API, the PEP extracts the certificate from the Transport layer. In order to identify clients by transport-level certificates, the run-time communication between the client and the Mediator must be over HTTPS and the client must pass a valid certificate.

To use this action, the following prerequisites must be met:

- In Integration Server, create a keystore and truststore, as described in *webMethods Integration Server Administrator's Guide*.

- In Integration Server, create an HTTPS port as described in the *webMethods Integration Server Administrator's Guide*.
- Configure Mediator by setting the HTTPS Ports Configuration parameter, as described in *Administering webMethods Mediator*.

Mediator rejects requests that do not include a client certificate during the SSL handshake over the Transport layer.

If Mediator cannot identify the client, Mediator fails the request and generates a Policy Violation event.

Input Parameters

Identify Consumer (String). The list of consumers against which the client certificate should be validated for identifying requests from a particular client.

<u>Value</u>	<u>Description</u>
Registered Consumers	Mediator tries to verify the client identify certificate against the list of consumer applications who are registered as consumers for the specified API.
Global Consumers	(Default). Mediator tries to verify the client identify certificate against a list of all global consumers available in the Mediator.

Evaluate Hostname

If you have a native API that requires to authenticate a client to the Integration Server using the hostname, you can use the Evaluate Hostname action to extract the client's hostname from the HTTP request header, and verify the client's identity.

This action extracts the specified hostname from an incoming request and locates the client defined by that hostname. For example, when you have configured this action for an API, the PEP extracts the hostname from the request's HTTP header at run time and searches its list of consumers for the client defined by the hostname.

Mediator evaluates the incoming request to identify and validate that the client's request originated from a particular host machine.

Mediator rejects requests that do not include the hostname of an Integration Server user.

If Mediator cannot identify the client, Mediator fails the request and generates a Policy Violation event.

Input Parameters

Identify Consumer (String). The list of consumers against which the hostname should be validated for identifying requests from a particular client.

<u>Value</u>	<u>Description</u>
Registered Consumers	Mediator tries to verify the client's hostname against the list of consumer applications who are registered as consumers for the specified API.
Global Consumers	(Default). Mediator tries to verify the client's hostname against a list of all global consumers available in the Mediator.

Evaluate HTTP Basic Authentication

If you have a native API that requires to authenticate a client to the Integration Server using the HTTP Basic Authentication, you can use the Evaluate HTTP Basic Authentication action to extract the client's credentials (user ID and password) from the Authorization request header, and verify the client's identity.

This action uses HTTP Basic authentication to verify the client's authentication credentials contained in the request's Authorization header. When this action is configured for an API, Mediator validates the credentials against the list of consumers available in the Integration Server on which Mediator is running. If you have selected the checkbox `Authenticate User` using the HTTP Basic Authentication, this type of client authentication is referred to as preemptive authentication.

If the user or password value in the Authorization header cannot be authenticated as a valid Integration Server user (or if the Authorization header is not present in the request), a 500 SOAP fault is returned, and the client is presented with a security challenge. If the client successfully responds to the challenge, the user is authenticated. This type of client authentication is referred to as non-preemptive authentication. If the client does not successfully respond to the challenge, a 401 WWW-Authenticate: Basic response is returned and the invocation is not routed to the policy engine.

If you select to omit the `Authenticate User` parameter (and regardless of whether an Authorization header is present in the request or not), then Mediator forwards the request to the native API, without attempting to authenticate the request.

In the case where a client sends a request with transport credentials (HTTP Basic Authentication) and message credentials (WSS Username Token or WSS X.509 Token), the message credentials take precedence over the transport credentials when Integration Server determines which credentials it should use for the session. For more information, see [and](#) .

If Mediator cannot identify the client, Mediator fails the request and generates a Policy Violation event.

Input Parameters

<code>Identify Consumer</code>	(String). The list of consumers against which authentication credentials (user ID and password) should be validated for identifying requests from a particular client.
--------------------------------	--

Value	Description
Do Not Identify	Mediator forwards the request to the native API, without attempting to verify client's credentials in incoming request.
Global Consumers	(Default). Mediator tries to verify the client's credentials against a list of all global consumers available in the Mediator.
Registered Consumers	Mediator tries to verify the client's credentials against the list of consumer applications who are registered as consumers for the specified API.
Authenticate User	Use this checkbox to specify the users who can access the APIs. If you select the checkbox, Mediator allows only the users specified in the Identify Consumer parameter to access the APIs. If you do not select the checkbox, Mediator allows all users to access the API. In this case, do not configure the Identify Consumer parameter.

Note:

If you have selected the Authenticate User option, the client that connects to the API must have an Integration Server user account.

Evaluate IP Address

If you have a native API that requires to authenticate a client to the Integration Server using the IP address, you can use the Evaluate IP Address action to extract the client's IP address from the HTTP request header, and verify the client's identity.

This action extracts the specified IP address from an incoming request and locates the client defined by that IP address. For example, when you have configured this action for a proxy API, the PEP extracts the IP address from the request's HTTP header at run time and searches its list of consumers for the client defined by the IP address.

Mediator evaluates the incoming request to identify and validate that the client's request originated from a particular IP address.

Mediator rejects requests that do not include the IP address of an Integration Server user.

If Mediator cannot identify the client, Mediator fails the request and generates a Policy Violation event.

Input Parameters

Identify Consumer (String). The list of consumers against which the IP address should be validated for identifying requests from a particular client.

Value	Description
-------	-------------

Registered Consumers	<p>Mediator tries to verify the client's credentials against the list of consumer applications who are registered as consumers for the specified API.</p> <p>Mediator evaluates whether the request header contains the X-Forwarded-For, which is used for identifying the IP address of a client through an HTTP proxy.</p>
Global Consumers	(Default). Mediator tries to verify the client's credentials against a list of all global consumers available in the Mediator.

Evaluate Kerberos Token

Evaluate Kerberos Token policy can be used in any of the following scenarios:

- when the native service does not support Kerberos authentication.
- when you want to centrally configure Kerberos authentication in Mediator for services where Mediator is configured to forward the request to a clustered group of native servers through load balancer.

Note:

For Evaluate Kerberos Token policy, JMS and HTTP are not supported as inbound protocols. Evaluate Kerberos Token policy complies to the **KerberosOverTransport** section described in the following article, <https://msdn.microsoft.com/en-us/library/aa751836.aspx>. Kerberos inbound authentication support is available at message level and at transport level.

Also, ensure that in the Extended Settings page of Integration Server, the the `watt.server.auth.skipForMediator` property is set to `true`.

Input Parameters

Enforcement Point (Only for SOAP-based APIs). You can select the level at which the Kerberos inbound authentication support is available.

<u>Value</u>	<u>Description</u>
Transport Level	To use Kerberos over Transport Level.
Message Level	To use Kerberos over Message Level.

Service Principal Name (String). A valid SPN. The specified value will be used by the client or the server to obtain a service ticket from the KDC server. The SPN is created in the Active Directory (AD) by the AD domain administrator using the following command:

```
Setspn -a <domain name>\<username> spnname
```

For example,

```
setspn -a eur\user1 spnname
```

The Service Principal Name is supported as a user name and a host name based form.

Note:

The Service Principal Name is supported in the username based format. This format represents the principal name as a named user defined in the LDAP or central user directory used for authentication to the KDC.

Service Principal Password	(String). A valid password of the SPN user or the SPN host. For example, if the setspn command is set for the domain user <i>eur\user1</i> , this field represents the password set for the domain user <i>eur\user1</i> .
Identify Consumer	(String). The list of consumers against which the Kerberos token must be validated for identifying requests from a particular client or server.

Value	Description
Do Not Identify	Mediator forwards the request to the native API, without identifying the consumer application(in global/registered consumer list) that corresponds to the principal identified after successful Kerberos authentication.
Global Consumers	(Default). Mediator tries to identify the consumer based on principal that it set after successful Kerberos authentication against the list of global consumer applications in Mediator.
Registered Consumers	Mediator tries to identify the consumer based on principal that it set after successful Kerberos authentication against the list of consumer applications who are registered as consumers for the specified API.

Evaluate OAuth2 Token

If you have a native API that requires to authenticate a client to the Integration Server using the OAuth 2.0 credentials (access token), you can use the Evaluate OAuth2 Authentication action to extract the client's credentials from the HTTP request header, and verify the client's identity.

This action extracts the specified OAuth access token from an incoming request and locates the client defined by that access token. For example, when you have configured this action for an API, the PEP extracts the OAuth access token from the request's HTTP header at run time and searches its list of consumers for the client that is defined by that access token.

Mediator evaluates the incoming request to identify and validate that the client's access token.

Mediator rejects requests that do not include the OAuth access token of an Integration Server user.

Mediator supports OAuth2.0 using the grant type Client Credentials.

If Mediator cannot identify the client, Mediator fails the request and generates a Policy Violation event.

Input Parameters

Identify User (String). The list of consumers against which the OAuth token should be validated for identifying requests from a particular client.

<u>Value</u>	<u>Description</u>
Do Not Identify	Mediator forwards the request to the native API, without attempting to verify client's credentials in incoming request.
Global Consumers	(Default). Mediator tries to verify the client's OAuth access token against a list of all global consumers available in the Mediator.
Registered Consumers	Mediator tries to verify the client's OAuth access token against the list of consumer applications who are registered as consumers for the specified API.

Authenticate Access Token (Boolean). (Optional). This option uses your resource server to verify clients. When Integration Server acts as a resource server, it receives requests from clients that include an access token. The resource server asks the authorization server to validate the access token and user. If the token is valid and the user has privileges to access the folders and services, the resource server executes the request.

For more information about using Integration Server to act as a resource server, see *webMethods Integration Server Administrator's Guide*.

<u>Value</u>	<u>Description</u>
True	(Default). Mediator verifies the client's OAuth access token against the list of consumers available in the Integration Server on which Mediator is running.
False	Mediator does not verify the client's OAuth access token.

Important:

As a best practice, we recommend that if the parameter **Identify User** is set to either use Registered Consumers or Do Not Identify, then the parameter **Authenticate Access Token** should set to True.

Kerberos Authentication (Outbound Scenarios)

Kerberos authentication policy can be used in any of the following scenarios:

Note:

Kerberos authentication support is available at message level and at transport level. Kerberos authentication policy complies to the **KerberosOverTransport** section described in the following article, [https://msdn.microsoft.com/en-us/library/aa751836\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/aa751836(v=vs.110).aspx).

Ensure that the **Evaluate HTTP Basic Authentication** policy is enforced and the `Use Existing Credentials` option is marked.

- When a service provider wants a web service client that does not have the ability to generate the Kerberos token to access a service enforced with the Kerberos policy. It is also used when service provider wants a web service client to access a service enforced with the kerberos policy.

Mediator tries to obtain the Kerberos token from the KDC server on behalf of the authenticated client.

Note:

Before configuring Kerberos, ensure that IS must be configured to LDAP as the incoming client credentials will be authenticated to verify whether its a valid LDAP user. Also, refer to the **Configuring Kerberos in Integration Server** chapter in the *webMethods Integration Server Administrator's Guide* to complete the prerequisites.

- When the service provider wants a web service client to access a service enforced with the Kerberos policy.

Mediator tries to obtain the Kerberos token from the KDC server by using the configured client principal name and password for the virtual service.

Note:

Before configuring Kerberos, refer to the **Configuring Kerberos in Integration Server** chapter in the *webMethods Integration Server Administrator's Guide* to complete the prerequisites.

Kerberos authentication can be performed using one of the following modes available under the **Authenticate Using** drop-down list in the Kerberos Authentication screen. The authentication can be performed using the appropriate modes when the service provider wants a web service client that does not have access to the Kerberos server to access a service enforced with the Kerberos policy:

- **Custom Credentials:** The values provided in the policy is used to obtain the Kerberos token to access the native service.
- **Delegate Incoming Credentials:** The values provided in the policy is used by the API providers to select whether to delegate the incoming kerberos token or act as a normal client.

Note:

To use the **Delegate Incoming Credentials** mode, ensure that in the `krb.conf` file, the `forwardable` parameter is set to `true`.

- **Secure Alias:** The secure alias will be used to obtain the kerberos token to access the native service. For information on configuring secure alias, refer to the **Mediator Runtime Aliases** section in *Working with the CentraSite Business UI Guide*.

- **Use Existing Credentials:** The existing incoming credentials will be used to get the kerberos token from the KDC server to access the native API. Ensure that the **Evaluate HTTP Basic Authentication** policy is enforced and the `Authenticate User` option is selected.

Note:

The Mediator to native service communication must be over SSL.

Input Parameters

Enforcement Point (Only for SOAP-based APIs). You can select the level at which the Kerberos outbound authentication support is available.

Value	Description
Transport Level	To use Kerberos over Transport Level.
Message Level	To use Kerberos over Message Level.

Authenticate Using: Custom Credentials

Value	Description
Client Principal	(String). A valid client LDAP user name.
Client Password	(String). A valid password of the client LDAP user.
Service Principal	(String). A valid Service Principal Name (SPN). The specified value will be used by the client to obtain a service ticket from the KDC server. The SPN is created in the Active Directory (AD) by the AD domain administrator using the following command: <pre>Setspn -a <domain name>\<username> spnname</pre> <p>For example,</p> <pre>setspn -a eur\user1 spnname</pre> <p>Note: Service Principal Name is currently only supported as a user name based form and not a service name based form. The SPN for the native service endpoint.</p>
Service Principal Name Form	The username form, for example, <code>kerberospoc/bob1.SPARTA.RNDLAB.LOC</code>

Authenticate Using: Delegate Incoming Credentials

Value	Description
Client Principal	(String). A valid client LDAP user name.

Client Password (String). A valid password of the client LDAP user.

Service Principal Name Form (String). A valid Service Principal Name (SPN). The specified value will be used by the client to obtain a service ticket from the KDC server. The SPN is created in the Active Directory (AD) by the AD domain administrator using the following command:

```
Setspn -a <domain name>\<username> spnname
```

For example,

```
setspn -a eur\user1 spnname
```

Note:

Service Principal Name is currently only supported as a user name based form and host based form. The SPN for the native service endpoint.

Service Principal Name Form The username form, for example, kerberos poc/bob1.SPARTA.RNDLAB.LOC

Authenticate Using: Secure Alias

Value	Description
Alias Name	(String). Name to the alias configured.

Authenticate Using: Use Existing Credentials

Service Principal Name Form (String). A valid Service Principal Name (SPN). The specified value will be used by the client to obtain a service ticket from the KDC server. The SPN is created in the Active Directory (AD) by the AD domain administrator using the following command:

```
Setspn -a <domain name>\<username> spnname
```

For example,

```
setspn -a eur\user1 spnname
```

Note:

Service Principal Name is currently only supported as a user name based form and not a service name based form. The SPN for the native service endpoint.

Service Principal Name Form The username form, for example, kerberos poc/bob1.SPARTA.RNDLAB.LOC

Evaluate WSS Username Token

If you have a native API that requires to authenticate a client to the Integration Server using the WS-Security authentication, you can use the Evaluate WSS Username Token action to extract the

client's credentials (username token and password) from the WS-Security SOAP message header, and verify the client's identity.

This action extracts the username token and password supplied in the message header of the request and locates the client defined by that username token and password. For example, when you have configured this action for an API, the PEP extracts the username token and password from the SOAP header at run time and searches its list of consumers for the client that is defined by the credentials.

To use this action, the following prerequisites must be met:

- In Integration Server, create a keystore and truststore. For detailed information about securing communications with the server, see the *webMethods Integration Server Administrator's Guide*.
- In Integration Server, create an HTTPS port. For detailed information about configuring ports, see the *webMethods Integration Server Administrator's Guide*.
- Configure Mediator by setting the HTTPS Ports Configuration parameter. For detailed information about configuring Mediator, see *Administering webMethods Mediator*.

Mediator rejects requests that do not include the username token and password of an Integration Server user. Mediator only supports clear text passwords with this kind of authentication.

In the case where a client sends a request with transport credentials (HTTP Basic Authentication) and message credentials (WSS Username Token or WSS X.509 Certificate), the message credentials take precedence over the transport credentials when Integration Server determines which credentials it should use for the session.

If Mediator cannot identify the client, Mediator fails the request and generates a Policy Violation event.

Input Parameters

Identify Consumer (String). The list of consumers against which the username token and password should be validated for identifying requests from a particular client.

Value	Description
Do Not Identify	Mediator forwards the request to the native API, without attempting to verify the client's username token in incoming request.
Global Consumers	(Default). Mediator tries to verify the client's WSS username token against a list of all global consumers available in the Mediator.
Registered Consumers	Mediator tries to verify the client's WSS username token against the list of consumer applications who are registered as consumers for the specified API.

Evaluate WSS X.509 Certificate

If you have a native API that requires to authenticate a client to the Integration Server using the WS-Security authentication, you can use the Evaluate WSS X.509 Certificate action to extract the client identity certificate from the WS-Security SOAP message header, and verify the client's identity.

This action extracts the certificate supplied in the header of an incoming SOAP request and locates the client defined by the information in that certificate. For example, when you have configured this action for an API, the PEP extracts the certificate from the SOAP header at run time and searches its list of consumers for the client that is defined by the certificate.

To use this action, the following prerequisites must be met:

- In Integration Server, create a keystore and truststore, as described in the *webMethods Integration Server Administrator's Guide*.
- In Integration Server, create an HTTPS port, as described in the *webMethods Integration Server Administrator's Guide*.
- Configure Mediator by setting the HTTPS Ports Configuration parameter, as described in *Administering webMethods Mediator*.

Mediator rejects requests that do not include the X.509 token of an Integration Server user.

In the case where a client sends a request with transport credentials (HTTP Basic Authentication) and message credentials (WSS Username Token or WSS X.509 Certificate), the message credentials take precedence over the transport credentials when Integration Server determines which credentials it should use for the session.

If Mediator cannot identify the client, Mediator fails the request and generates a Policy Violation event.

Input Parameters

Identify Consumer (String). The list of consumers against which the X.509 certificate should be validated for identifying requests from a particular client.

Value	Description
Do Not Identify	Mediator forwards the request to the native API, without attempting to verify client's certificate in incoming request.
Global Consumers	(Default). Mediator tries to verify the client's X.509 certificate against a list of all global consumers available in the Mediator.
Registered Consumers	Mediator tries to verify the client's X.509 certificate against the list of consumer applications who are registered as consumers for the specified API.

Evaluate XPath Expression

Note:

This action does not support JSON-based REST APIs.

If you have an API which includes consumer authentication using XPath, you can use the Evaluate XPath Expression action to extract the custom identification credentials from the request. You can then verify the consumer's identity using this information.

The Evaluate XPath Expression action extracts the custom authentication credentials that is supplied in the request which is represented using an XPath expression. The custom authorization can be in the form of tokens, or a username and password token combination. For example, when you configure this action for an API, the PEP extracts the custom identification from the request using an XPath expression at runtime and searches its list of consumers for the XPath defined in the global or registered consumers list.

Mediator rejects requests that do not include the XPath consumer identification defined in the global or registered consumers list.

If Mediator cannot identify the consumer, Mediator fails the request and generates a Policy Violation event.

Input Parameters

Identify Consumer (String). The list of consumers against which the XPath expression should be validated for identifying requests from a particular client.

Value	Description
Do Not Identify	Mediator forwards the request to the native API, without attempting to verify client's XPath expression in incoming request.
Global Consumers	(Default). Mediator tries to verify the client's XPath expression against a list of all global consumers available in the Mediator.
Registered Consumers	Mediator tries to verify the client's XPath expression against the list of consumer applications who are registered as consumers for the specified API.

Namespace (String). (Optional). The namespace of the XPath expression to be validated.

XPath Expression (String). An argument to evaluate the XPath expression contained in the request. See the sample below.

Let's take a look at an example. For the following SOAP message:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```

```

<soap:Header>
</soap:Header>
<soap:Body>
  <catalog xmlns="http://www.store.com">
    <name>My Book</name>
    <author>ABC</author>
    <price>100</price>
  </catalog>
</soap:Body>
</soap:Envelope>

```

The XPath expression is as follows:

```
/soap:Envelope/soap:Body/catalog/author
```

To select the element or token without the namespace, use the following:

```
//*[local-name()='Envelope']/*[local-name()='Body']/*[local-name()='catalog']
/*[local-name()='author'][1]/text()
```

The result: SoftwareAG

HTTP Basic Authentication

This action uses the HTTP authentication mechanism to validate incoming requests from clients. Mediator authorizes the basic credentials (username and password) against a list of all global consumers available in the Mediator.

If the username or password value in the Authorization header cannot be authenticated as a valid Integration Server user (or if the Authorization header is not present in the request), a 500 SOAP fault is returned, and the client is presented with a security challenge. If the client successfully responds to the challenge, the user is authenticated. If the client does not successfully respond to the challenge, a 401 WWW-Authenticate: Basic response is returned and the invocation is not routed to the policy engine. As a result, no events are recorded for that invocation, and its key performance indicator (KPI) data are not included in the performance metrics.

If none of the authentication actions ([“HTTP Basic Authentication” on page 1279](#), [“NTLM Authentication” on page 1298](#) or [“OAuth2 Authentication” on page 1300](#)) is configured for a proxy API, Mediator forwards the request to the native API, without attempting to authenticate the request.

Input Parameters

Authenticate Using (String). The user credentials for authenticating client requests to the native API.

Value	Description
Existing Credentials	(Default). Mediator authenticates requests based on the credentials specified in the HTTP header. It passes the Authorization header present in the original client request to the native API.

Custom Credentials Mediator authenticates requests according to the values you specify in the **User**, **Password** and **Domain** fields.

Field	Description
Username	(String). Account name of a consumer who is available in the Integration Server on which Mediator is running.
Password	(String). A valid password of the consumer.
Domain	(String). (Optional). Domain used by the server to authenticate the consumer.

Invoke webMethods Integration Server

Specifically, you would need to configure the *Invoke webMethods Integration Server* action to:

- Pre-process the request messages into the format required by the native API, before Mediator sends the requests to the native APIs.
- Pre-process the native API's response messages into the format required by the clients, before Mediator returns the responses to the clients.

In some cases an API might need to process messages.

For example, you might need to accommodate differences between the message content that a client is capable of submitting and the message content that a native API expects. For example, if the client submits an order record using a slightly different structure than the structure expected by the native API, you can use this action to process the record submitted by the client to the structure required by the native API.

In the Request Handling sequence, this action invokes the webMethods IS service to pre-process the request received from the client and before it is submitted to the native API.

In the Response Processing sequence, this action invokes the webMethods IS service to process the response message received from the native API and before it is returned to the client.

Note:

A webMethods IS service must be running on the same Integration Server as Mediator. It can call out a C++ or Java or .NET function. It can also call other Integration Server services to manipulate the message.

Input Parameters

webMethods IS Service (String). Enter a name for the webMethods IS Service. This service is used to manipulate the request or response (the axis2 MessageContext instance).

Mediator passes to the invoked IS service the request message context (the axis2 MessageContext instance), which contains the request-specific information. Also, you can use the public IS services that accept MessageContext as input to manipulate the response contents.

JMS Routing Rule

This action allows you to specify a JMS queue to which the Mediator is to submit the request, and the destination to which the native API is to return the response.

To use the JMS Routing Rule action, you publish multiple APIs for a single native API. For example, to make a particular native API available to clients over both HTTP and JMS, you would create two APIs for the native API: one that accepts requests over HTTP and another that accepts requests over JMS. Both APIs would route requests to the same native API on the back end.

Note:

To make it easier to manage APIs, consider adopting a naming convention like the one shown above. Doing so makes it easier to identify APIs and the native API with which they are associated. Keep in mind however, that unlike native APIs, the names of APIs cannot contain spaces or special characters (except `_` and `-`). Consequently, if you adopt a convention that involves using the name of the native API as part of the API name, then the names of the native APIs themselves must not contain characters that are invalid in API names.

To use this action the following prerequisites must be met:

- Create an alias to a JNDI Provider (in the Integration Server Administrator, go to **Settings > Web Services**).
- To establish an active connection between Integration Server and the JMS provider, you must configure Integration Server to use a JMS connection alias (in the Integration Server Administrator, go to **Settings > Messaging > JMS Settings**).
- Create a WS (Web Service) endpoint alias for provider Web Service Descriptor (WSD) that uses a JMS binder. In the Integration Server Administrator, navigate to **Settings > Web Services** and complete the Alias Name, Description, Descriptor Type, and Transport Type fields.
- Configure a WS (Web Service) endpoint trigger (in the Integration Server Administrator, go to **Settings > Messaging > JMS Trigger Management**).
- Create a WS (Web Service) endpoint alias for consumer Web Service Descriptor (WSD) that has a JMS binder. In the Integration Server Administrator, navigate to **Settings > Web Services** and complete the Alias Name, Description, Descriptor Type, and Transport Type fields.

- Additionally, in the proxy API's **Message Flow** area, make sure that you delete the predefined *Straight Through Routing* and *HTTP Basic Authentication* actions from the **Receive** stage. This is because, these actions are mutually exclusive.

For detailed information and procedures, see the *webMethods Integration Server Administrator's Guide*.

Input Parameters

Connection URL (String). Specify a connection alias for connecting to the JMS provider (for example, an Integration Server alias or a JNDI URL). For example, a JNDI URL of the form: `jms:queue:dynamicQueues/MyRequestQueue?wm-wsendpointalias=MediatorConsumer &targetService=vs-jms-in-echo`

Note that the `wm-wsendpointalias` parameter is required for Integration Server/Mediator to look up the JMS consumer alias to send the request to the specified queue (for example, `MyRequestQueue`), which is a dynamic queue in ActiveMQ. Also, the `targetService` parameter is required if sending to another API that uses JMS as the entry protocol.

Reply to Destination (Optional). Specify a queue name where a reply to a message should be sent.

Priority Type an integer that represents the priority of this JMS message with respect to other messages that are in the same queue. The priority value determines the order in which the messages are routed. The lower the **Priority** value, the higher the priority (that is, 0 is the highest priority, and messages with this priority value are executed first).

- Priority values 0 through 10.
- The default priority for a JMS message is 0.

Time to Live (Optional). A numeric value (in milliseconds) that specifies the expiration time of the JMS message. If the time-to-live is specified as zero, expiration is set to zero which indicates the message does not expire.

The default value is 0.

Delivery Mode (Optional). The type of message delivery to the endpoint.

Value	Description
Persistent	The message is stored by the JMS server before delivering it to the client.
Non-Persistent	<i>Default.</i> The message is not stored before delivery.

Multiple JMS Messages in a Queue

To determine the order in which to execute the JMS messages in a queue, Mediator examines each message's Priority setting.

The Priority setting contains a non-negative integer that indicates the JMS message's priority. A priority value of 0 represents the highest possible priority.

Note:

A JMS message's **Priority** property is used *only* when there are multiple JMS messages to route in the queue. If the queue has only one message to route, the **Priority** property is ignored entirely.

When a queue includes multiple JMS messages, Mediator routes the messages serially, in priority order from lowest to highest (that is, it routes with message the *lowest* priority value first). Each messages in the queue is routed to completion before the next one begins.

If two or more messages have the same priority value, their order is indeterminate. Mediator routes these messages in serial fashion after all lower priority messages and before any higher priority messages. However, you cannot predict their order

Example

If Mediator were given the following JMS messages to route for an API:

JMS Message	Priority
JMS Message A	11
JMS Message B	25
JMS Message C	11
JMS Message D	0

It would route the messages in the following order:

JMS Message	Priority
JMS Message D	0
JMS Message A then JMS Message C (or vice versa) The order of these two messages cannot be controlled or predicted because they have the same priority.	11
JMS Message B	25

Load Balancing and Failover Routing

If you have a native API that is hosted at two or more endpoints, you can use the Load Balancing and Failover Routing to distribute requests among the endpoints.

Requests are distributed across multiple endpoints. The requests are intelligently routed based on the round-robin execution strategy. The load for a service is balanced by directing requests to two or more services in a pool, until the optimum level is achieved. The application routes requests to services in the pool sequentially, starting from the first to the last service without considering the individual performance of the services. After the requests have been forwarded to all the services in the pool, the first service is selected for the next loop of forwarding.

Load-balanced endpoints also have automatic failover capability. If a load-balanced endpoint is unavailable (for example, if a connection is refused), then that endpoint is marked as down for the number of seconds you specify in the `Timeout` field (during which the endpoint is not used for sending the request), and the next configured endpoint is tried. If all the configured load-balanced endpoints are down, then a failure is sent back to the client. After the timeout expires, each endpoint marked is available again to send the request.

Input Parameters

Route To (URI). Type the URLs of two or more endpoints in a pool to which the requests are routed. The application routes the requests to endpoints in the pool sequentially, starting from the first to the last endpoint without considering the individual performance of the endpoints. After the requests have been forwarded to all the endpoints in the pool, the first endpoint is selected for the next loop of forwarding.

Enter the URL of the endpoint to route the request to. For example:

```
http://mycontainer/creditCheckService
```

To specify additional endpoints, use the plus button next to the field to add rows.

Click the **Configure Endpoint Properties** icon (next to the field) if you want to configure a set of properties for the endpoints individually.

Alternatively, Mediator offers Local Optimization capability if the native API is hosted on the same Integration Server as Mediator. With local optimization, API invocation happens in-memory and not through a network hop.

```
local://<Service-full-path>
```

OR

```
local://<server>:<port>/ws/<Service-full-path>
```

For example:

```
local://MyAPIFolder:MyLocalAPI
```

which points to the endpoint API `MyLocalAPI` which is present under the folder `MyAPIFolder` in Integration Server.

Note:

Local Optimization is not applicable to REST based APIs.

Configure
Endpoint
Properties 
(icon)

(Optional). This icon displays the **Endpoint Properties** dialog box that enables you to configure a set of properties for the Mediator to route incoming requests to the native API as follows:

SOAP
Optimization
Method (Only for SOAP-based APIs). Mediator can use the following optimization methods to parse SOAP requests to the native API:

Value	Description
MTOM	Mediator uses the Message Transmission Optimization Mechanism (MTOM) to parse SOAP requests to the API.
SwA	Mediator uses the SOAP with Attachment (SwA) technique to parse SOAP requests to the API.
None	(Default). Mediator does not use any optimization method to parse the SOAP requests to the API.

Note:

Keep the following points in mind:

- Bridging between SwA and MTOM is not supported. If a client sends a SwA request, Mediator can only forward SwA to the native API. The same is true for MTOM, and applies to responses received from the native API. That is, a SwA or MTOM response received by Mediator from a native API is forwarded to the client using the same format it received.
- When sending SOAP requests that do not contain a MTOM or SWA attachment to a native API that returns an MTOM or SWA response, the request 'Accept' header must be set to multipart/related. This is necessary so Mediator knows how to parse the response properly.

HTTP Connection
Timeout (Number). (Optional). The time interval (in seconds) after which a connection attempt timeouts. If a value 0 is specified (or if the value is not specified), Mediator uses the value specified in the Connection Timeout field (in the Integration Server Administrator, go to **Settings > Extended**). Default: 30 seconds.

Read Timeout (Number). (Optional). The time interval (in seconds) after which a socket read attempt timeouts. If a value 0 is specified (or if the value is not specified), Mediator uses the value specified in the `Read Timeout` field (in the Integration Server Administrator, go to **> Settings > Extended.**). Default: 30 seconds.

SSL Configuration (Optional). To enable SSL client authentication that Mediator uses to authenticate incoming requests for the native API, you must specify values for both the `Client Certificate Alias` field and the `IS Keystore Alias` field. If you specify a value for only one of these fields, a deployment error occurs.

Note:

SSL client authentication is optional; you may leave both fields blank.

Prerequisite: You must set up the key alias and keystore properties in the Integration Server. For the procedure, see *webMethods Integration Server Administrator's Guide*.

You uses these properties to specify the following fields:

Value	Description
<code>Client Certificate Alias</code>	The client's private key to be used for performing SSL client authentication.
<code>IS Keystore Alias</code>	The keystore alias of the instance of Integration Server on which Mediator is running. This value (along with the value of <code>Client Certificate Alias</code>) is used for performing SSL client authentication.

WS Security Header (Only for SOAP-based APIs). Indicates whether Mediator should pass the WS-Security headers of the incoming requests to the native API.

Value	Description
<code>Remove processed security headers</code>	(Default). Removes the security header if it is processed by Mediator (that is, if Mediator processes the header according to the API's security run-time policy). Mediator does remove the security header if both of

the following conditions are true: 1) Mediator did not process the security header, and 2) the `mustUnderstand` attribute of the security header is 0 or false).

Pass all security headers	Passes the security header, even if it is processed by Mediator (that is, even if Mediator processes the header according to the API's security action).
---------------------------	--

Log Invocation

This action logs request or response payloads. You can specify the log destination and the logging frequency. This action also logs other information about the requests or responses, such as the API name, operation name, the Integration Server user, a timestamp, and the response time.

Input Parameters

Payloads (String). Specify whether to log all request or response payloads.

Value	Description
Request	Logs all request payloads.
Response	Logs all response payloads.

Log Generation Frequency (String). Specify how frequently to log the payload.

Value	Description
Always	Logs all requests and responses.
On Success	Logs only the successful responses and requests.
On Failure	Logs only the failed requests and responses.

Send Data To (String). Specify where to log the payload.

Important:

Ensure that Mediator is configured to log the payloads to the destination(s) you specify here. For details, see *Administering webMethods Mediator*.

Value	Description
API Portal	Mediator can use API Portal destination to publish data about run-time events and key performance indicator (KPI) metrics.

Mark the checkbox **API Portal** and select the API Portal instance from the drop-down list. When you mark the checkbox, CentraSite displays the list of API Portal instances that are registered and are available to you. To specify multiple API Portal instances, use the plus button to add rows.

Note:

When you publish an API that is configured with an API Portal destination to Mediator, CentraSite automatically creates an entry of the configured API Portal destination in the Integration Server Administrator's **Solutions > Mediator > Administration > API Portal** page.

Mediator uses the API Portal destination to publish the following event types:

- Transaction Events
- Monitoring Events
- Lifecycle Events
- Policy Violation Events
- Error Events
- Performance Metrics

You can also use the Mediator capability to specify the intervals at which the events and KPI metrics must be published to the API Portal destination.

Prerequisite: You must configure CentraSite to communicate with API Portal using the **Add Gateway** action.

CentraSite

Logs the payloads in the API's Events profile in CentraSite.

Prerequisite: You must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > CentraSite Communication**). For the procedure, see *Administering webMethods Mediator*.

Elasticsearch

Mediator can use Elasticsearch destination to publish data about run-time events and key performance indicator (KPI) metrics.

Mediator uses the Elasticsearch destination to publish the following event types:

- Transaction Events
- Monitoring Events
- Lifecycle Events
- Policy Violation Events
- Error Events
- Performance Metrics

You can also use the Mediator capability to specify the intervals at which the events and KPI metrics must be published to the Elasticsearch destination.

Prerequisite: You must configure the Elasticsearch destination (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > Elasticsearch**). For the procedure, see *Administering webMethods Mediator*.

Local Log

Logs the payloads in the server log of the Integration Server on which Mediator is running.

Also select a value in the `Log Level` field:

- Info: logs the error-level, warning-level, and informational-level alerts.
- Warn: logs the error-level and warning-level alerts.
- Error: logs only error-level alerts.

Important:

The Integration Server Administrator's logging level for Mediator should match the logging level specified for this action (go to **Settings > Logging > Server Logger**).

SNMP

Logs the payloads in CentraSite's SNMP server or a third-party SNMP server.

Prerequisite: You must configure the SNMP server destination (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > SNMP**).

Email

Sends the payloads to an SMTP email server, which sends them to the email address(es) you specify here.

Mediator sends the payloads as email attachments that are compressed using gzip data compression. To specify multiple addresses, use the plus button to add rows.

Prerequisite: You must configure the SMTP server destination (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > Email**). For the procedure, see *Administering webMethods Mediator*.

Audit Log

Logs the payload to the Integration Server audit logger. For more information about logging, see the *webMethods Audit Logging Guide*.

Note:

If you expect a high volume of events in your system, it is recommended that you select the Audit Log destination for this action.

EDA/Database

Logs the payloads in an EDA endpoint/Database destination that you configured in Integration Server Administrator:

- An EDA endpoint (that is, a default endpoint configured in the universal messaging configuration).
- A Database (that is, a JDBC connection pool is defined in Integration Server and associated with the Mediator functional alias).

Prerequisite: You must configure the EDA/Database destination in Integration Server on the **Solutions > Mediator > Administration > EDA/Database Configuration** page. For details, see *Administering webMethods Mediator*.

Monitor Service Level Agreement

This action monitors a set of run-time performance conditions for an API, and sends alerts to a specified destination when the performance conditions are violated. This action enables you to monitor run-time performance for *one or more specified clients*.

You can configure this action to define a *Service Level Agreement (SLA)*, which is a set of conditions that defines the level of performance that a client should expect from an API. You can use this action to identify whether an API threshold rules are met or exceeded. For example, you might define an agreement with a particular client that sends an alert to the client (consumer application) if responses are not sent within a certain maximum response time. You can configure SLAs for each API or consumer application combination.

For the counter-based metrics (Total Request Count, Success Count, Fault Count), Mediator sends an alert as soon as the performance condition is violated, without having to wait until the end of the metrics tracking interval. You can select whether to send an alert only once during the interval, or every time the violation occurs during the interval. (Mediator sends another alert the next time a condition is violated during a subsequent interval.)

For the aggregated metrics (Average Response Time, Minimum Response Time, Maximum Response Time), Mediator aggregates the response times at the end of the interval, and then sends an alert if the performance condition is violated.

This action does not include metrics for failed invocations.

Note:

To enable Mediator to publish performance metrics, you must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > CentraSite Communication**). For the procedure, see *Administering webMethods Mediator*.

Input Parameters

Action Configuration parameters

Specify one or more conditions to monitor. To do this, specify a metric, operator, and value for each metric. To specify multiple conditions, use the plus button to add multiple rows. If multiple parameters are used, they are connected by the AND operator.

Name

String Array. The metrics to monitor.

<u>Value</u>	<u>Description</u>
Availability	Indicates whether the service was available to the specified consumers in the current interval.
Average Response Time	The average amount of time it took the service to complete all invocations in the current interval. Response time is measured from the moment Mediator receives the request until the moment it returns the response to the caller. You can specify a value to monitor if the response time is within the set limit or take required steps. For example, if you specify 2 seconds, a monitoring alert will be generated if the response time exceeds 2 seconds during the current interval.
Fault Count	The number of faults returned in the current interval. You can specify a

	number exceeding which an alert will be generated indicate that necessary actions to taken to control the fault count. For example, if you specify 5 in this field, a monitoring alert will be generated if the fault count exceeds 5 during the current interval.				
Maximum Response Time	The maximum amount of time to respond to a request in the current interval. You can specify a value to monitor if the response time does not exceed the time provided. A monitoring alert will be generated if the maximum time taken to respond is exceeded.				
Minimum Response Time	The minimum amount of time to respond to a request in the current interval.				
Successful Request Count	The number of successful requests in the current interval.				
Total Request Count	The total number of requests (successful and unsuccessful) in the current interval.				
Operator	<i>String Array.</i> Select an appropriate operator.				
Value	<i>String Array</i> Specify an appropriate value.				
Alert for Consumer Applications	<i>Object Array.</i> Specify the Application asset(s) to which this Service Level Agreement applies. To specify multiple Application assets, use the plus button to add multiple rows.				
Alert parameters	<i>Object.</i> Specify the following parameters for the alerts that reports on the Service Level Agreement conditions:				
Alert Interval	<i>Number.</i> The time period (in minutes) in which to monitor performance before sending an alert if a condition is violated.				
Alert Frequency	<i>String.</i> Specifies how frequently to issue alerts for the counter-based metrics (Total Request Count, Success Count, Fault Count).				
	<table border="0"> <thead> <tr> <th><u>Value</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>Every Time</td> <td>Issue an alert every time one of the specified conditions is violated.</td> </tr> </tbody> </table>	<u>Value</u>	<u>Description</u>	Every Time	Issue an alert every time one of the specified conditions is violated.
<u>Value</u>	<u>Description</u>				
Every Time	Issue an alert every time one of the specified conditions is violated.				

	Only Once	Issue an alert only the first time one of the specified conditions is violated.
Rule Expiration Date		<i>String</i> . Specifies the date on which this Service Monitoring Performance action expires, in format MM/DD/YYYY.
Reply to Destination		<i>String</i> . Specifies where to log the alert.

Important:

Ensure that Mediator is configured to send event notifications to the destination(s) you specify here. For details, see *Administering webMethods Mediator*.

Value	Description
CentraSite	<p>Sends the alerts to the virtual service's Events profile in CentraSite.</p> <p>Prerequisite: You must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to Solutions > Mediator > Administration > CentraSite Communication). For the procedure, see <i>Administering webMethods Mediator</i>.</p>
Local Log	<p>Sends the alerts to the server log of the Integration Server on which Mediator is running.</p> <p>Also select a value in the Log Level field:</p> <ul style="list-style-type: none"> ■ Info: Logs error-level, warning-level, and informational-level alerts. ■ Warn: Logs error-level and warning-level alerts. ■ Error: Logs only error-level alerts. <p>Important: The Integration Server Administrator's logging level for Mediator should match the logging level specified for this action (go to Settings > Logging > Server Logger).</p>

SNMP Sends the alerts to CentraSite's SNMP server or a third-party SNMP server.

Prerequisite: You must configure the SNMP server destination (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > Email**). For the procedure, see *Administering webMethods Mediator*.

Email Sends the alerts to an SMTP email server, which sends them to the email address(es) you specify here. To specify multiple addresses, use the plus button to add rows.

Prerequisite: You must configure the SMTP server destination (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > Email**). For the procedure, see *Administering webMethods Mediator*.

EDA/Database Sends the alerts to an EDA endpoint/Database destination that you configured in Integration Server Administrator:

- An EDA endpoint (that is, a default endpoint configured in the universal messaging configuration).
- A Database (that is, a JDBC connection pool is defined in Integration Server and associated with the Mediator functional alias).

Prerequisite: You must configure the EDA/Database destination in Integration Server on the **Solutions > Mediator > Administration > EDA/Database Configuration** page. For details, see *Administering webMethods Mediator*.

Alert Message *String. Optional.* Specify a text message to include in the alert.

Monitor Service Performance

This action is similar to the [Monitor Service Level Agreement](#) action. Both actions can monitor the same set of run-time performance conditions for an API, and then send alerts when the performance conditions are violated. However, this action monitors run-time performance for *a specific client*.

For the counter-based metrics (Total Request Count, Success Count, Fault Count), Mediator sends an alert as soon as the performance condition is violated, without having to wait until the end of the metrics tracking interval. You can select whether to send an alert only once during the interval, or every time the violation occurs during the interval. (Mediator sends another alert the next time a condition is violated during a subsequent interval.)

For the aggregated metrics (Average Response Time, Minimum Response Time, Maximum Response Time), Mediator aggregates the response times at the end of the interval, and then sends an alert if the performance condition is violated.

This action does not include metrics for failed invocations.

Note:

To enable Mediator to publish performance metrics, you must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > CentraSite Communication**). For detailed information about configuring communication with CentraSite, see *Administering webMethods Mediator*.

Input Parameters

Action Configuration parameters Specify one or more conditions to monitor. To do this, specify a metric, operator, and a value for each metric. To specify multiple conditions, use the plus button to add multiple rows. If multiple parameters are used, they are connected by the AND operator.

Name *String Array.* The metrics to monitor.

Value	Description
Availability	Indicates whether the service was available to the specified consumers in the current interval.
Average Response Time	The average amount of time it took the service to complete all invocations in the current interval. Response time is measured from the moment Mediator receives the request until the moment it returns the response to the caller. You can specify a value to monitor if the response time is within the set limit or take required steps. For example, if you specify 2 seconds, a

monitoring alert will be generated if the response time exceeds 2 seconds during the current interval.

Fault Count The number of faults returned in the current interval. You can specify a number exceeding which an alert will be generated indicate that necessary actions to taken to control the fault count. For example, if you specify 5 in this field, a monitoring alert will be generated if the fault count exceeds 5 during the current interval.

Maximum Response Time The maximum amount of time to respond to a request in the current interval. You can specify a value to monitor if the response time does not exceed the time provided. A monitoring alert will be generated if the maximum time taken to respond is exceeded.

Minimum Response Time The minimum amount of time to respond to a request in the current interval.

Successful Request Count The number of successful requests in the current interval.

Total Request Count The total number of requests (successful and unsuccessful) in the current interval.

Operator *String Array*. Select an appropriate operator.

Value *String Array*. Specify an appropriate value.

Alert parameters *Object*. Specify the following parameters for the alerts that reports on the conditions:

Alert Interval *Number*. The time period (in minutes) in which to monitor performance before sending an alert if a condition is violated.

Alert Frequency *String*. Specifies how frequently to issue alerts for the counter-based metrics (Total Request Count, Success Count, Fault Count).

Value	Description
Every Time	Issue an alert every time one of the specified conditions is violated.
Only Once	Issue an alert only the first time one of the specified conditions is violated.

Reply to Destination *String*. Specifies where to send the alerts.

Important:

Ensure that Mediator is configured to send event notifications to the destination(s) you specify here. For details, see *Administering webMethods Mediator*

Value	Description
CentraSite	<p>Sends the alerts to the virtual service's Events profile in CentraSite.</p> <p>Prerequisite: You must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to Solutions > Mediator > Administration > CentraSite Communication). For the procedure, see <i>Administering webMethods Mediator</i>.</p>
Local Log	<p>Sends the alerts to the server log of the Integration Server on which Mediator is running.</p> <p>Also select a value in the Log Level field:</p> <ul style="list-style-type: none"> ■ Info: Logs error-level, warning-level, and informational-level alerts. ■ Warn: Logs error-level and warning-level alerts. ■ Error: Logs only error-level alerts. <p>Important: The Integration Server Administrator's logging level for Mediator should match the logging level specified for this action (go to Settings > Logging > Server Logger).</p>
SNMP	<p>Sends the alerts to CentraSite's SNMP server or a third-party SNMP server.</p> <p>Prerequisite: You must configure the SNMP server destination (in the Integration Server Administrator, go to Solutions > Mediator > Administration > Email). For the procedure, see <i>Administering webMethods Mediator</i>.</p>
Email	<p>Sends the alerts to an SMTP email server, which sends them to the email address(es) you specify here. To specify multiple addresses, use the plus button to add rows.</p> <p>Prerequisite: You must configure the SMTP server destination (in the Integration Server</p>

Administrator, go to **Solutions > Mediator > Administration > Email**). For the procedure, see *Administering webMethods Mediator*.

EDA/Database

Sends the alerts to an EDA endpoint/Database destination that you configured in Integration Server Administrator:

- An EDA endpoint (that is, a default endpoint configured in the universal messaging configuration).
- A Database (that is, a JDBC connection pool is defined in Integration Server and associated with the Mediator functional alias).

Prerequisite: You must configure the EDA/Database destination in Integration Server on the **Solutions > Mediator > Administration > EDA/Database Configuration** page. For details, see *Administering webMethods Mediator*.

Alert Message

String. Optional. Specify a text message to include in the alert.

NTLM Authentication

This action uses the NTLM authentication to validate incoming requests from clients. Mediator authorizes the NTLM credentials (username and password) against a list of all global consumers available in the Mediator.

If the username or password value in the Authorization header cannot be authenticated as a valid Integration Server user (or if the Authorization header is not present in the request), a 500 SOAP fault is returned, and the client is presented with a security challenge. If the client successfully responds to the challenge, the user is authenticated. If the client does not successfully respond to the challenge, a 401 Unauthorized WWW-Authenticate: NTLM (for NTLM authentication) or WWW-Authenticate: Negotiate (for Kerberos authentication) is returned in the response header and the invocation is not routed to the policy engine. As a result, no events are recorded for that invocation, and its key performance indicator (KPI) data are not included in the performance metrics.

Note:

If Mediator is used to access a native API protected by NTLM (which is typically hosted in IIS), then the native API in IIS should be configured to use NTLM as the authentication scheme. If the authentication scheme is configured as Windows, then NTLM must be in its list.

If none of the authentication actions (HTTP Basic Authentication, NTLM Authentication, or OAuth2 Authentication) is configured for a proxy API, Mediator forwards the request to the native API, without attempting to authenticate the request.

Input Parameters

Authenticate Using (String). Specifies the user credentials for authenticating client requests to the native API.

Note:

If Mediator is used to access a native API protected by NTLM (which is typically hosted in IIS), then the native API in IIS should be configured to use NTLM as the authentication scheme. If the authentication scheme is configured as Windows, then NTLM should be in its list. The Negotiate handshake is supported in the near future. This note applies to all three of the following options for NTLM.

Value	Description
Existing Credentials	(Default). Mediator uses the user credentials passed in the request header for an NTLM handshake with the server.
Custom Credentials	Mediator uses the values you specify in the User , Password and Domain fields for an NTLM handshake with the server.

Field	Description
Username	(String). Account name of a consumer who is available in the Integration Server on which Mediator is running.
Password	(String). A valid password of the consumer.
Domain	(String). (Optional). Domain used by the server to authenticate the consumer.

Transparent Mediator supports Kerberos handshake in Transparent mode. The following additional settings are required for Kerberos:

- Configure the client with `clientCredentialType` set to Windows.
- Set the value of `watt.pg.disableNtlmAuthHandler` property to `true` in the extended settings for the Integration Server
- Set the property `handleClientErrorCode` to `true` in `pg-core.xml` as follows:

```
<bean
  id="httpResponseCodeCallback"
  class="com.softwareag.pg.axis2.transpo
```

```

rts.ISHTTPResponseCodeCallback">
<property name="handleClientErrorCode"
          value="true"/>
</bean>

```

For more information about configuring the extended settings, see *webMethods Integration Server Administrator's Guide*.

OAuth2 Authentication

This action uses the OAuth 2.0 authentication to validate incoming requests from clients. Mediator authorizes the OAuth 2.0 credentials (access token) against a list of all global consumers available in the Mediator.

If the access token value in the Authorization header cannot be authenticated as a valid Integration Server user (or if the Authorization header is not present in the request), a 500 SOAP fault is returned, and the client is presented with a security challenge. If the client successfully responds to the challenge, the user is authenticated. If the client does not successfully respond to the challenge, a WWW-Authenticate: OAuth response is returned and the invocation is not routed to the policy engine. As a result, no events are recorded for that invocation, and its key performance indicator (KPI) data are not included in the performance metrics.

If none of the authentication actions ([HTTP Basic Authentication](#), [NTLM Authentication](#) or [OAuth2 Authentication](#)) is configured for a proxy API, Mediator forwards the request to the native API, without attempting to authenticate the request.

Input Parameters

Authenticate Using (String). Specifies the OAuth2 access token for authenticating client requests to the native API.

Value	Description
Existing Token	(Default). Mediator uses the OAuth2 access token specified in the HTTP Authorization header to validate client requests for a native API.
Custom Token	Mediator uses the access token you specify in the OAuth2 Token , field to validate client requests for a native API.

Field	Description
OAuth2 Token	(String). Specifies an OAuth2 access token to be deployed by Mediator. The consumer need not pass the OAuth2 token during service invocation.

Response Transformation

This action specifies:

- The XSLT transformation file to transform response messages from native APIs into a format required by the client.

In some cases a message needs to be transformed prior to sending to the client.

For example, you may have to accommodate differences between the message content that a native API is capable of submitting and the message content that a client expects. For example, if the native API submits an order record using a slightly different structure than the structure expected by the client, you can use this action to transform the record submitted by the native API to the structure required by the client.

When this action is configured for a proxy API, the native API's response messages are transformed into the format required by the client, before Mediator returns the responses to the clients.

Input Parameters

Transformation File (File). Click **Choose**, select the XSL transformation file from your file system and click **OK**.

When you virtualize an API, the transformation file is validated. If there are no validation errors, the XSLT file is displayed as a download link in the same dialog. If the transformation file is invalid (for example, non-XSLT file), this is indicated by a warning icon.

Note:

If you make changes to the XSLT file in the future, you must republish the API.

Important:

The XSL file uploaded by the user should not contain the XML declaration in it (for example, `xml version=1.0 encoding="UTF-8"`). This is because when the API is published to Mediator, Mediator embeds the XSL file in the virtual service definition (VSD), and since the VSD itself is in XML format, there cannot be an XML declaration line in the middle of it. This can lead to unexpected deployment issues which can be avoided by making sure the XSL file does not contain the declaration line.

Request Transformation

This action specifies:

- The XSLT Transformation File to transform request messages from clients into a format required by the native API.

In some cases a native API might need to transform messages.

For example, you may have to accommodate differences between the message content that a client is capable of submitting and the message content that a native API expects. For example, if the client submits an order record using a slightly different structure than the structure expected by the native API, you can use this action to transform the record submitted by the client to the structure required by the native API.

When this action is configured for a proxy API, the incoming requests from the clients are transformed into a format required by the native API, before Mediator sends the requests to the native APIs.

Input Parameters

Transformation File (File). Click **Choose**, select the XSL transformation file from your file system and click **OK**.

When you virtualize an API, the transformation file is validated. If there are no validation errors, the XSLT file is displayed as a download link in the same dialog. If the transformation file is invalid (for example, non-XSLT file), this is indicated by a warning icon.

Note:

If you make changes to the XSLT file in the future, you must republish the API.

Important:

The XSL file uploaded by the user should not contain the XML declaration in it (for example, `xml version=1.0 encoding=UTF-8`). This is because when the API is published to Mediator, Mediator embeds the XSL file in the virtual service definition (VSD), and since the VSD itself is in XML format, there cannot be an XML declaration line in the middle of it. This can lead to unexpected deployment issues which can be avoided by making sure the XSL file does not contain the declaration line.

Require Encryption

This action requires that a request's XML element, which is represented by an XPath expression or parts of soap request such as soap body or soap headers be encrypted.

To use this action, the following prerequisites must be met:

1. **Configure Integration Server:** Set up keystores and truststores in Integration Server, as described in the *webMethods Integration Server Administrator's Guide*.
2. **Configure Mediator:** In the Integration Server Administrator, navigate to **Solutions > Mediator > Administration > General** and complete the IS Keystore Name, IS Truststore Name, and Alias (signing) fields, as described in the *Administering webMethods Mediator* guide.

When this action is configured for a proxy API, Mediator provides decryption of incoming requests and encryption of outgoing responses. Mediator can encrypt and decrypt only individual elements in the SOAP message body that are defined by the XPath expressions configured for the action. Mediator requires that requests contain the encrypted elements that match those in the XPath expression. You must encrypt the entire element, not just the data between the element tags. Mediator rejects requests if the element name is not encrypted.

Important:

Do not encrypt the entire SOAP body because a SOAP request without an element appears to Mediator as malformed.

Mediator attempts to encrypt the response elements that match the XPath expressions with those defined for the action. If the response does not have any elements that match the XPath expression, Mediator does not encrypt the response before sending. If the XPath expression resolves a portion of the response message, but Mediator cannot locate a certificate to encrypt the response, then Mediator sends a SOAP fault exception to the client and a Policy Violation event notification to CentraSite.

How Mediator Encrypts Responses

The Require Encryption action encrypts the response back to the client by dynamically setting a public key alias at run time. Mediator uses any one of the following approaches to determine the public key alias:

- If Mediator can access the X.509 certificate of the client based on the incoming request signature, it uses useReqSigCert as the public key alias.

OR

- If an Evaluate action is present in the message flow and it successfully identifies a client, then Mediator looks for a public key alias with that client name in the IS Keystore Name property. The IS Keystore Name property is specified in the Integration Server Administrator, under **Solutions > Mediator > Administration > General**. This property should be set to an Integration Server keystore that Mediator can use.

For an Evaluate action that allows for anonymous usage, Mediator does *not* require a client name in order to send encrypted responses. In this case, Mediator can use one of the following to encrypt the response in the following order, depending on what is present in the security element:

- A signing certificate.
- Client name.
- WSS username, SAML token, or X.509 certificate.
- HTTP authorized user.

OR

- If Mediator can determine the current IS user from the request (that is, if an Integration Server WS-Stack determined that Subject is present), then the first principal in that subject is used.

OR

- Mediator uses either the WS-Security username token or the HTTP Basic-Auth username value. Mediator uses this approach only if all the other approaches fail to determine the public key alias. There should be a public key entry with the same name as the identified username.

Input Parameters

Encrypt By Requires that a request's XML be encrypted.

Value	Description
Element	Select this option to encrypt the entire element, which is represented by an XPath expression.
Part	Select this option to encrypt the part of soap request such as soap body or soap headers.

If Encrypt By Element is selected

Namespace (String). Namespace of the element required to be signed.

Prefix Enter the namespace prefix in the following format: `xmlns:<prefix-name>` . For example, `xmlns:soapenv`.

URI The generated XPath element in the policy should look similar to this:

```
<sp:SignedElements xmlns:sp=
"http://docs.oasis-open.org/ws-sx/
ws-securitypolicy/200702">
<sp:XPath xmlns:soapenv=
"http://schemas.xmlsoap.org/soap/envelope
/">//soapenv:Body</sp:XPath>
</sp:SignedElements>
```

Element to be Encrypted (String). An XPath expression that represents the XML element to be signed.

If Encrypt By Part is selected

Encrypt Part Mark the SOAP Body checkbox to encrypt a part of the soap request.

```
-<sp:EncryptedParts xmlns:sp=
"http://docs.oasis-open.org/ws-sx/
ws-securitypolicy/200702">
<sp:Body/>
</sp:EncryptedParts>
```

Encrypt SOAP Headers Select this option to encrypt the header of the soap request. To specify multiple headers, use the plus button to add rows and minus button to delete rows.

Name (String). A name for the SOAP header field.

Namespace (String). Namespace of the soap header required to be signed.

```
-<sp:SignedParts xmlns:sp=
"http://docs.oasis-open.org
/ws-sx/ws-securitypolicy/200702">
<sp:Body/>
<sp:Header Namespace="http://
www.w3.org/2005/08/addressing
" Name="To"/>
<sp:Header Namespace
="http://www.w3.org/2005/08/
addressing" Name="From"/>
</sp:SignedParts>
```

For example, for the following SOAP message:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
</soap:Header>
  <soap:Body>
    <catalog xmlns="http://www.store.com">
      <name>My Book</name>
      <author>ABC</author>
      <price>100</price>
    </catalog>
  </soap:Body>
</soap:Envelope>
```

The XPath expression appears as follows:

```
/soap:Envelope/soap:Body
```

Require HTTP / HTTPS

If you have a native API that requires clients to communicate with the server using the HTTP and HTTPS protocols, you can use the Require HTTP or HTTPS protocol action.

This action allows you to bridge the transport protocols between the client and the Mediator. For example, suppose you have a native API that is exposed over HTTPS and an API that receives requests over HTTP. In this situation, you can configure the API's Require HTTP or HTTPS action to accept HTTP requests and configure its Routing action to route the request to the native API using HTTPS.

Input Parameters

Protocol (String). Specifies the protocol over which the Mediator accepts requests from the client.

Note: CentraSite supports HTTP version 1.1 only.

Important:

Before you deploy an API over HTTPS, ensure that the Integration Server on which the Mediator is running has been configured for SSL. In addition, make sure you specify an HTTPS port in the Mediator's Ports Configuration page. (In the Integration Server Administrator, go to **Solutions > Mediator > Administration > General** and specify the port in the **HTTPS Ports Configuration** field.) For details on the Port Configuration page, see *Administering webMethods Mediator*.)

Value	Description
HTTP	(Default). Mediator accepts requests that are sent using the HTTP protocol.
HTTPS	Mediator accepts requests that are sent using the HTTPS protocol.

You can select *both* HTTP and HTTPS, if needed.

SOAP Version (For SOAP-based APIs). (String). Specifies the SOAP version of the requests which the Mediator accepts from the client.

Value	Description
SOAP 1.1	(Default). Mediator accepts requests that are in the SOAP 1.1 format.
SOAP 1.2	Mediator accepts requests that are in the SOAP 1.2 format.

Require JMS

If you have a native API that requires clients to communicate with the server using the Java Message Service (JMS) protocol, you can use the Require JMS protocol action.

This action allows you to bridge protocols between the client and the native API. For example, suppose you have a native API that is exposed over JMS and a client that submits SOAP requests over HTTP. In this situation, you can configure the API's Require JMS Protocol action to accept SOAP requests over JMS and configure its JMS Routing Rule action to route the request to the web service using JMS.

When this action is configured for a proxy API, you can intentionally expose an API over a JMS protocol. For example, if you have a native API that is exposed over HTTP, you might expose the API over JMS simply to gain the asynchronous-messaging and guaranteed-delivery benefits that one gains by using JMS as the message transport.

Complete the following before using the Require JMS action:

- Create an alias to a JNDI Provider (in the Integration Server Administrator, go to **Settings > Web Services**).

- To establish an active connection between Integration Server and the JMS provider, you must configure Integration Server to use a JMS connection alias (in the Integration Server Administrator, go to **Settings > Messaging > JMS Settings**).
- Create a WS (Web Service) endpoint alias for provider Web Service Descriptor (WSD) that uses a JMS binder. In the Integration Server Administrator, navigate to **Settings > Web Services** and complete the Alias Name, Description, Descriptor Type, and Transport Type fields.
- Configure a WS (Web Service) endpoint trigger (in the Integration Server Administrator, go to **Settings > Messaging > JMS Trigger Management**).
- Create a WS (Web Service) endpoint alias for consumer Web Service Descriptor (WSD) that has a JMS binder. In the Integration Server Administrator, navigate to **Settings > Web Services** and complete the Alias Name, Description, Descriptor Type, and Transport Type fields.
- Additionally, in the API's Message Flow, ensure that you delete the predefined *Require HTTP / HTTPS Protocol* action from the Receive stage. This is because, these actions are mutually exclusive.

For detailed procedures, see *webMethods Integration Server Administrator's Guide*.

Input Parameters

JMS Provider Alias	(String). Specify the name of Integration Servers, JMS provider alias. The alias should include the JNDI destination name and the JMS connection factory.
SOAP Version	(String). Specify the SOAP version of the requests which the Mediator accepts from the client.

Value	Description
SOAP 1.1	(Default). Mediator accept srequests that are in the SOAP 1.1 format.
SOAP 1.2	Mediator accepts requests that are in the SOAP 1.2 format.

Require Signing

This action is applicable only for SOAP APIs. Requires that a request's XML elements, which is represented by an XPath expression or parts of soap request such as soap body or soap headers be signed.

Prerequisites

1. Configure Integration Server: Set up keystores and truststores in Integration Server, as described in the *webMethods Integration Server Administrator's Guide*.

2. Configure Mediator: In the Integration Server Administrator, navigate to **Solutions > Mediator > Administration > General** and complete the IS Keystore Name, IS Truststore Name and Alias (signing) fields, as described in *Administering webMethods Mediator*. Mediator uses the signing alias specified in the Alias (signing) field to sign the response.

When this action is configured for a proxy API, Mediator validates that the requests are properly signed, and provides signing for responses. Mediator provides support for signing an entire SOAP message body or individual elements of the SOAP message body. Mediator uses a digital signature element in the security header to verify that all elements matching the XPath expression are signed. If the request contains elements that are not signed or no signature is present, then Mediator rejects the request.

Note:

You must map the public certificate of the key, used to sign the request, to an Integration Server user. If the certificate is not mapped, Mediator returns a SOAP fault to the caller.

Input Parameters

Sign By Requires that a request's XML be signed.

Value	Description
Element	Select this option to sign the entire element, which is represented by an XPath expression.
Part	Select this option to sign the part of soap request such as soap body or soap headers.

If Sign By Element is selected

Namespace (String). Namespace of the element required to be signed.

Prefix Enter the namespace prefix in the following format: `xmlns:<prefix-name>` . For example: `xmlns:soapenv`.

URI The generated XPath element in the policy should look similar to this:

```
<sp:SignedElements xmlns:sp=
"http://docs.oasis-open.org/ws-sx/
ws-securitypolicy/200702">
<sp:XPath xmlns:soapenv=
"http://schemas.xmlsoap.org/soap/envelope
/">//soapenv:Body</sp:XPath>
</sp:SignedElements>
```

Element Required to be Signed (String). An XPath expression that represents the XML element that is required to be signed.

If Sign By Part is selected

Sign Part Mark the SOAP Body checkbox to sign a part of the soap request.

```
-<sp:EncryptedParts xmlns:sp=
"http://docs.oasis-open.org/ws-sx
/ws-securitypolicy/200702">
<sp:Body/>
</sp:EncryptedParts>
```

Sign SOAP Headers Select this option to sign the header of the soap request. To specify multiple headers, use the plus button to add rows and minus button to delete rows.

Name	(String). A name for the SOAP header field.
Namespace	(String). Namespace of the soap header required to be signed.

```
-<sp:SignedParts xmlns:
sp="http://docs.oasis-open.org
/ws-sx/ws-securitypolicy/200702">
<sp:Body/>
<sp:Header Namespace="http://
www.w3.org/2005/08/addressing
" Name="To"/>
<sp:Header Namespace
="http://www.w3.org/2005/08/
addressing" Name="From"/>
</sp:SignedParts>
```

Consider the following example of a SOAP message:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
</soap:Header>
  <soap:Body>
    <catalog xmlns="http://www.store.com">
      <name>My Book</name>
      <author>ABC</author>
      <price>100</price>
    </catalog>
  </soap:Body>
</soap:Envelope>
```

The XPath expression for the SOAP expression appears as follows:

```
/soap:Envelope/soap:Body
```

Require SSL

This action requires that requests be sent through SSL client certificates.

When this action is configured for a proxy API, Mediator ensures that requests are sent to the server using the HTTPS protocol (SSL). The action also specifies whether the client certificate is required. This allows Mediator to verify the client sending the request. If the action requires the client certificate but the certificate is not presented, Mediator rejects the message.

When a client certificate is required by the action, the Integration Server HTTPS port should be configured to request or require a client certificate.

Note:

In Integration Server, create an HTTPS port, as described in the *webMethods Integration Server Administrator's Guide*.

Input Parameters

Client Certificate Required	Specifies whether client certificates are required for the purposes of: <ul style="list-style-type: none">■ Verifying the signature of signed SOAP requests or decrypting encrypted SOAP requests.■ Signing SOAP responses or encrypting SOAP responses.
-----------------------------	---

Require Timestamps

When this action is set for the API, Mediator requires that timestamps be included in the request header. Mediator checks the timestamp value against the current time to ensure that the request is not an old message. This serves to protect your system against attempts at message tampering, such as replay attacks.

Mediator rejects the request if either of the following happens:

- Mediator receives a timestamp that exceeds the time defined by the timestamp element.
- A timestamp element is not included in the request.

Note:

This action has no input parameters; you simply drag and drop the action into the **Message Flow** area.

Input Parameters

None.

Require WSS SAML Token

This action is applicable only for SOAP APIs and uses a WSS Security Assertion Markup Language (SAML) assertion token to validate API clients. The following subject confirmation methods are supported:

When this action is configured for a proxy API, Mediator uses a WSS Security Assertion Markup Language (SAML) assertion token to validate clients for an API. The following subject confirmation methods are supported:

- Bearer: You can select the Bearer method when the client wants a security token to be issued that does not require a proof of possession.

- **Holder of Key (Symmetric):** You can select the Holder of Key (Symmetric) method when either the client or the server needs to generate security tokens such as X509Tokens. A symmetric key is established using that security token and further signing and encryption is done using this token.
- **Holder of Key (Asymmetric):** You can select the Holder of Key (Asymmetric) method when both the client and the server have security token such as X509 certificates. In this method, the client uses its private key to sign and the recipient's (Mediator) public key to encrypt.

Note:

For information about configuring your system for SAML token processing, see *Administering webMethods Mediator*.

Important:

To configure a SAML attribute that can be used to identify the user, open the `is_jaas.cnf` file available in the `<IntegrationServerInstall_Directory>\instances\default\config` folder and modify the configuration under `WSS_Message_IS`. For example,

```
{
    /*
    * Please do not rearrange the following SoftwareAG
    * login modules; add your login modules before or after
    * these three modules
    */
    com.wm.app.b2b.server.auth.jaas.SamlAssertLoginModule requisite
    samlAttributeName="http://integration.fiserv.com/
    identity/claims/v1/FirstName";
    com.wm.app.b2b.server.auth.jaas.X509LoginModule requisite;
    com.wm.app.b2b.server.auth.jaas.BasicLoginModule requisite;
};
```

Any value can be configured for the `samlAttributeName` parameter.

Input Parameters

SAML Version (String). Specifies the WSS SAML Token version to use: 1.1 or 2.0.

SAML Subject Confirmation (String). Specifies the SAML subject confirmation methods:

Value	Description
Bearer	Select this option if the clients want a security token to be issued that does not require a proof of possession.
Holder of Key (Asymmetric)	Select this option if the clients and server use the SAML V1.1 or V2.0 Holder-of-Key method that

Note:

If the clients use SAML 2.0 Sender-Vouches tokens, configure your system as described in *Administering WebMethods Mediator*.

allows for transport of holder-of-key assertions. In this scenario, the client uses its private key to sign and the recipient's (Mediator) public key to encrypt.

Holder of Key (Symmetric) (Default). Select this option if clients use the SAML V1.1 or V2.0 Holder-of-Key method that allows for transport of holder-of-key assertions. In this scenario, the client presents a holder-of-key SAML assertion acquired from its preferred identity provider to access a web-based resource at an API provider.

WS-Trust Version (String). Specifies the WSS SAML Token version to use: 1.1 or 2.0.

Algorithm Suite Select any algorithm suite that is defined by the WS-SecurityPolicy specification. For example, Basic128, Basic256, TripleDes, and so on.

Encrypt Signature To encrypt the signature. Select either of the following:

- Yes: To encrypt the signature.
- No: Not to encrypt the signature.

Layout Specifies a requirement for a particular security header layout.

Holder of Key Asymmetric Parameter The public key is shared with Mediator and the private key is secure.

<u>Value</u>	<u>Description</u>
Initiator Token Inclusion	Identifies the inclusion value for the client's security token assertion.
Recipient Token Inclusion	Identifies the inclusion value for the recipient's security token assertion.

Holder of Key Symmetric Parameter Encrypts the signature, soap header, and body.

<u>Value</u>	<u>Description</u>
Initiator Token Inclusion	Identifies the inclusion value for the client's security token assertion.
Recipient Token Inclusion	Identifies the inclusion value for the recipient's security token assertion.

Issuer Address Specifies the SAML issuer address. For example, `<saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer>`

Metadata Reference Address The address from where the metadata reference document can be obtained.

Key Size The number of bits in a key used by a cryptographic algorithm. For example, 256 bits.

Request Security Token Template Parameters Defines extensions to the <wst:RequestSecurityToken> element for requesting specific types of keys, algorithms, or key and algorithms, as specified by a given policy in the return token(s). In some cases, the service may support a variety of key types, sizes, and algorithms. These parameters allow a requestor to indicate its desired values. The issuer's policy indicates if input values must be adhered to and faults generated for invalid inputs, or if the issuer must provide alternative values in the response.

<u>Value</u>	<u>Description</u>
Key	Key type of the security token template.
Value	String. A value for the request token.

SAML Authentication

Used when native API enforces SAML authentication. Based on the modes selected, Mediator either uses the WSS Username mode to obtain the SAML assertion token from STS to invoke the native service or it uses the Kerberos Over Transport mode to obtain the SAML token and assertion to access the native API.

Input Parameters

Signing Alias Specifies the alias (key) used when signing the message.

Encryption Alias Specifies the alias (key) used when encrypting the message.

Issuer Communication (String). Specifies information about the issuer of a SAML assertion and the ways to communicate with the native API.

Action	Actions performed by the issuers.	
	<u>Value</u>	<u>Description</u>
	Act as Delegation	The user delegates the request to another user. The user delegates the SAML request to the delegator. The delegator uses a signature element to authenticate the SAML request.

Normal Client Client requesting the SAML token.

Communicate Using Modes through which the communication can occur.

Value	Description
WSS Username (Message)	The WSS username token supplied in the header of the SOAP request that the consumer application submits to the virtual service.
Kerberos Over Transport (Message)	Trasports the Kerberos token over the Transport Layer Security (TLS) protocol to provide additional security features.

WSS Username Configuration Credentials for the WSS Username Configuration.

Username	(String). The username of the wss configuration.
Password	(String). The password to be used together with the Username parameter as authentication credentials.

Endpoint Endpoint of the service.

SAML Version (String). Specifies the WSS SAML Token version to use: 1.1 or 2.0.

WS- Trust Version (String). Specifies the WSS SAML Token version to use: 1.1 or 2.0.

Applies To (Optional). Specifies the scope for which this security token is required. For example, the services to which this token is applied.

Extended Parameters Other additional parameters.

Value	Description
Key Size	The number of bits in a key used by a cryptographic algorithm. For example, 256 bits.
Key Type	The type of key used in the security token.
SignatureAlgorithm	The signature algorithm used to sign the issued token.
EncryptionAlgorithm	The encryption algorithm used to encrypt the issued token.

CanonicalizationAlgorithm	The canonicalization algorithm used when signing the issued token.
ComputedKeyAlgorithm	The key derivation algorithm to use if using a symmetric key for the proof key, where proof key is computed using client, server, or combined entropy.
Encryption	The key to use when encrypting the issued token.
ProofEncryption	The key to use when encrypting the proof key.
KeyWrapAlgorithm	The algorithm used to encrypt the symmetric key.
SignWith	The signature algorithm the client intends to employ when using the proof key to sign.
EncryptWith	Indicates the symmetric algorithm that client uses to protect messages sent to the server when using the proof key.

Set Custom Headers

When this action is configured for a proxy API, Mediator includes custom HTTP headers to the client requests before submitting to the native APIs.

Input Parameters

Header (String). Mediator uses the HTTP headers that you specify in the **Name** and **Value** columns below. If you need to specify multiple headers, use the plus button to add rows.

Value	Description
Name	(String). A name for the HTTP header field. The header field name (\$field) is not case sensitive.
Value	(String). A value for the HTTP header field.

Sample

Let's imagine you have a Name field Authorization. This is encoded in Base64 scheme as follows: QXV0aG9yaXphdGlvbg==.

Set JMS Headers

Every JMS message includes message header properties that are always passed from provider to client. The purpose of the header properties is to convey extra information to the client outside the normal content of the message body.

When this action is configured for a proxy API, Mediator uses the JMS header properties to authenticate client requests before submitting to the native APIs.

To use this action the following prerequisites must be met:

- Create an alias to a JNDI Provider (in the Integration Server Administrator, go to **Settings > Web Services**).
- To establish an active connection between Integration Server and the JMS provider, you must configure Integration Server to use a JMS connection alias (in the Integration Server Administrator, go to **Settings > Messaging > JMS Settings**).
- Create a WS (Web Service) endpoint alias for provider Web Service Descriptor (WSD) that uses a JMS binder. In the Integration Server Administrator, navigate to **Settings > Web Services** and complete the Alias Name, Description, Descriptor Type, and Transport Type fields.
- Configure a WS (Web Service) endpoint trigger (in the Integration Server Administrator, go to **Settings > Messaging > JMS Trigger Management**).
- Create a WS (Web Service) endpoint alias for consumer Web Service Descriptor (WSD) that has a JMS binder. In the Integration Server Administrator, navigate to **Settings > Web Services** and complete the Alias Name, Description, Descriptor Type, and Transport Type fields.

For detailed instructions, see *webMethods Integration Server Administrator's Guide*.

Input Parameters

Header (String). The JMS message headers that Mediator uses to authenticate incoming requests for the native API. To add additional rows, use the plus button.

Value	Description
Name	(String). A name for the JMS message header field. The header field name (\$field) is not case sensitive.
Value	(String). A value for the JMS message header field.

Settable JMS Header Properties

Property Name	Property Type	Getter Method
Message ID	string	getJMSMessageID()
Priority	int	getJMSPriority()
Time To Live	long	getTimeToLive()
Delivery Mode	int	getJMSDeliveryMode()
Message Expiration	long	getJMSExpiration()

Property Name	Property Type	Getter Method
Correlation ID	string	getJMSCorralationID()
Redelivered	boolean	getJMSRedelivered()
Time Stamp	long	getJMSTimeStamp()
Type	string	getJMSType()

Set Media Type

This action specifies the content type for a REST request or response. You would need to configure the **Set Media Type** action to:

- Specify the content type for a REST request received from a client. If the content type header is missing in a client request sent to an API, Mediator adds the content type specified here before sending the request to the native API.
- Specify the content type for a REST response (both for a REST API and SOAP API exposed as REST) before sending the response to the client. If a native API response (both for a REST API and SOAP API exposed as REST) does not contain a content type header, Mediator adds the content type specified here before sending the response to the client.

Note:

This action is applicable only if the action **Enable REST Support** is set for a SOAP API.

Input Parameters

Media Type (String). The content type header that CentraSite uses to send the REST request or response.

Examples for content types: `application/json`, `application/xml`

Set Message Properties

The message property fields are similar to header fields described previously in the *Set JMS Headers* action, except these fields are set exclusively by the consumer application. When a client receives a message, the properties are in read-only mode. If a client tries to modify any of the properties, a `MessageNotWriteableException` is thrown.

The properties are standard Java name or value pairs. The property names must conform to the message selector syntax specifications defined in the `Message` interface.

Property fields are most often used for message selection and filtering. By using a property field, a message consumer can interrogate the property field and perform message filtering and selection.

When this action is configured for a proxy API, Mediator uses the message properties to authenticate client requests before submitting to the native APIs.

To use this action the following prerequisites must be met:

- Create an alias to a JNDI Provider (in the Integration Server Administrator, go to **Settings > Web Services**).
- To establish an active connection between Integration Server and the JMS provider, you must configure Integration Server to use a JMS connection alias (in the Integration Server Administrator, go to **Settings > Messaging > JMS Settings**).
- Create a WS (Web Service) endpoint alias for provider Web Service Descriptor (WSD) that uses a JMS binder. In the Integration Server Administrator, navigate to **Settings > Web Services** and complete the Alias Name, Description, Descriptor Type, and Transport Type fields.
- Configure a WS (Web Service) endpoint trigger (in the Integration Server Administrator, go to **Settings > Messaging > JMS Trigger Management**).
- Create a WS (Web Service) endpoint alias for consumer Web Service Descriptor (WSD) that has a JMS binder. In the Integration Server Administrator, navigate to **Settings > Web Services** and complete the Alias Name, Description, Descriptor Type, and Transport Type fields.

For detailed information, see *webMethods Integration Server Administrator's Guide*.

Input Parameters

Property (String). The custom message properties Mediator uses to authenticate incoming requests for the native API. To add additional rows, use the plus button.

<u>Value</u>	<u>Description</u>
Name	(String). The name of the property.
Value	(String). The value of the property.

Straight Through Routing

This action routes the incoming requests to the Mediator directly to the native API.

1. Configure Integration Server: Set up keystores and truststores in Integration Server, as described in the *webMethods Integration Server Administrator's Guide*.
2. Configure Mediator: In the Integration Server Administrator, navigate to **Solutions > Mediator > Administration > General** and complete the IS Keystore Name, IS Truststore Name and Alias (signing) fields, as described in *Administering webMethods Mediator*.

When this action is configured for an API, Mediator ensures that requests from the client are parsed directly to the native API you specify. This action also includes a set of configuration properties for the Mediator to process the incoming requests for the native API.

Input Parameters

Route To (URI). Type the URL of the native API endpoint to route the request to in case all routing rules evaluate to False. For example:

```
http://mycontainer/creditCheckService
```

Click the **Configure Endpoint Properties** icon (next to the **Route To** field) if you want to configure a set of properties for the specified endpoint.

Alternatively, Mediator offers Local Optimization capability if the native endpoint is hosted on the same Integration Server as Mediator. With local optimization, API invocation happens in-memory and not through a network hop.

Specify the native API in either of the following forms:

```
local://<Service-full-path>
```

OR

```
local://<server>:<port>/ws/<Service-full-path>
```

For example:

```
local://MyAPIFolder:MyLocalAPI
```

which points to the endpoint API `MyLocalAPI` which is present under the folder `MyAPIFolder` in Integration Server.

Note:

Local Optimization is not applicable to REST based APIs.

Configure Endpoint Properties  (icon)

(Optional). This icon displays the **Endpoint Properties** dialog box that enables you to configure a set of properties for the Mediator to route incoming requests to the native API as follows:

SOAP Optimization Method (For SOAP-based APIs). Specifies the optimization methods that Mediator can use to parse SOAP requests to the native API.

Value	Description
MTOM	Mediator uses the Message Transmission Optimization Mechanism (MTOM) to parse SOAP requests to the API.
SwA	Mediator uses the SOAP with Attachment (SwA) technique to parse SOAP requests to the API.

None (Default). Mediator does not use any optimization method to parse the SOAP requests to the API.

Note:

Keep the following points in mind:

- Bridging between SwA and MTOM is not supported. If a client sends a SwA request, Mediator can only forward SwA to the native API. The same is true for MTOM, and applies to responses received from the native API. That is, a SwA or MTOM response received by Mediator from a native API is forwarded to the client using the same format it received.
- When sending SOAP requests that do not contain a MTOM or SWA attachment to a native API that returns an MTOM or SWA response, the request Accept header must be set to multipart/related. This is necessary so Mediator knows how to parse the response properly.

HTTP Connection Timeout (Number). (Optional). Specifies the time interval (in seconds) after which a connection attempt timeouts. If a value 0 is specified (or if the value is not specified), Mediator uses the value specified in the `Connection Timeout` field (in the Integration Server Administrator, go to **Settings > Extended**). Default: 30 seconds.

Read Timeout (Number). (Optional). The time interval (in seconds) after which a socket read attempt will timeout.

The precedence of the `Read Timeout` configuration is as follows:

1. If a value is specified for the `Read Timeout` field in the routing endpoint alias, Mediator will use the value specified in the **Runtime Alias > Endpoint Alias > Endpoint Properties > Read Timeout** field. The read timeout value defined at an alias level takes precedence over the timeout values defined at an API level and the global configuration.
2. If a value 0 is specified (or if the value is not specified) for the `Read Timeout` field in the

routing endpoint alias, then Mediator will use the value specified in the `Read Timeout` field of this routing action. The read timeout value defined at an API level takes precedence over the global configuration.

3. If a value 0 is specified (or if the value is not specified) for the `Read Timeout` field in this routing action (at an API level), then Mediator will use the value of the global property `pg.endpoint.readTimeout` located in the file `Integration Server_directory/packages/WmMediator/config/resources/pg-config.properties` (in the Mediator Administration console, go to **> Settings > Extended Settings > pg.endpoint.readTimeout** property.).

Note:

If a value for the `Read Timeout` configuration is not specified in any of the above configuration parameters, then Mediator will use the default 30 seconds.

SSL Configuration Optional). To enable SSL client authentication that Mediator uses to authenticate incoming requests for the native API, you must specify values for both the `Client Certificate Alias` field and the `IS Keystore Alias` field. If you specify a value for only one of these fields, a deployment error occurs.

Note:

SSL client authentication is optional; you may leave both fields blank.

Prerequisite: You must set up the key alias and keystore properties in the Integration Server. For the procedure, see *webMethods Integration Server Administrator's Guide*.

Use these properties to specify the following fields:

<u>Value</u>	<u>Description</u>
<code>Client Certificate Alias</code>	The client's private key to be used for performing SSL client authentication.
<code>IS Keystore Alias</code>	The keystore alias of the instance of Integration Server on which

Mediator is running. This value (along with the value of Client Certificate Alias) is used for performing SSL client authentication.

WS Security Header (For SOAP-based APIs). Indicates whether Mediator should pass the WS-Security headers of the incoming requests to the native API.

Value	Description
Remove processed security headers	(Default). Removes the security header if it is processed by Mediator (that is, if Mediator processes the header according to the API's security run-time action). Mediator does <i>not</i> remove the security header if <i>both</i> of the following conditions are true: 1) Mediator did not process the security header, and 2) the <code>mustUnderstand</code> attribute of the security header is 0/false).
Pass all security headers	Passes the security header, even if it is processed by Mediator (that is, even if Mediator processes the header according to the API's security action).

Throttling Traffic Optimization

This action limits the number of API invocations during a specified time interval, and sends alerts to a specified destination when the performance conditions are violated.

Reasons for limiting the API invocation traffic include:

- To avoid overloading the back-end services and their infrastructure.
- To limit specific clients in terms of resource usage (that is, you can use the Monitor Service Level Agreement action to monitor performance conditions for a particular client, together with Throttle API Usage to limit the resource usage).
- To shield vulnerable servers, services, and even specific operations.
- For API consumption metering (billable pay-per-use APIs).

Note:

To enable Mediator to publish performance metrics, you must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to **Solutions >**

Mediator > Administration > CentraSite Communication). For the procedure, see *Administering webMethods Mediator*.

Input Parameters

Soft Limit (Number). (Optional). The maximum number of invocations allowed per `Interval Value` before issuing an alert. Reaching the soft limit does not affect further processing of requests (until the `Hard Limit` is reached).

Note:

The limit is reached when the total number of invocations coming from *all* the clients (specified in the `Limit Traffic for Applications` field) reaches the limit. `Soft Limit` is computed in an asynchronous manner; thus when multiple requests are made at the same time, it may be possible that the `Soft Limit` alert is not strictly accurate.

Alert Message for Soft Limit (String). (Optional). A text message to include in the soft limit alert.

Hard Limit The maximum number of invocations allowed per `Interval Value` before stopping the processing of further requests and issuing an alert. Typically, this number should be higher than the soft limit.

Note:

The limit is reached when the total number of invocations coming from *all* the clients (specified in the `Limit Traffic for Consumers` field) reaches the limit. `Hard Limit` is computed in an asynchronous manner; thus when multiple requests are made at the same time, it may be possible that the `Hard Limit` alert is not strictly accurate.

Alert Message for Hard Limit (String). (Optional). A text message to include in the hard limit alert.

Alert for Consumer Applications (String). The consumer application that this action applies to. To specify multiple consumers, use the plus button to add rows, or select **Any Consumer** to apply this action to any consumer application.

Alert Interval (String). The amount and unit (Minutes, Hours, Days or Weeks) of time for the soft limit and hard limit to be reached.

Alert Frequency (String). Frequency to issue alerts.

<u>Value</u>	<u>Description</u>
Every Time	(Default). Issues an alert every time the specified condition is violated.

Only Once Issues an alert only the first time the specified condition is violated.

Alert Destination (String). (Optional). A place to log the alerts.

Important:

Ensure that Mediator is configured to send event notifications to the destination you specify here. For details about alerts and transaction logging, see *Administering webMethods Mediator*.

Value	Description
API Portal	<p>Mediator can use API Portal destination to publish data about run-time events and key performance indicator (KPI) metrics.</p> <p>Mark the checkbox API Portal and select an API Portal instance from the drop-down list. When you mark the checkbox, CentraSite displays the list of API Portal instances that are registered and are available to you. To specify multiple API Portal instances, use the plus button to add rows.</p> <p>Note: When you publish an API that is configured with an API Portal destination to Mediator, CentraSite automatically creates an entry of the configured API Portal destination in the Integration Server Administrator's Solutions > Mediator > Administration > API Portal page.</p> <p>Mediator uses the API Portal destination to publish the following event types:</p> <ul style="list-style-type: none"> ■ Transaction Events ■ Monitoring Events ■ Lifecycle Events ■ Policy Violation Events ■ Error Events ■ Performance Metrics <p>You can also use the Mediator capability to specify the intervals at which the events and KPI metrics must be published to the API Portal destination.</p>

CentraSite	<p>Prerequisite: You must configure CentraSite to communicate with API Portal using the Add Gateway action.</p>
	<p>Sends the alerts to the API's Events profile in CentraSite.</p>
Elasticsearch	<p>Prerequisite: You must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to Solutions > Mediator > Administration > CentraSite Communication). For information about how to configure communication with CentraSite, see <i>Administering webMethods Mediator</i>.</p>
	<p>Mediator can use Elasticsearch destination to publish data about run-time events and key performance indicator (KPI) metrics.</p>
	<p>Mediator uses the Elasticsearch destination to publish the following event types:</p>
	<ul style="list-style-type: none"> ■ Transaction Events ■ Monitoring Events ■ Lifecycle Events ■ Policy Violation Events ■ Error Events ■ Performance Metrics
	<p>You can also use the Mediator capability to specify the intervals at which the events and KPI metrics must be published to the Elasticsearch destination.</p>
	<p>Prerequisite: You must configure the Elasticsearch destination (in the Integration Server Administrator, go to Solutions > Mediator > Administration > Elasticsearch). For the procedure, see <i>Administering webMethods Mediator</i>.</p>
Local Log	<p>Sends the alerts to the server log of the Integration Server on which Mediator is running.</p>
	<p>Also select a value in the <code>Log Level</code> field:</p>
	<ul style="list-style-type: none"> ■ Info: Logs error-level, warning-level, and informational-level alerts.

- Warn: Logs error-level and warning-level alerts.
- Error: Logs only error-level alerts.

Important:

The Integration Server Administrator's logging level for Mediator should match the logging level specified for this action (go to **Settings > Logging > Server Logger**).

SNMP	<p>Sends the alerts to CentraSite's SNMP server or a third-party SNMP server.</p> <p>Prerequisite: You must configure the SNMP server destination (in the Integration Server Administrator, go to Solutions > Mediator > Administration > SNMP). For the procedure, see <i>Administering webMethods Mediator</i>.</p>
Email	<p>Sends the alerts to an SMTP email server, which sends them to the email address(es) you specify here. To specify multiple addresses, use the plus button to add rows.</p> <p>Prerequisite: You must configure the SMTP server destination (in the Integration Server Administrator, go to Solutions > Mediator > Administration > Email). For the procedure, see <i>Administering webMethods Mediator</i>.</p>
EDA/Database	<p>Sends the alerts to an EDA endpoint or Database destination that you configured in Integration Server Administrator:</p> <ul style="list-style-type: none"> ■ An EDA endpoint (that is, a default endpoint configured in the universal messaging configuration). ■ A Database (that is, a JDBC connection pool is defined in Integration Server and associated with the Mediator functional alias). <p>Prerequisite: You must configure the EDA/Database destination in Integration Server on the Solutions > Mediator > Administration > EDA/Database Configuration page. For details, see <i>Administering webMethods Mediator</i>.</p>

Service Result Cache

This action enables caching of the results of SOAP and REST API invocations based on the caching criteria that you define. If the Service Result Cache action is set for an API, Mediator enables caching for the API when the API is deployed using the information for the action received from CentraSite. You can define the elements for which the API responses are to be cached based on the criteria: HTTP Header, Path, or XPath Expression. You can also limit the values to store in the cache using a whitelist. And, for the elements that are stored in the cache, you can specify other parameters such as Time to Live (TTL) and maximum response payload size.

Note:

In the case of REST based APIs, caching is supported only for the results of the HTTP method GET and not for other methods. Service result cache is not supported for a JSON request in a REST based API but is supported for a JSON response.

Caching the results of an API request:

- Increases the throughput of the API call
- Improves the scalability of the API

Note:

We recommend the use of caching only for elements that do not require live data and state values.

Input Parameters

Configure Caching Based On Select a caching criteria. Mediator uses this information to determine the request component that is the actual payload based on which the results of the API invocation are cached. The options are:

- **HTTP Header** Uses the HTTP header in the API request. You can use this criteria for REST based APIs that accept payloads only in HTTP format.
- **Path** Uses the entire invocation URL including the query parameter. This option is applicable mainly for REST based API requests that use the URL or path parameter as the payload.
- **XPath Expression** Uses the XPath expression in the API request. You can use this criteria for: SOAP based API requests whose payload is a SOAP envelope, and REST based API requests that accept payloads in XML format.

Header Name (Only if you select HTTP Header).

Specifies the HTTP header name.

Namespace (Only if you select XPath Expression).

(Optional). Specifies the namespace of the XPath expression:

- Prefix - The prefix for the namespace. For example, soapenv or axis
- URI - The namespace URI - For example,
http://schemas.xmlsoap.org/soap/envelope/ or
http://ws.apache.org/axis

XPath Expression (Only if you select XPath Expression).

Specifies the XPath expression in the API request.

Add to Whitelist (Only if you select HTTP Header or XPath Expression).

(Optional). Mediator caches the API responses only for requests whose cache criteria matches with the one set for the action, and whose criteria evaluation results in any one of the values in this list.

Time to Live (Optional). Specifies the lifespan (Days Hours Minutes, for example: 5d 4h 1m) of the elements in the cache after which the elements are considered out-of-date.

Note:

The maximum value that you can set for the Time to Live (TTL) parameter is 24855 days (unlimited). If no value is specified, the TTL is unlimited (does not expire). If you set the TTL value to 0d 0h 0m, the API results are not cached.

Maximum Response Payload Size Specifies the maximum payload size for the API in kilo bytes.

The value -1 stands for unlimited payload size.

Mediator uses the Ehcache capability provided by Integration Server to cache the results of the API calls. You can configure caching for a single Mediator node or for a cluster. For details on configuring Ehcache, see *webMethods Integration Server Administrator's Guide*.

Recommendations and Best Practices for Service Result Caching

This section provides guidance on the use of caching for the results of an API request.

- Caching is not recommended in cases where consumers absolutely rely on current information retrieved from a back-end service.
- As caching occurs in memory, ensure that you are not operating in a memory constrained environment.
- Balance the memory consumption with the API response sizes. If your API returns huge responses or large binary attachments, limit caching by specifying the maximum size of the cache or by defining data eviction policies to avoid excessive memory consumption.
- Design the APIs for which you want to implement caching to be idempotent to support reliable caching.

- Configure a suitable value for the Time to Live (TTL) parameter for the cache entries based on your business needs and the use cases for your API.

For example, if your API serves static product catalog data which is only updated once a quarter select a large TTL value. If your API serves for example environmental data where a certain age of the data is tolerated, select a TTL value like 15 minutes.

Note:

A TTL value that you set in a standalone Mediator is reset if the Mediator restarts. For example, if you set 15 minutes as the TTL value of an element which is inserted after 10 minutes, and Mediator restarts. The element has a TTL value of 15 minutes after the Mediator is restarted.

- Caching of the results of an API call is recommended:
 - If the response time of a direct query to the database is high or subject to high latencies, and if the data requested does not change frequently or is static.
 - If the client applications can use slightly outdated, cached data. For example, a weather API can be supplied data from a cache which is an hour old.
 - If there are temporary service interruptions on the server side, to mitigate these interruptions with data from the cache.
 - If the API that requests data is subject to traffic management rules, usage quotas, or if billing is based on the number of calls to the API, you can overcome these by using cached data.
- Caching may not be effective or is not recommended:
 - If the response of a direct request for data to the database is very fast and scaling is not an issue, using cached data may not provide additional benefits.
 - If your API uses non-idempotent requests.

Validate SAML Audience URIs

The Validate SAML Audience URIs policy is used to validate the Audience Restriction in the conditions section of the SAML assertion. It verifies whether any of the valid Audience URI within one valid condition element in SAML assertion matches with any of the configured URI. If two conditions are available, then one of the audience URIs in the first condition, and one of the audience URIs in the second condition must match with any of the configured URIs in this policy for the virtual service.

This policy is used in the following scenarios:

- When the native service is enforced with the SAML policy and if the service provider wants to delegate Audience Restriction validation to Mediator.
- When SAML policy is enforced for the virtual service in Mediator.

For more information on Audience URI, see conditions and audience restriction sections in the SAML specification available in the <https://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf> location.

Input Parameters

URI (URI). The audience URI.

Match Criteria To match the values, select one of the following values:

Value	Description
Allow Sublevels	Any one of the audience URI in the incoming SAML assertion either has to be an exact match or it can have sub paths to the configured URI. For example, if <code>http://yahoo.com</code> is configured as the URI and the Allow Sublevels option is selected, the audience URI has <code>http://yahoo.com/mygroup</code> and condition is matched because the main URI matches with the configured URI (<code>http://yahoo.com</code>). The extra path <code>mygroup</code> is a sublevel path.
Exact match	(Default). Any one of the audience URI in the incoming SAML assertion is verified for the exact match with the configured URI. For example, if <code>http://yahoo.com</code> is configured as the URI and the Exact match option is selected, the audience URI must be configured with <code>http://yahoo.com</code> in order to match the condition.

Validate Schema

This action validates all XML request and response messages against an XML schema referenced in the WSDL.

Mediator can enforce this action for messages sent between APIs. When this action is configured for a proxy API, Mediator validates XML request messages, response messages, or both, against the XML schema referenced in the WSDL.

Input Parameters

Validate SOAP Message(s) (Object). Validates request and response messages. You may select both Request and Response.

Value	Description
Request	Validate all requests.
Response	Validate all responses.

Important:

Be aware that Mediator does not remove `wsu:Id` attributes that may have been added to a request by a client as a result of security operations against request elements (that is, signatures and encryptions). In this case, to avoid schema validation failures you would have to add a [“Request Transformation” on page 1301](#) action or a [“Response Transformation” on page 1301](#) action to the API so that the requests and responses are passed to an XSL transformation file that removes the `wsu:Id` attribute.

Built-In Run-Time Actions Reference (CentraSite Control)

This section describes the built-in run-time actions that you can include in run-time policies for Virtual Services. You use these actions only when you are using CentraSite Control to create run-time policies for Virtual Services. The content is organized under the following sections:

Summary of Actions in the WS-SecurityPolicy Category

Mediator provides two kinds of actions that support WS-SecurityPolicy 1.2: authentication actions and XML security actions.

Authentication Actions (WS-SecurityPolicy 1.2)

Mediator uses the following authentication actions to verify that the requests for virtual services contain a specified WS-Security element:

Require WSS Username Token Uses WS-SecurityPolicy authentication to validate user names and passwords that are transmitted in the SOAP message header for the WSS Username token.

Require WSS X.509 Token Identifies consumers based on a WSS X.509 token.

Require WSS SAML Token Uses a WSS Security Assertion Markup Language (SAML) assertion token to validate service consumers.

XML Security Actions (WS-SecurityPolicy 1.2)

These actions provide confidentiality (through encryption) and integrity (through signatures) for request and response messages.

Require Signing Requires that a request's XML element (which is represented by an XPath expression) be signed.

Require Encryption Requires that a request's XML element (which is represented by an XPath expression) be encrypted.

Require SSL Requires that requests be sent through SSL client certificates and can be used by both SOAP and REST services.

Require Timestamps Requires that timestamps be included in the request header. Mediator checks the timestamp value against the current time to ensure that the request is not an old message. This serves to protect your system against attempts at message tampering, such as replay attacks.

Summary of Actions in the Monitoring Category

Mediator provides the following run-time monitoring actions:

Monitor Service Performance This action monitors a user-specified set of run-time performance conditions for a virtual service and sends alerts to a specified destination when these performance conditions are violated.

Monitor Service Level Agreement This action provides the same functionality as Monitor Service Performance but this action is different because it enables you to monitor a virtual service's run-time performance especially for particular consumer(s). You can configure this action to define a *Service Level Agreement (SLA)*, which is set of conditions that defines the level of performance that a specified consumer should expect from a service.

Throttling Traffic Optimization (Not available in Mediator versions below 9.0.) This action limits the number of service invocations during a specified time interval and sends alerts to a specified destination when the performance conditions are violated. You can use this action to avoid overloading the back-end services and their infrastructure, to limit specific consumers in terms of resource usage, and so on.

Summary of Actions in the Additional Category

Mediator provides the following actions, which you can use in conjunction with the actions above:

Identify Consumer You use this action in conjunction with an authentication action (Require WSS Username Token, Require WSS X.509 Token, or Require HTTP Basic Authentication). Alternatively, you can use this action alone to identify consumers only by host name or IP address.

Require HTTP Basic Authentication This action uses HTTP basic authentication to verify the consumer's authentication credentials contained in the request's Authorization header against the Integration Server's user account.

Authorize User This action authorizes consumers against a list of users and a list of groups registered in the Integration Server on which Mediator is running. You use this action in conjunction with an authentication action Require WSS Username Token, Require WSS SAML Token, or Require HTTP Basic Authentication.

Authorize Against Registered Consumers	This action authorizes consumer applications against all consumer applications who are registered in CentraSite as consumers for the service.
Log Invocations	Logs request or response payloads to a destination you specify.
Validate Schema	Validates all XML request or response messages against an XML schema referenced in the WSDL.

Configuring Destinations for Alerts and Logs

You can configure the destinations to which Mediator should send the alerts and transaction logging payloads generated by the built-in run-time actions that you include in run-time policies for virtual services.

The destinations can be configured at the:

- **Service level**, for the event types: Transaction Event and Monitoring Event
- **Global level** (for all services), for metrics and the event types: Policy Violation Event, Error Event, and Lifecycle Event

You can configure one of the following destinations for alerts:

- CentraSite
- Local Log
- SNMP (CentraSite SNMP server or third-party SNMP server configured in Integration Server)
- E-mail
- EDA/Database (EDA endpoint or a database that you configured in Integration Server)

You can select one of the following destinations for transaction payloads:

- CentraSite
- Local Log
- SNMP (CentraSite SNMP server or third-party SNMP server configured in Integration Server)
- E-mail
- Audit Log (only for Log Invocation)
- EDA/Database (EDA endpoint or a database that you configured in Integration Server)

You can configure the EDA/Database destination for the following policy actions:

- Log Invocation
- Monitor Service Level Agreement
- Monitor Service performance

■ Throttling Traffic Optimization

The `watt.server.auth.skipForMediator` Property

This property specifies whether Integration Server authenticates requests for Mediator. You must set this property to true.

No request to Mediator should be authenticated by Integration Server. Instead, authentication should be handled by Mediator. Thus, to enable Mediator to authenticate requests, you must set `skipForMediator` to true (by default it is false).

When this parameter is set to true, Integration Server skips authentication for Mediator requests and allows the user credentials (of any type) to pass through so that Mediator can authenticate them. If you change the setting of this parameter, you must restart Integration Server for the changes to take effect.

> To set `skipForMediator` to true

1. In the Integration Server Administrator, click **Settings > Extended**.

2. Click **Show and Hide Keys**.

Look for the `watt.server.auth.skipForMediator` property and ensure it is set to true.

3. If the `watt.server.auth.skipForMediator` property is not present, add it as follows:

a. Click **Edit Extended Settings**.

b. Type `watt.server.auth.skipForMediator=true` on a separate line.

c. Click **Save**.

d. Restart Integration Server.

Usage Cases for Identifying or Authenticating Consumers

When deciding which type of identifier to use to identify a consumer application, consider the following points:

- Whatever identifier you select to identify a consumer application, it must be unique to the application. Identifiers that represent user names are often not suitable because the identified users might submit requests for multiple applications.
- Identifying applications by IP address or host name is often a suitable choice, however, it does create a dependency on the network infrastructure. If a consumer application moves to a new machine, or its IP address changes, you must update the identifiers in the application asset.
- Using X.509 certificates or a custom token that is extracted from the SOAP message itself (using an XPATH expression), is often the most trouble-free way to identify a consumer application.

Following are some common combinations of actions used to authenticate or identify consumers:

■ **Scenario 1: Identify consumers by IP address or host name**

- The simplest way to identify consumers is to use the Identify Consumer action and set its `Identify User Using` parameter to specify either a host name or an IP address (or a range of IP addresses).

■ **Scenario 2: Authenticate consumers by HTTP authentication token**

Use the following actions:

- Identify Consumer action and set its `Identify User Using` parameter to HTTP Authentication Token (to identify consumers using the token derived from the HTTP header).
- Require HTTP Basic Authentication.
- Additionally, you can use one or both of the following:
 - Authorize User action (to authorize a list of users and groups registered in the Integration Server on which Mediator is running).
 - Authorize Against Registered Consumers action (to authorize consumer applications against all Application assets registered as consumers for a service in CentraSite).

■ **Scenario 3: Authenticate consumers by WS-Security authentication token**

Use the following actions:

- Identify Consumer action, and set its `Identify User Using` parameter to WS-Security Authentication Token (to identify consumers using the token derived from the WSS Header).
- Require WSS Username Token action.
- Additionally, you can use one or both of the following:
 - Authorize User action (to authorize a list of users and groups registered in the Integration Server on which Mediator is running).
 - Authorize Against Registered Consumers action (to authorize consumer applications against all Application assets registered as consumers for a service in CentraSite).

■ **Scenario 4: Authenticate consumers by WSS X.509 token**

- Identify Consumer action, and set its `Identify User Using` parameter to Consumer Certificate (to identify consumers using the WSS X.509 token).
- Require WSS X.509 Token action.
- Require SSL action.

Built-in Actions for Run-Time Policies (CentraSite Control)

This section describes the built-in run-time actions that you can include in run-time policies for virtual services.

Authorize Against Registered Consumers

Note:

Dependency requirement: A policy that includes this action must also include the Identify Consumer action. However, if the Identify Consumer action is set to identify users through the **HTTP Authentication Token** option, then Authorize Against Registered Consumers should not be included in the policy.

Authorizes consumer applications against all consumer applications who are registered in CentraSite as consumers for the service.

Input Parameters

None.

Authorize User

Note:

Dependency requirement: A policy that includes this action must also include *one* of the following: the Require WSS SAML Token action or the Identify Consumer action with one of the following options selected: HTTP Authentication Token or WS-Security Authentication Token.

Authorizes consumers against a list of users and a list of groups registered in the Integration Server on which Mediator is running.

Input Parameters

Perform authorization against *Boolean*. Authorizes consumers against a list of users who list of users are registered in the Integration Server on which Mediator is running. Specify one or more users in the fields below this option.

Perform authorization against *Boolean*. Authorizes consumers against a list of groups list of groups who are registered in the Integration Server on which Mediator is running. Specify one or more groups in the fields below this option.

Note:

By default, both of the input parameters are selected. If you de-select one of these parameters, the fields showing the list of users (or groups) is not displayed.

Identify Consumer

Mediator uses this action to identify consumer applications based on the kind of consumer identifier (IP address, HTTP authorization token, and so on.) you specify. Alternatively, this action provides an option to allow anonymous users to access the assets.

Input Parameters

Anonymous Usage Allowed	<i>Boolean</i> . Specifies whether to allow all users to access the asset, without restriction.								
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>False</td> <td><i>Default</i>. Allows only the users specified in the Identify User Using parameter to access the assets.</td> </tr> <tr> <td>True</td> <td>Allow all users to access the asset. In this case, do not configure the Identify User Using parameter.</td> </tr> </tbody> </table>	Value	Description	False	<i>Default</i> . Allows only the users specified in the Identify User Using parameter to access the assets.	True	Allow all users to access the asset. In this case, do not configure the Identify User Using parameter.		
Value	Description								
False	<i>Default</i> . Allows only the users specified in the Identify User Using parameter to access the assets.								
True	Allow all users to access the asset. In this case, do not configure the Identify User Using parameter.								
Identify User Using	<i>String</i> . Specifies the kind of consumer identifier that the action uses to identify consumer applications.								
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>IP Address</td> <td>Identifies one or more consumer applications based on their originating IP addresses.</td> </tr> <tr> <td>Host Name</td> <td>Identifies consumer applications based on a host name.</td> </tr> <tr> <td>HTTP Authentication Token</td> <td>Uses HTTP Basic authentication to verify the consumer's authentication credentials contained in the request's Authorization header. Mediator authorizes the credentials against the list of consumers available in the Integration Server on which Mediator is running. This type of consumer authentication is referred to as preemptive authentication. If you want to use preemptive authentication, you should also include the action Require HTTP Basic Authentication in the policy. If you select to omit Require HTTP Basic Authentication, the client is presented with a security challenge. If the client successfully responds to the challenge, the user is authenticated. This type of consumer</td> </tr> </tbody> </table>	Value	Description	IP Address	Identifies one or more consumer applications based on their originating IP addresses.	Host Name	Identifies consumer applications based on a host name.	HTTP Authentication Token	Uses HTTP Basic authentication to verify the consumer's authentication credentials contained in the request's Authorization header. Mediator authorizes the credentials against the list of consumers available in the Integration Server on which Mediator is running. This type of consumer authentication is referred to as preemptive authentication. If you want to use preemptive authentication, you should also include the action Require HTTP Basic Authentication in the policy. If you select to omit Require HTTP Basic Authentication, the client is presented with a security challenge. If the client successfully responds to the challenge, the user is authenticated. This type of consumer
Value	Description								
IP Address	Identifies one or more consumer applications based on their originating IP addresses.								
Host Name	Identifies consumer applications based on a host name.								
HTTP Authentication Token	Uses HTTP Basic authentication to verify the consumer's authentication credentials contained in the request's Authorization header. Mediator authorizes the credentials against the list of consumers available in the Integration Server on which Mediator is running. This type of consumer authentication is referred to as preemptive authentication. If you want to use preemptive authentication, you should also include the action Require HTTP Basic Authentication in the policy. If you select to omit Require HTTP Basic Authentication, the client is presented with a security challenge. If the client successfully responds to the challenge, the user is authenticated. This type of consumer								

authentication is referred to as non-preemptive authentication.

Note:

If you select the value HTTP Authentication Token, do not include the Authorize Against Registered Consumers action in the policy. This is an invalid combination.

WS-Security Authentication Token	Validate user names and passwords that are transmitted in the SOAP message header in the WSS Username Token. If you select this value, you should also include the action Require WSS Username Token in the policy.
Custom Identification	Validates consumer applications based on an XML element (represented by an XPath expression).
Consumer Certificate	Identifies consumer applications based on information in a WSS X.509 certificate. If you select this value, you should also include the action Require WSS X.509 Token or the action Require Signing in the policy.
Client Certificate for SSL Connectivity	Validates the client's certificate that the consumer application submits to the asset in CentraSite. The client certificate that is used to identify the consumer is supplied by the client to the Mediator during the SSL handshake over the transport layer. In order to identify consumers by transport-level certificates, the run-time communication between the client and the Mediator must be over HTTPS and the client must pass a valid certificate.

To use this option, the following prerequisites must be met:

- In Integration Server, create a keystore and truststore, as described in *webMethods Integration Server Administrator's Guide*.
- In Integration Server, create an HTTPS port, as described in *webMethods Integration Server Administrator's Guide*.

- Configure Mediator by setting the IS Keystore and IS Truststore parameters, as described in *Administering webMethods Mediator*.
- Configure Mediator by setting the HTTPS Ports Configuration parameter, as described in *Administering webMethods Mediator*.

When deciding which type of identifier to use to identify a consumer application, consider the following points:

- Whatever identifier you select to identify a consumer application, it must be unique to the application. Identifiers that represent user names are often not suitable because the identified users might submit requests for multiple applications.
- Identifying applications by IP address or host name is often a suitable choice, however, it does create a dependency on the network infrastructure. If a consumer application moves to a new machine, or its IP address changes, you must update the identifiers in the application asset.
- Using X.509 certificates or a custom token that is extracted from the SOAP or XML message itself (using an XPATH expression), is often the most trouble-free way to identify a consumer application.

Log Invocation

Logs request or response payloads. You can specify the log destination and the logging frequency. This action also logs other information about the requests or responses, such as the service name, operation name, the Integration Server user, a timestamp, and the response time.

Note:

You can include this action multiple times in a policy.

Input Parameters

Log the Following Payloads *String. Optional.* Specifies whether to log all request payloads, all response payloads, or both.

Value	Description
Request	Log all request payloads.
Response	Log all response payloads.

Log Generation Frequency *String.* Specifies how frequently to log the payload.

Value	Description
Always	Log all requests and responses.

On Success Log only the successful responses and requests.

On Failure Log only the failed requests and responses.

Send Data To *String*. Specifies where to log the payload.

Important:

Ensure that Mediator is configured to log the payloads to the destination(s) you specify here. For details about alerts and transaction logging, see *Administering webMethods Mediator*.

Value	Description
CentraSite	<p>Logs the payloads in the virtual service's Events profile in CentraSite.</p> <p>Prerequisite: You must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to Solutions > Mediator > Administration > CentraSite Communication). For the procedure, see <i>Administering webMethods Mediator</i>.</p>
Local Log	<p>Logs the payloads in the server log of the Integration Server on which Mediator is running.</p> <p>Also select a value in the Log Level field:</p> <ul style="list-style-type: none"> ■ Info: Logs error-level, warning-level, and informational-level alerts. ■ Warn: Logs error-level and warning-level alerts. ■ Error: Logs only error-level alerts. <p>Important: The Integration Server Administrator's logging level for Mediator should match the logging level specified for this action (go to Settings > Logging > Server Logger).</p>
SNMP	<p>Logs the payloads in CentraSite's SNMP server or a third-party SNMP server.</p> <p>Prerequisite: You must configure the SNMP server destination (in the Integration Server Administrator, go to Solutions > Mediator > Administration > SNMP). For the procedure, see <i>Administering webMethods Mediator</i>.</p>
Email	Sends the payloads to an SMTP email server, which sends them to the email address(es) you specify

here. Mediator sends the payloads as email attachments that are compressed using gzip data compression. To specify multiple addresses, use the plus button to add rows.

Prerequisite: You must configure the SMTP server destination (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > Email**). For the procedure, see *Administering webMethods Mediator*.

Audit Log Logs the payloads in the Integration Server audit logger. For more information about logging, see the *webMethods Audit Logging Guide*.

Note:

If you expect a high volume of events in your system, it is recommended that you select the Audit Log destination for this action.

EDA/Database Logs the payloads in an EDA endpoint or Database destination that you configured in Integration Server Administrator:

- An EDA endpoint (that is, a default endpoint configured in the universal messaging configuration).
- A Database (that is, a JDBC connection pool is defined in Integration Server and associated with the Mediator functional alias).

Prerequisite: You must configure the EDA/Database destination in Integration Server on the **Solutions > Mediator > Administration > EDA/Database Configuration** page. For details, see *Administering webMethods Mediator*.

Monitor Service Performance

This action monitors a user-specified set of run-time performance conditions for a virtual service, and sends alerts to a specified destination when the performance conditions are violated. You can include this action multiple times in a single policy.

For the counter-based metrics (Total Request Count, Success Count, Fault Count), Mediator sends an alert as soon as the performance condition is violated, without having to wait until the end of the metrics tracking interval. You can select whether to send an alert only once during the interval, or every time the violation occurs during the interval. (Mediator sends another alert the next time a condition is violated during a subsequent interval.)

For the aggregated metrics (Average Response Time, Minimum Response Time, Maximum Response Time), Mediator aggregates the response times at the end of the interval, and then sends an alert if the performance condition is violated.

This action does not include metrics for failed invocations.

Note:

To enable Mediator to publish performance metrics, you must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > CentraSite Communication**). For the procedure, see *Administering webMethods Mediator*.

Input Parameters

Action Configuration parameters Specify one or more conditions to monitor. To do this, specify a metric, operator, and a value for each metric. To specify multiple conditions, use the plus button to add multiple rows. If multiple parameters are used, they are connected by the AND operator.

Name *String Array*. The metrics to monitor.

<u>Value</u>	<u>Description</u>
Availability	Indicates whether the service was available to the specified consumers in the current interval.
Average Response Time	The average amount of time it took the service to complete all invocations in the current interval. Response time is measured from the moment Mediator receives the request until the moment it returns the response to the caller. You can specify a value to monitor if the response time is within the set limit or take required steps. For example, if you specify 2 seconds, a monitoring alert will be generated if the response time exceeds 2 seconds during the current interval.
Fault Count	The number of faults returned in the current interval. You can specify a number exceeding which an alert will be generated indicate that necessary actions to taken to control the fault count. For example, if you specify 5 in this field, a monitoring alert will be generated if the fault count exceeds 5 during the current interval.
Maximum Response Time	The maximum amount of time to respond to a request in the current interval. You can specify a value to monitor if the response time does not exceed the time provided. A monitoring alert

will be generated if the maximum time taken to respond is exceeded.

Minimum Response Time	The minimum amount of time to respond to a request in the current interval.
Successful Request Count	The number of successful requests in the current interval.
Total Request Count	The total number of requests (successful and unsuccessful) in the current interval.
Operator	<i>String Array</i> . Select an appropriate operator.
Value	<i>String Array</i> . Specify an appropriate value.
Alert parameters	<i>Object</i> . Specify the following parameters for the alerts that reports on the conditions:
Alert Interval	<i>Number</i> . The time period (in minutes) in which to monitor performance before sending an alert if a condition is violated.
Alert Frequency	<i>String</i> . Specifies how frequently to issue alerts for the counter-based metrics (Total Request Count, Success Count, Fault Count).

Value	Description
Every Time	Issue an alert every time one of the specified conditions is violated.
Only Once	Issue an alert only the first time one of the specified conditions is violated.

Reply to Destination *String*. Specifies where to send the alerts.

Important:

Ensure that Mediator is configured to send event notifications to the destination(s) you specify here. For details, see *Administering webMethods Mediator*

Value	Description
CentraSite	Sends the alerts to the virtual service's Events profile in CentraSite. Prerequisite: You must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to Solutions > Mediator > Administration > CentraSite Communication). For the procedure, see <i>Administering webMethods Mediator</i> .

Local Log	<p>Sends the alerts to the server log of the Integration Server on which Mediator is running.</p> <p>Also select a value in the Log Level field:</p> <ul style="list-style-type: none"> ■ Info: Logs error-level, warning-level, and informational-level alerts. ■ Warn: Logs error-level and warning-level alerts. ■ Error: Logs only error-level alerts. <div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p>Important: The Integration Server Administrator's logging level for Mediator should match the logging level specified for this action (go to Settings > Logging > Server Logger).</p> </div>
SNMP	<p>Sends the alerts to CentraSite's SNMP server or a third-party SNMP server.</p> <p>Prerequisite: You must configure the SNMP server destination (in the Integration Server Administrator, go to Solutions > Mediator > Administration > Email). For the procedure, see <i>Administering webMethods Mediator</i>.</p>
Email	<p>Sends the alerts to an SMTP email server, which sends them to the email address(es) you specify here. To specify multiple addresses, use the plus button to add rows.</p> <p>Prerequisite: You must configure the SMTP server destination (in the Integration Server Administrator, go to Solutions > Mediator > Administration > Email). For the procedure, see <i>Administering webMethods Mediator</i>.</p>
EDA/Database	<p>Sends the alerts to an EDA endpoint/Database destination that you configured in Integration Server Administrator:</p> <ul style="list-style-type: none"> ■ An EDA endpoint (that is, a default endpoint configured in the universal messaging configuration). ■ A Database (that is, a JDBC connection pool is defined in Integration Server and associated with the Mediator functional alias).

Prerequisite: You must configure the EDA/Database destination in Integration Server on the **Solutions > Mediator > Administration > EDA/Database Configuration** page. For details, see *Administering webMethods Mediator*.

Alert Message *String. Optional.* Specify a text message to include in the alert.

Monitor Service Level Agreement

This action is similar to the Monitor Service Performance action. Both actions can monitor the same set of run-time performance conditions for a virtual service, and then send alerts when the performance conditions are violated. This action is different because it enables you to monitor run-time performance for *one or more specified consumers*. You can include this action multiple times in a single policy.

You can configure this action to define a *Service Level Agreement (SLA)*, which is a set of conditions that defines the level of performance that a consumer should expect from a service. You can use this action to identify whether a service's threshold rules are met or exceeded. For example, you might define an agreement with a particular consumer that sends an alert to the consumer if responses are not sent within a certain maximum response time. You can configure SLAs for each virtual service or consumer application combination.

For the counter-based metrics (Total Request Count, Success Count, Fault Count), Mediator sends an alert as soon as the performance condition is violated, without having to wait until the end of the metrics tracking interval. You can select whether to send an alert only once during the interval, or every time the violation occurs during the interval. (Mediator sends another alert the next time a condition is violated during a subsequent interval.)

For the aggregated metrics (Average Response Time, Minimum Response Time, Maximum Response Time), Mediator aggregates the response times at the end of the interval, and then sends an alert if the performance condition is violated.

This action does not include metrics for failed invocations.

Note:

To enable Mediator to publish performance metrics, you must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > CentraSite Communication**). For the procedure, see *Administering webMethods Mediator*.

Input Parameters

Action Configuration parameters	Specify one or more conditions to monitor. To do this, specify a metric, operator, and value for each metric. To specify multiple conditions, use the plus button to add multiple rows. If multiple parameters are used, they are connected by the AND operator.
---------------------------------	--

Name

String Array. The metrics to monitor.

Value	Description
Availability	Indicates whether the service was available to the specified consumers in the current interval.
Average Response Time	The average amount of time it took the service to complete all invocations in the current interval. Response time is measured from the moment Mediator receives the request until the moment it returns the response to the caller. You can specify a value to monitor if the response time is within the set limit or take required steps. For example, if you specify 2 seconds, a monitoring alert will be generated if the response time exceeds 2 seconds during the current interval.
Fault Count	The number of faults returned in the current interval. You can specify a number exceeding which an alert will be generated indicate that necessary actions to taken to control the fault count. For example, if you specify 5 in this field, a monitoring alert will be generated if the fault count exceeds 5 during the current interval.
Maximum Response Time	The maximum amount of time to respond to a request in the current interval. You can specify a value to monitor if the response time does not exceed the time provided. A monitoring alert will be generated if the maximum time taken to respond is exceeded.
Minimum Response Time	The minimum amount of time to respond to a request in the current interval.
Successful Request Count	The number of successful requests in the current interval.

Total Request Count	The total number of requests (successful and unsuccessful) in the current interval.						
Operator	<i>String Array</i> . Select an appropriate operator.						
Value	<i>String Array</i> Specify an appropriate value.						
Alert for Consumer Applications	<i>Object Array</i> . Specify the Application asset(s) to which this Service Level Agreement applies. To specify multiple Application assets, use the plus button to add multiple rows.						
Alert parameters	<i>Object</i> . Specify the following parameters for the alerts that reports on the Service Level Agreement conditions:						
Alert Interval	<i>Number</i> . The time period (in minutes) in which to monitor performance before sending an alert if a condition is violated.						
Alert Frequency	<i>String</i> . Specifies how frequently to issue alerts for the counter-based metrics (Total Request Count, Success Count, Fault Count).						
	<table> <thead> <tr> <th><u>Value</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>Every Time</td> <td>Issue an alert every time one of the specified conditions is violated.</td> </tr> <tr> <td>Only Once</td> <td>Issue an alert only the first time one of the specified conditions is violated.</td> </tr> </tbody> </table>	<u>Value</u>	<u>Description</u>	Every Time	Issue an alert every time one of the specified conditions is violated.	Only Once	Issue an alert only the first time one of the specified conditions is violated.
<u>Value</u>	<u>Description</u>						
Every Time	Issue an alert every time one of the specified conditions is violated.						
Only Once	Issue an alert only the first time one of the specified conditions is violated.						
Rule Expiration Date	<i>String</i> . Specifies the date on which this Service Monitoring Performance action expires, in format MM/DD/YYYY.						
Reply to Destination	<i>String</i> . Specifies where to log the alert.						

Important:

Ensure that Mediator is configured to send event notifications to the destination(s) you specify here. For details, see *Administering webMethods Mediator*.

<u>Value</u>	<u>Description</u>
CentraSite	Sends the alerts to the virtual service's Events profile in CentraSite. Prerequisite: You must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to Solutions > Mediator > Administration > CentraSite Communication). For the

Local Log	<p>procedure, see <i>Administering webMethods Mediator</i>.</p>
	<p>Sends the alerts to the server log of the Integration Server on which Mediator is running.</p>
	<p>Also select a value in the Log Level field:</p>
	<ul style="list-style-type: none"> ■ Info: Logs error-level, warning-level, and informational-level alerts. ■ Warn: Logs error-level and warning-level alerts. ■ Error: Logs only error-level alerts.
	<p>Important: The Integration Server Administrator's logging level for Mediator should match the logging level specified for this action (go to Settings > Logging > Server Logger).</p>
SNMP	<p>Sends the alerts to CentraSite's SNMP server or a third-party SNMP server.</p>
	<p>Prerequisite: You must configure the SNMP server destination (in the Integration Server Administrator, go to Solutions > Mediator > Administration > Email). For the procedure, see <i>Administering webMethods Mediator</i>.</p>
Email	<p>Sends the alerts to an SMTP email server, which sends them to the email address(es) you specify here. To specify multiple addresses, use the plus button to add rows.</p>
	<p>Prerequisite: You must configure the SMTP server destination (in the Integration Server Administrator, go to Solutions > Mediator > Administration > Email). For the procedure, see <i>Administering webMethods Mediator</i>.</p>

EDA/Database	<p>Sends the alerts to an EDA endpoint/Database destination that you configured in Integration Server Administrator:</p> <ul style="list-style-type: none"> ■ An EDA endpoint (that is, a default endpoint configured in the universal messaging configuration). ■ A Database (that is, a JDBC connection pool is defined in Integration Server and associated with the Mediator functional alias). <p>Prerequisite: You must configure the EDA/Database destination in Integration Server on the Solutions > Mediator > Administration > EDA/Database Configuration page. For details, see <i>Administering webMethods Mediator</i>.</p>
Alert Message	<p><i>String. Optional.</i> Specify a text message to include in the alert.</p>

Require Encryption

Requires that a request's XML element (which is represented by an XPath expression) be encrypted. This action supports WS-SecurityPolicy 1.2 and cannot be used with REST services.

Prerequisites

1. **Configure Integration Server:** Set up keystores and truststores in Integration Server, as described in *webMethods Integration Server Administrator's Guide*.
2. **Configure Mediator:** In the Integration Server Administrator, navigate to **Solutions > Mediator > Administration > General** and complete the IS Keystore Name, IS Truststore Name and Alias (signing) fields, as described in *Administering webMethods Mediator*.

When this policy action is set for the virtual service, Mediator provides decryption of incoming requests and encryption of outgoing responses. Mediator can encrypt and decrypt only individual elements in the SOAP message body that are defined by the XPath expressions configured for the policy action. Mediator requires that requests contain the encrypted elements that match those in the XPath expression. You must encrypt the entire element, not just the data between the element tags. Mediator rejects requests if the element name is not encrypted.

Important:

Do not encrypt the entire SOAP body because a SOAP request without an element appears to Mediator to be malformed.

Mediator attempts to encrypt the response elements that match the XPath expressions with those defined for the policy. If the response does not have any elements that match the XPath expression, Mediator does not encrypt the response before sending. If the XPath expression resolves a portion of the response message, but Mediator cannot locate a certificate to encrypt the response, then Mediator sends a SOAP fault exception to the consumer and a Policy Violation event notification to CentraSite.

The Require Encryption action encrypts the response back to the client by dynamically setting a public key alias at run time. Mediator determines the public key alias as follows:

1. If Mediator can access the X.509 certificate of the client (based on the incoming request signature), it uses useReqSigCert as the public key alias.

OR

2. If the Identify Consumer action is present in the policy (and it successfully identifies a consumer application), then Mediator looks for a public key alias with that consumer name in the IS Keystore Name property. The IS Keystore Name property is specified in the Integration Server Administrator, under **Solutions > Mediator > Administration > General**. This property should be set to an Integration Server keystore that Mediator uses.

For an Identify Consumer action that allows for anonymous usage, Mediator does *not* require a consumer name in order to send encrypted responses. In this case, Mediator can use one of the following to encrypt the response in the following order, depending on what is present in the security element:

- A signing certificate.
- Consumer name.
- WSS username, SAML token, or X.509 certificate.
- HTTP authorized user.

OR

3. If Mediator can determine the current IS user from the request (that is, if an Integration Server WS-Stack determined that Subject is present), then the first principal in that subject is used.

OR

4. If the above steps all fail, then Mediator uses either the WS-Security username token or the HTTP Basic-Auth user name value. There must be a public key entry with the same name as the identified username.

Note:

You can include this action multiple times in a single policy.

Input Parameters

Namespace *String. Optional.* Namespace of the element required to be encrypted.

Note:

Enter the namespace prefix in the following format:
`xmlns:<prefix-name>` . For example: `xmlns:soapenv`.

The generated XPath element in the policy should look similar to this:

```
<sp:SignedElements
xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-security
policy/200702">
<sp:XPath
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

//soapenv:Body</sp:XPath>
</sp:SignedElements>
```

Element Required to be Encrypted *String.* An XPath expression that represents the XML element that is required to be encrypted.

Require HTTP Basic Authentication

This action uses HTTP Basic authentication to verify the consumer's authentication credentials contained in the request's Authorization header. Mediator authorizes the credentials against the list of consumers available in the Integration Server on which Mediator is running. This type of consumer authentication is referred to as preemptive authentication. If you want to perform preemptive authentication, a policy that includes this action must also include the Identify Consumer action.

If the username or password value in the Authorization header cannot be authenticated as a valid Integration Server user (or if the Authorization header is not present in the request), a 500 SOAP fault is returned, and the client is presented with a security challenge. If the client successfully responds to the challenge, the user is authenticated. This type of consumer authentication is referred to as non-preemptive authentication. If the client does not successfully respond to the challenge, a 401 WWW-Authenticate: Basic response is returned and the invocation is not routed to the policy engine. As a result, no events are recorded for that invocation, and its key performance indicator (KPI) data are not included in the performance metrics.

If you select to omit the Require HTTP Basic Authentication action (and regardless of whether an Authorization header is present in the request or not), then:

- Mediator forwards the request to the native service, without attempting to authenticate the request.
- The native service returns a 401 WWW-Authenticate: Basic response, which Mediator forwards to the client, the client is presented with a security challenge. If the client successfully responds to the challenge, the user is authenticated.

In the case where a consumer sends a request with transport credentials (HTTP Basic authentication) and message credentials (WSS Username or WSS X.509 token), the message credentials take precedence over the transport credentials when Integration Server determines which credentials

it should use for the session. For more information, see [Require WSS User Token](#) and [Require WSS X.509 Token](#). In addition, you must ensure that the service consumer that connects to the virtual service has an Integration Server user account.

Note:

Do not include the [Require HTTP Basic Authentication](#) action in a virtual service's run-time policy if you selected the **OAuth2** option in the virtual service's Routing Protocol step.

Input Parameters**Note:**

This input parameter is not available in Mediator versions prior to 9.0.

`Authenticate Credentials` *Required.* Authorizes consumers against the list of consumers available in the Integration Server on which Mediator is running.

Require Signing

This action requires that a request's XML element (which is represented by an XPath expression) be signed. This action supports WS-SecurityPolicy 1.2.

Prerequisites

1. [Configure Integration Server](#): Set up keystores and truststores in Integration Server, as described in *webMethods Integration Server Administrator's Guide*.
2. [Configure Mediator](#): In the Integration Server Administrator, navigate to **Solutions > Mediator > Administration > General** and complete the IS Keystore Name, IS Truststore Name, and Alias (signing) fields, as described in *Administering webMethods Mediator*. Mediator uses the signing alias specified in the Alias (signing) field to sign the response.

When this action is set for the virtual service, Mediator validates that the requests are properly signed, and provides signing for responses. Mediator provides support both for signing an entire SOAP message body or individual elements of the SOAP message body.

Mediator uses a digital signature element in the security header to verify that all elements matching the XPath expression were signed. If the request contains elements that were not signed or no signature is present, then Mediator rejects the request.

Note:

Keep the following in mind:

- You must map the public certificate of the key used to sign the request to an Integration Server user. If the certificate is not mapped, Mediator returns a SOAP fault to the caller.
- You can include this action multiple times in a policy.

Input Parameters

`Namespace` *String. Optional.* Namespace of the element required to be signed.

Note:

Enter the namespace prefix in the following format:
`xmlns:<prefix-name>` . For example: `xmlns:soapenv`.

The generated XPath element in the policy should look similar to this:

```
<sp:SignedElements
xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-security
policy/200702">
<sp:XPath
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
//soapenv:Body</sp:XPath>
</sp:SignedElements>
```

Element Required to be Signed *String*. An XPath expression that represents the XML element that is required to be signed.

Require SSL

Requires that requests be sent through SSL client certificates. This action supports WS-SecurityPolicy 1.2 and can be used for both SOAP and REST services.

When this action is set for the virtual service, Mediator ensures that requests are sent to the server using the HTTPS protocol (SSL). The action also specifies whether the client certificate is required. This allows Mediator to verify the client sending the request. If the policy requires the client certificate, but it is not presented, Mediator rejects the message.

When a client certificate is required, the Integration Server HTTPS port should be configured to request or require a client certificate.

Input Parameters

Client Certificate Required *Boolean*. Specifies whether client certificates are required for the purposes of:

- Verifying the signature of signed SOAP requests or decrypting encrypted SOAP requests.
- Signing SOAP responses or encrypting SOAP responses.

Value	Description
Yes	Require client certificates.
No	Default. Do not require client certificates.

Require Timestamps**Note:**

Dependency requirement: A policy that includes this action must also include *any* one of the following actions: Require Signing, Require Encryption.

When this policy action is set for the virtual service, Mediator requires that timestamps be included in the request header. Mediator checks the timestamp value against the current time to ensure that the request is not an old message. This serves to protect your system against attempts at message tampering, such as replay attacks. This action supports WS-SecurityPolicy 1.2 and cannot be used with REST services.

Mediator rejects the request if either of the following happens:

- Mediator receives a timestamp that exceeds the time defined by the timestamp element.
- A timestamp element is not included in the request.

Input Parameters

None.

Require WSS SAML Token

When this action is set for a virtual service, Mediator uses a WSS Security Assertion Markup Language (SAML) assertion token to validate service consumers. This action supports WS-SecurityPolicy 1.2 and cannot be used with REST services.

For more information about configuring your system for SAML token processing, see *Administering webMethods Mediator*.

Input Parameters

SAML Subject Confirmation *String*. Select one of the following SAML subject confirmation methods:

Value	Description
Holder of Key	<p><i>Default</i>. Select this option if consumers use the SAML V1.1 or V2.0 Holder-of-Key Web Browser SSO Profile, which allows for transport of holder-of-key assertions. In this scenario, the consumer presents a holder-of-key SAML assertion acquired from its preferred identity provider to access a web-based resource at a service provider.</p> <p>If you select <code>Holder of Key</code>, Mediator also implicitly selects the timestamp and signing assertions to the virtual service definition (VSD). Thus, you should not add the Require Timestamps and Require Signing policy actions to a virtual service if the Require WSS SAML Token action is already applied.</p>

Bearer Select this option if consumers use SAML V1.1 Bearer token authentication, in which a Bearer token mechanism relies upon bearer semantics as a means by which the consumer conveys to Mediator the sender's identity.

If you select *Bearer*, the timestamp and signing assertions are added to the virtual service definition (VSD).

Note:

If consumers use SAML 2.0 Sender-Vouches tokens, configure your system as described in *Administering webMethods Mediator*.

SAML Version *String*. Specifies the WSS SAML Token version to use: 1.1 or 2.0.

Require WSS Username Token

Note:

Dependency requirement: A policy that includes this action must also include the Identify Consumer action.

When this policy action is set for the virtual service, Mediator uses WS-SecurityPolicy authentication to validate user names and passwords that are transmitted in the SOAP message header for the WSS Username token. This action supports WS-SecurityPolicy 1.2 and cannot be used with REST services.

In the case where a consumer is sending a request with both transport credentials (HTTP basic authentication) and message credentials (WSS Username or X.509 token), the message credentials take precedent over the transport credentials when Integration Server is determining which credentials it should use for the session.

Mediator rejects requests that do not include the username token and password of an Integration Server user. Mediator only supports clear text passwords with this kind of authentication.

Input Parameters

None.

Require WSS X.509 Token

Note:

Dependency requirement: A policy that includes this action must also include the Identify Consumer action.

Identifies consumers based on a WSS X.509 token. This action supports WS-SecurityPolicy 1.2 and cannot be used with REST services.

In the case where a consumer is sending a request with both transport credentials (HTTP Basic authentication) and message credentials (WSS X.509 token or WSS Username), the message credentials take precedence over the transport credentials when Integration Server is determining which credentials it should use for the session. In addition, you must ensure that the service consumer that connects to the virtual service has an Integration Server user account.

Input Parameters

None.

Throttling Traffic Optimization

Note:

Keep the following in mind:

- This action is not available in Mediator versions below 9.0.
- Dependency requirement: A policy that includes this action must also include the Identify Consumer action if the `Limit Traffic for Applications` option is selected.

This action limits the number of service invocations during a specified time interval, and sends alerts to a specified destination when the performance conditions are violated.

Reasons for limiting the service invocation traffic include:

- To avoid overloading the back-end services and their infrastructure.
- To limit specific consumers in terms of resource usage (that is, you can use the Monitor Service Level Agreement action to monitor performance conditions for a particular consumer, together with Throttling Traffic Optimization to limit the resource usage).
- To shield vulnerable servers, services, and even specific operations.
- For service consumption metering (billable pay-per-use services).

Note:

To enable Mediator to publish performance metrics, you must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > CentraSite Communication**). For the procedure, see *Administering webMethods Mediator*.

Input Parameters

`Soft Limit` *Number. Optional.* Specifies the maximum number of invocations allowed per `Interval` before issuing an alert. Reaching the soft limit does not affect further processing of requests (until the `Hard Limit` is reached).

Note:

The limit is reached when the total number of invocations coming from *all* the consumer applications (specified in the `Limit Traffic for Applications` field) reaches the limit. `Soft Limit` is computed in

an asynchronous manner; thus when multiple requests are made at the same time, it may be possible that the Soft Limit alert does not be strictly accurate.

Hard Limit *Number. Required.* Specifies the maximum number of invocations allowed per alert interval before stopping the processing of further requests and issuing an alert. Typically, this number should be higher than the soft limit.

Note:

The limit is reached when the total number of invocations coming from *all* the consumer applications (specified in the `Limit Traffic for Applications` field) reaches the limit. Hard Limit is computed in an asynchronous manner; thus when multiple requests are made at the same time, it may be possible that the Hard Limit alert does not be strictly accurate.

Limit Traffic for Applications *String.* Specifies the consumer application(s) that this action applies to. To specify multiple consumer applications, use the plus button to add rows, or select **Any Consumer** to apply this action to any consumer application.

Interval *Number.* Specifies the amount of time for the soft limit and hard limit to be reached.

Frequency *String.* Specifies how frequently to issue alerts.

Value	Description
Every Time	Issue an alert every time the specified condition is violated.
Only Once	Issue an alert only the first time the specified condition is violated.

Reply To Destination *String. Optional.* Specifies where to log the alerts.

Important:

Ensure that Mediator is configured to send event notifications to the destination(s) you specify here. For details, see *Administering webMethods Mediator*.

Value	Description
CentraSite	Sends the alerts to the virtual service's Events profile in CentraSite. Prerequisite: You must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to Solutions > Mediator > Administration >

	<p>CentraSite Communication). For the procedure, see <i>Administering webMethods Mediator</i>.</p>
Local Log	<p>Sends the alerts to the server log of the Integration Server on which Mediator is running.</p> <p>Also select a value in the <code>Log Level</code> field:</p> <ul style="list-style-type: none"> ■ Info: Logs error-level, warning-level, and informational-level alerts. ■ Warn: Logs error-level and warning-level alerts. ■ Error: Logs only error-level alerts. <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Important: The Integration Server Administrator's logging level for Mediator should match the logging level specified for this action (go to Settings > Logging > Server Logger).</p> </div>
SNMP	<p>Sends the alerts to CentraSite's SNMP server or a third-party SNMP server.</p> <p>Prerequisite: You must configure the SNMP server destination (in the Integration Server Administrator, go to Solutions > Mediator > Administration > Email). For the procedure, see <i>Administering webMethods Mediator</i>.</p>
Email	<p>Sends the alerts to an SMTP email server, which sends them to the email address(es) you specify here. To specify multiple addresses, use the plus button to add rows.</p> <p>Prerequisite: You must configure the SMTP server destination (in the Integration Server Administrator, go to Solutions > Mediator > Administration > Email). For the procedure, see <i>Administering webMethods Mediator</i>.</p>
EDA/Database	<p>Sends the alerts to an EDA endpoint/Database destination that you configured in Integration Server Administrator:</p>

- An EDA endpoint (that is, a default endpoint configured in the universal messaging configuration).
- A Database (that is, a JDBC connection pool is defined in Integration Server and associated with the Mediator functional alias).

Prerequisite: You must configure the EDA/Database destination in Integration Server on the **Solutions > Mediator > Administration > EDA/Database Configuration** page. For details, see *Administering webMethods Mediator*.

Alert Message for *String*. *Optional*. Specify a text message to include in the soft limit alert.
Soft Limit

Alert Message for *String*. *Optional*. Specify a text message to include in the hard limit alert.
Hard Limit

Validate Schema

This action validates all XML request and response messages against an XML schema referenced in the WSDL.

Mediator can enforce this policy action for messages sent between services. When this policy is set for the virtual service, Mediator validates XML request messages, response messages, or both, against the XML schema referenced in the WSDL.

Input Parameters

Validate SOAP Message(s) *Object*. Validates request and response messages. You may select both Request and Response.

Value	Description
Request	Validate all requests.
Response	Validate all responses.

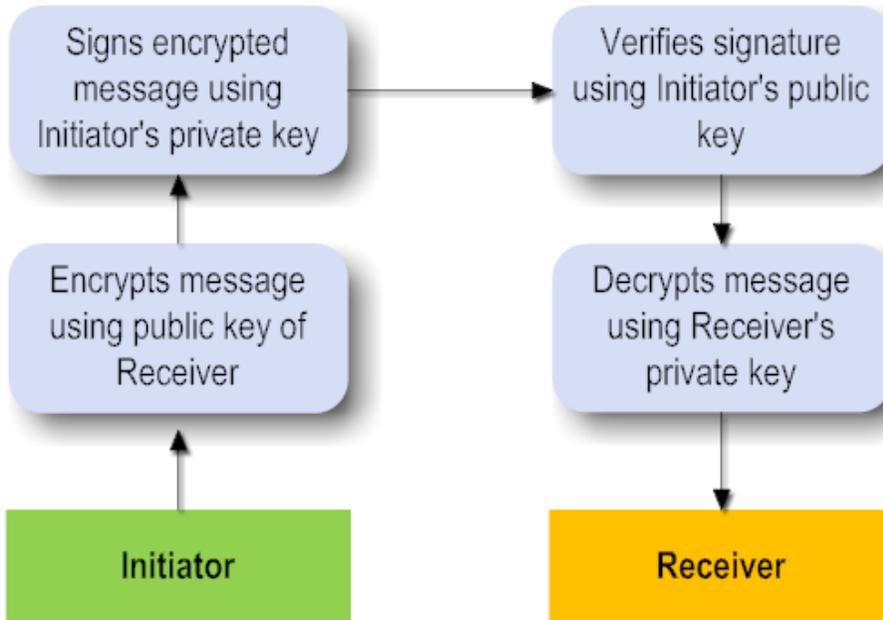
Important:

Be aware that Mediator does not remove `wsu:Id` attributes that may have been added to a request by a consumer as a result of security operations against request elements (that is, signatures and encryptions). In this case, to avoid schema validation failures you would have to add a Request Handling step to the virtual service so that the requests are passed to an XSL transformation file that removes the `wsu:Id` attribute.

Asymmetric Binding Configuration

WS-SecurityPolicy specification deals with three types of Security Bindings. A security binding determines how the message transfer is to be done between the recipient and the initiator.

Asymmetric Binding is used when both the Initiator and the Recipient possess public and private keys. The message transfer takes place using Public Key Infrastructure.



An Asymmetric Binding element in the WSDL looks like this:

```

<sp:AsymmetricBinding
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
  <wsp:Policy>
    <sp:InitiatorToken>
    </sp:InitiatorToken>
    <sp:RecipientToken>
    </sp:RecipientToken>
    <sp:AlgorithmSuite>
      <wsp:Policy>
        <sp:TripleDesRsa15/>
      </wsp:Policy>
    </sp:AlgorithmSuite>
    <sp:Layout>
      <wsp:Policy>
        <sp:Strict/>
      </wsp:Policy>
    </sp:Layout>
    <sp:IncludeTimestamp/>
    <sp:OnlySignEntireHeadersAndBody/>
  </wsp:Policy>
</sp:AsymmetricBinding>
  
```

The following run-time actions that support WS-Security policies use a common Asymmetric Binding element:

- The Require Encryption action.
- The Require Signing action.
- The Require WSS SAML Token action.
- The Require WSS X.509 Token action.

The Asymmetric Binding Components

The components of a security binding are:

Recipient Token Inclusion

The value of Recipient Token Inclusion specifies how to include the Recipient token during message exchange from Initiator to Recipient or Recipient to Initiator. It takes the same values as Initiator Token Inclusion above.

Algorithm Suite

The value of Algorithm Suite specifies the algorithm suite to be used for this asymmetric binding. The possible algorithms supported are:

- Basic256
- Basic192
- Basic128
- TripleDes
- Basic256Rsa15
- Basic192Rsa15
- Basic128Rsa15
- TripleDesRsa15
- Basic256Sha256
- Basic192Sha256
- Basic128Sha256
- TripleDesSha256
- Basic256Sha256Rsa15
- Basic192Sha256Rsa15
- Basic128Sha256Rsa15

Layout

Layout describes the way information is added to the message header. The possible values are:

Value	Description
Strict	Items are added to the security header in a principle of declare before use.
Lax	Items are added to the security header in any order that conforms to WSS: SOAP Message Security.
LaxTsFirst	Same as Lax except that the first item in the security header <i>must</i> be a <code>wsse:Timestamp</code> . The <code>wsse:Timestamp</code> property <i>must</i> also be set to true in this case.
LaxTsLast	Same as Lax except that the last item in the security header <i>must</i> be a <code>wsse:Timestamp</code> . The <code>wsse:Timestamp</code> property <i>must</i> also be set to true in this case.

Obtaining Details of Asymmetric Binding

Pre-requisites:

To obtain the current asymmetric binding configuration values through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `get Asymmetric Binding` for this purpose.

> To obtain the current Asymmetric binding configuration

- Run the command `get Asymmetric Binding`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get Asymmetric Binding [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password of the user identified by the parameter <code>USER-ID</code> .

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get Asymmetric Binding -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
```

The configuration is maintained in the form of an XML file which is loaded with default values.

```
<?xml version="1.0" encoding="UTF-8" ?>
<AsymmetricBindingConfiguration>
<initiatorTokenInclusion>AlwaysToRecipient</initiatorTokenInclusion>
<recipientTokenInclusion>Never</recipientTokenInclusion>
<algorithmSuite>TripleDesRsa15</algorithmSuite>
<layout>Strict</layout>
</AsymmetricBindingConfiguration>
```

Modifying Details of Asymmetric Binding

Pre-requisites:

To change the current asymmetric binding configuration values through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `set Asymmetric Binding` for this purpose.

> To modify details of the asymmetric binding configuration

- Run the command `set Asymmetric Binding`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Asymmetric Binding [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> [-initiatorTokenInclusion <INITIATOR-TOKEN-INCLUSION>] [-recieipientTokenInclusion <RECIPIENT-TOKEN-INCLUSION>] [-algorithmSuite <ALGORITHM-SUITE>] [-layout <LAYOUT>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, Administrator.
PASSWORD	The password of the user identified by the parameter USER-ID.
INITIATOR-TOKEN-INCLUSION	Inclusion value for the Initiator Token. *
RECIPIENT-TOKEN-INCLUSION	Inclusion value for the Recipient Token. *
ALGORITHM-SUITE	The algorithm to be used. *
LAYOUT	The layout to be used. *

* At least one of the following parameters is required: `initiatorTokenInclusion`, `recipientTokenInclusion`, `algorithmSuite` and `layout`.

The configuration is maintained in the form of an XML file which is loaded with default values.

```
<?xml version="1.0" encoding="UTF-8" ?>
<AsymmetricBindingConfiguration>
<initiatorTokenInclusion>AlwaysToRecipient</initiatorTokenInclusion>
<recipientTokenInclusion>Never</recipientTokenInclusion>
<algorithmSuite>TripleDesRsa15</algorithmSuite>
<layout>Strict</layout>
</AsymmetricBindingConfiguration>
```

Removing Asymmetric Binding

Pre-requisites:

To remove the current asymmetric binding configuration through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `remove Asymmetric Binding` for this purpose.

> To remove an asymmetric binding configuration

- Run the command `remove Asymmetric Binding`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd remove Asymmetric Binding [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password of the user identified by the parameter <code>USER-ID</code> .

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd remove Asymmetric Binding
-url http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
```

Gateway Management

You can create, update, list, and delete gateways in CentraSite in one of the following ways:

- CentraSite Business UI
- CentraSite Command Line Interface

Note:

The target model in CentraSite is deprecated starting 9.8 release. As a result:

- The **Add Target** action is deprecated in CentraSite Control.
- You cannot edit the target details in CentraSite Control. You can only view details such as: Name, Description, and Target Type of the existing Mediator targets.
- However, you can still deploy, undeploy, and redeploy Virtual APIs to Mediator gateway using CentraSite Control.

Targets created in the previous versions are migrated to Mediator gateways when you migrate the data. For details, see *Upgrading webMethods and Intelligent Business Operations Products* guide.

To use an instance of CentraSite with webMethods API Gateway, webMethods API Portal, webMethods Mediator, or webMethods Insight you must define the gateway that identifies an instance of the asset type API Gateway, API Portal, Mediator, or Insight Server, and other policy enforcement points you want to use. The gateway instance specifies the address of the deployment endpoint that CentraSite uses to interact with API Gateway, API Portal, Mediator, or Insight Server to deploy the Virtual Service (API) assets.

Before You Configure CentraSite for API Gateway

Before you start configuring CentraSite for API Gateway, ensure that the following products are installed:

- webMethods API Gateway
- webMethods Integration Server (Installing API Gateway implicitly installs Integration Server).
- webMethods API Portal

For more information about installing these products, see *Installing Software AG Products*.

Before You Configure CentraSite for API Portal

Before you start configuring CentraSite for API Portal, ensure that the following products are installed:

- webMethods API Portal
- webMethods API Gateway or webMethods Mediator (Installing API Gateway or Mediator implicitly installs Integration Server).
- webMethods Integration Server

For more information about installing these products, see *Installing Software AG Products*.

Before You Configure CentraSite for Mediator

Before you start configuring CentraSite for Mediator, ensure that the following products are installed:

- webMethods Mediator

- webMethods Integration Server (Installing Mediator implicitly installs Integration Server).

For more information about installing these products, see *Installing Software AG Products*.

Before You Configure CentraSite for Insight Server

Before you start configuring CentraSite for Insight Server, ensure that the product webMethods Insight is installed.

For more information about installing Insight Server, see *Installing Software AG Products*.

Managing Gateways through CentraSite Business UI

This section describes operations you can perform to manage gateways through CentraSite Business UI.

Note:

You cannot manage run-time aspects using Mediator gateway if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

Creating API Gateway Asset

Pre-requisites:

To create and manage (that is, view, modify, and delete) API Gateway assets for an organization, you must have one of the following roles:

- CentraSite Administrator: instances of API Gateway in any organization in CentraSite.
- Organization Administrator: instances of API Gateway in your organization.
- API Gateway Administrator: instances of API Gateway in the specific organization to which your API Gateway Administrator role applies.

To establish communication between CentraSite and an API Gateway instance, you must first capture the configuration details of the API Gateway instance with which you want to communicate, and then publish the CentraSite information to API Gateway.

You specify the configuration details of an API Gateway using the **Add Gateway** action in the **Governance Rules** activity. If you do not see the **Add Gateway** action, it is probably because you do not have the required role to configure and register an API Gateway asset in CentraSite.

> To create an API Gateway asset

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. On the actions bar of the Search Results page, click **Add Gateway**.

3. In the Create New Gateway page, provide the required information for each of the displayed data fields:

Field	Description
Name	<p>Name of the API Gateway asset.</p> <p>An API Gateway name can contain any character (including spaces).</p> <p>The API Gateway name must be unique within the registry. The API Gateway name cannot be the same as any existing API Gateway name.</p>
Description	<p>(Optional). Description of the API Gateway. This description appears when a user displays the list of API Gateway instances in CentraSite Business UI.</p>
Gateway	<p>Select API Gateway.</p>
Organization	<p>Name of an organization where you want to register this API Gateway asset. This value is set to Default Organization.</p> <p>The drop-down list contains the list of organizations to which you are permitted to register the API Gateway asset.</p>

4. In the CentraSite Communication Information (API Gateway to CentraSite) section, provide the following information:

Field	Description
Username	<p>The CentraSite user ID for authenticating against CentraSite when API Gateway communicates with CentraSite.</p> <p>This implies the user ID of a user who has the CentraSite Administrator role or the API Gateway Administrator role.</p> <div style="background-color: #f0f0f0; padding: 10px;"> <p>Note: This user should have at least Modify permission to update details of the APIs published to this API Gateway. This user could also be part of the <i>MyAPIGateway</i> Synchronization Group created for this API Gateway. Users in the <i>MyAPIGateway</i> Synchronization Group will always have Modify permission on the API once it is published to API Gateway.</p> </div>

Field	Description
Password	The password of the CentraSite user specified in the Username field.

Note:

The **CentraSite Endpoint** field shows the URL (scheme, host, and port) of the CentraSite Application Server Tier (CAST) in the format, <scheme>://<host>:<port>. The scheme is http or https. The host is the machine on which CAST is running, and port is the port on which CentraSite is listening. The value for the **CentraSite Endpoint** field is determined by the URL that you use to access the CentraSite Business UI.

The CentraSite Communication Information is used to send run-time events, performance metrics, and other information from API Gateway to CentraSite. This information is updated in the CentraSite Communication section under **CentraSite Destination** in API Gateway.

- In the API Gateway Communication Information (CentraSite to API Gateway) section, provide the following information:

Field	Description
API Gateway Endpoint	<p>The API Gateway's deployment endpoint, which is the endpoint that CentraSite uses to interact with API Gateway for deployment of Virtual Service assets.</p> <p>The API Gateway Endpoint URL has the following format:</p> <p>http://<host>:<port></p> <p>Example: http://myHostname:5555</p>
API Gateway WebApp URL	<p>(Optional). The Web application URL of API Gateway. For example, http://myHostname:9072.</p>

Note:

Various aspects of the way **API Gateway WebApp URL** functions are as follows:

- If a value is specified for this attribute, then CentraSite uses this URL as the API Gateway Web application URL.
- If a value is not specified for this attribute, then CentraSite automatically populates the dynamic URL based on the value that was sent by API Gateway on publishing the API Gateway asset from CentraSite to the API Gateway instance.
- API Gateway sends the Web application URL based on the value specified in the **Web application load balancer URL** field in API Gateway (go to <Username> > **Administration** > **General** > **Load**

Field	Description
	<p>balancer) as described in <i>webMethods API Gateway Administrator's Guide</i>. For example,</p> <ul style="list-style-type: none"> ■ if a value is specified for this field, API Gateway sends the defined Web application load balancer URL to CentraSite. ■ if the value is not specified for this field, API Gateway sends the default hostname and port number as Web application URL to CentraSite. For example: <code>http://myHostname:9072</code>
Use CentraSite Credentials	<p>Selecting the check box enables reuse of the CentraSite credentials for authenticating against API Gateway.</p> <p>When you select the Use CentraSite Credentials check box, the subsequent Username and Password fields are automatically disabled.</p>
Username	<p>The Integration Server user who is permitted to publish assets to API Gateway. By default, users of the Integration Server's API Gateway Administrator group are permitted to publish assets to this gateway.</p>
Password	<p>The password for the Integration Server user specified in the Username box.</p>
Sandbox	<p>(Optional). The sandbox category that is to be used to classify this API Gateway.</p> <ol style="list-style-type: none"> a. Click Choose. The Sandbox dialog box displays the available sandbox categories. b. Select the checkbox next to the name of the sandbox category you want to use to classify the API Gateway's URL. c. Click OK. <p>CentraSite includes a set of predefined categories for the taxonomy node Sandbox classifying API Gateway.</p> <p>The available sandbox categories are:</p> <ul style="list-style-type: none"> ■ Development ■ Production ■ Test <p>For information on the Sandbox categories that CentraSite supports out-of-the-box, in CentraSite Control, go to</p>

Field	Description
	<p>Administration > Taxonomies. In the Taxonomies page, navigate to Sandbox in the list of taxonomies.</p> <p>If you want to use sandbox categories that are not supported by CentraSite, you can define your custom categories.</p>

Note:

Although it is possible to define subcategories for the predefined and custom categories within the **Sandbox** taxonomy, you cannot use these subcategories to classify the URL. CentraSite only displays the names of the top-level categories (that is, categories that are defined for the **Sandbox** taxonomy) for the classification.

6. Click **Publish**.

An API Gateway asset instance is created in the specified organization, and registered with the CentraSite Registry Repository. The details page for the API Gateway asset that you just created is displayed. For each API Gateway asset that is successfully created, CentraSite creates a *MyAPIGateway* Synchronization Group, where *MyAPIGateway* is the name of the *MyAPIGateway* asset. You can then add users with access to API Gateway to this Synchronization Group to send metrics for all services. This group gets permissions for all services published to this API Gateway.

Note:

When trying to create and publish an API Gateway asset to the API Gateway instance, if the API Gateway instance is already registered with another instance of API Gateway asset, publish of CentraSite communication and SNMP configuration to this API Gateway instance fails. However, the API Gateway asset is successfully created in CentraSite.

Creating API Portal Gateway Asset**Pre-requisites:**

To create and manage (that is, view, modify, and delete) API Portal assets for an organization, you must have one of the following roles:

- CentraSite Administrator: instances of API Portal gateway in any organization in CentraSite.
- Organization Administrator: instances of API Portal gateway in your organization.
- API Portal Administrator: instances of API Portal gateway in the specific organization to which your API Portal Administrator role applies.

To establish communication between CentraSite and an API Portal gateway instance, you must first capture the configuration details of the API Portal instance with which you want to communicate, and then publish the CentraSite information to API Portal.

You specify the configuration details of an API Portal gateway using the **Add Gateway** action in the **Governance Rules** activity. If you do not see the **Add Gateway** action, it is probably because you do not have the required role to configure and register an API Portal gateway asset in CentraSite.

➤ **To create an API Portal gateway asset**

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. On the actions bar of the Search Results page, click **Add Gateway**.
3. In the Create New Gateway page, provide the required information for each of the displayed data fields:

Field	Description
Name	<p>Name of the API Portal asset.</p> <p>An API Portal name can contain any character (including spaces).</p> <p>The API Portal name does not need to be unique within the registry. However, to reduce ambiguity, you must avoid giving multiple API Portal instances the same name.</p> <p>As a best practice, consider adopting appropriate naming conventions to ensure that API Portal instances are distinctly named within an organization.</p>
Description	(Optional). Description of the API Portal gateway. This description appears when a user displays the list of API Portal instances in CentraSite Business UI.
Gateway	Select API Portal .
Organization	<p>Name of an organization where you want to register this API Portal gateway. This value is set to Default Organization.</p> <p>The drop-down list contains the list of organizations to which you are permitted to register the API Portal gateway.</p>
Onboarding Consumer Organization	<p>Name of an organization where you want to add the users of API Portal gateway. This value is set to Default Organization.</p> <ul style="list-style-type: none"> ■ Use Existing (default): Select an existing organization.

Field	Description
	<ul style="list-style-type: none"> ■ Create New: Type a new organization name. <p>Note: CentraSite associates the API Portal user with the selected organization during user onboarding process.</p>

4. In the CentraSite Communication Information (API Portal to CentraSite) section, provide the following information:

Field	Description
Username	<p>The CentraSite user ID for authenticating against CentraSite when API Portal communicates with CentraSite.</p> <p>This implies the user ID of a user who has the CentraSite Administrator role or the API Portal Administrator role.</p>
Password	<p>The password of the CentraSite user specified in the Username field.</p>

Note:

The **CentraSite Endpoint** field shows the URL (scheme, host, and port) of the CentraSite Application Server Tier (CAST) in the format, <scheme>://<host>:<port>. The scheme is http or https. The host is the machine on which CAST is running, and port is the port on which CentraSite is listening. The value for the **CentraSite Endpoint** field is determined by the URL that you use to access the CentraSite Business UI.

The CentraSite Communication Information is used to send run-time events, performance metrics, and other information from API Portal to CentraSite.

5. In the API Portal Communication Information (CentraSite to API Portal) section, provide the following information:

Field	Description
API Portal Endpoint	<p>The URL (host and port) of API Portal instance.</p> <p>The API Portal Endpoint URL has the following format:</p> <p>http://<host>:<port>/<WebAppContext></p> <p>Example: http://myServer:18101/abs</p>
Tenant	<p>The name of a tenant residing in API Portal.</p> <p>By default, CentraSite populates this field with the default tenant shipped with an instance of API Portal.</p>

Field	Description
Use CentraSite Credentials	<p>Selecting the check box enables reuse of the CentraSite credentials for authenticating against API Portal.</p> <p>When you select the Use CentraSite Credentials check box, the subsequent Username and Password fields are automatically disabled.</p>
Username	<p>The API Portal user ID as configured in the UMC.</p> <p>This implies the user ID of a user who has the API Provider role in API Portal.</p>
Password	<p>The password of the API Portal user specified in the Username field.</p>
Sandbox	<p>(Optional). The sandbox category that is to be used to classify this API Portal gateway.</p> <ol style="list-style-type: none"> Click Choose. The Sandbox dialog box displays the available sandbox categories. Select the checkbox next to the name of the sandbox category you want to use to classify the API Portal's URL. Click OK. <p>CentraSite includes a set of predefined categories for the taxonomy node Sandbox classifying API Portal.</p> <p>The available sandbox categories are:</p> <ul style="list-style-type: none"> ■ Development ■ Production ■ Test <p>For information on the Sandbox categories that CentraSite supports out-of-the-box, in CentraSite Control, go to Administration > Taxonomies. In the Taxonomies page, navigate to Sandbox in the list of taxonomies.</p> <p>If you want to use sandbox categories that are not supported by CentraSite, you can define your custom categories.</p> <div style="background-color: #f0f0f0; padding: 10px;"> <p>Note: Although it is possible to define subcategories for the predefined and custom categories within the Sandbox taxonomy, you cannot use these subcategories to classify</p> </div>

Field	Description
	the URL. CentraSite only displays the names of the top-level categories (that is, categories that are defined for the Sandbox taxonomy) for the classification.

6. Click **Publish**.

An API Portal Gateway asset instance is created in the specified organization, and registered with the CentraSite Registry Repository. The details page for the API Portal Gateway asset that you just created is displayed.

Creating Mediator or Insight Server Gateway Asset

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

Pre-requisites:

To create and manage (that is, view, modify, and delete) Mediator and Insight Server assets for an organization, you must have one of the following roles:

- CentraSite Administrator: instances of Mediator and Insight Server gateways in any organization in CentraSite.
- Organization Administrator: instances of Mediator and Insight Server gateways in your organization.
- Mediator Administrator: instances of Mediator and Insight Server gateways in the specific organization to which your Mediator Administrator role applies.

To establish communication between CentraSite and a Mediator or Insight Server gateway instance, you must first capture configuration details of the Mediator or Insight Server instance with which you want to communicate, and then publish the CentraSite information to Mediator or Insight Server.

You specify the configuration details of a Mediator or Insight Server gateway using the **Add Gateway** action in the **Governance Rules** activity. If you do not see the **Add Gateway** action, it is probably because you do not have the required role to configure and register a Mediator or Insight Server gateway asset in CentraSite.

➤ To create a Mediator or Insight Server gateway

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. On the actions bar of the Search Results page, click **Add Gateway**.

3. In the Create New Gateway page, provide the required information for each of the displayed data fields:

Field	Description
Name	<p>Name of the Mediator or Insight Server asset.</p> <p>A Mediator or Insight Server gateway can contain any character (including spaces).</p> <p>The Mediator or Insight Server name does not need to be unique within the registry. However, to reduce ambiguity, you must avoid giving multiple Mediator or Insight Server instances the same name.</p> <p>As a best practice, consider adopting appropriate naming conventions to ensure that Mediator or Insight Server instances are distinctly named within an organization.</p>
Description	<p>(Optional). Description of the Mediator or Insight Server gateway. This description appears when a user displays the list of Mediator or Insight Server instances in CentraSite Business UI.</p>
Gateway	<p>Select Mediator or Insight.</p>
Organization	<p>Name of an organization where you want to register this Mediator or Insight Server gateway. This value is set to Default Organization.</p> <p>The drop-down list contains the list of organizations to which you are permitted to register the Mediator or Insight Server gateway.</p>

4. For Mediator gateway only. In the CentraSite Communication Information (Mediator to CentraSite) section, provide the following information:

Field	Description
Username	<p>The CentraSite user ID for authenticating against CentraSite when Mediator communicates with CentraSite.</p> <p>This implies the user ID of a user who has the CentraSite Administrator role or the Mediator Administrator role.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: This user should have at least Modify permission to update details of the APIs published to this Mediator. This user could also be part of the <i>MyMediatorGateway</i> Synchronization Group created for this API Gateway.</p> </div>

Field	Description
	Users in the <i>MyMediatorGateway</i> Synchronization Group will always have Modify permission on the API once it is published to Mediator.
Password	The password of the CentraSite user specified in the Username field.

Note:

The **CentraSite Endpoint** field shows the URL (scheme, host, and port) of the CentraSite Application Server Tier (CAST) in the format, <scheme>://<host>:<port>. The scheme is http or https. The host is the machine on which CAST is running, and port is the port on which CentraSite is listening. The value for the **CentraSite Endpoint** field is determined by the URL that you use to access the CentraSite Business UI.

The CentraSite Communication Information is used to send run-time events, performance metrics, and other information from Mediator to CentraSite. This information is updated in the **CentraSite Communication** page under **Mediator Administration** in Integration Server.

- For Mediator gateway only. In the Mediator Communication Information (CentraSite to Mediator) section, provide the following information:

Field	Description
Mediator Endpoint	<p>The Mediator's deployment endpoint, which is the endpoint that CentraSite uses to interact with Mediator for deployment of Virtual Service assets.</p> <p>The Mediator Endpoint URL has the following format:</p> <p>http://<host>:<port>.</p> <p>Example: http://myHostname:5555</p>
Use CentraSite Credentials	<p>Selecting the check box enables reuse of the CentraSite credentials for authenticating against Mediator.</p> <p>When you select the Use CentraSite Credentials check box, the subsequent Username and Password fields are automatically disabled.</p>
Username	The Integration Server user who is permitted to deploy assets to Mediator gateway. By default, only a few of the Integration Server's Administrator group are permitted to deploy assets to this gateway.

Note:

This note explains how to permit other users to deploy assets to this target. Mediator exposes several web

Field	Description
	<p>service operations to allow CentraSite to manage deployed assets. This web service is invoked by CentraSite any time a user deploys or undeploys a virtual service or consumer application to Mediator. The Username and Password fields identify an Integration Server user who is permitted to execute the Integration Server services associated with Mediator's deployer service. After installation, only members of the Integration Server's Administrator group are permitted to invoke these services. However, administrators have the flexibility to allow their own users or groups to invoke them. Access to these services is controlled by an ACL, called <code>MediatorDeployer</code>. Initially, only the predefined Administrator group is assigned to this ACL. An Integration Server administrator can remove this group and add other groups or individual users. For example, you can create your own deployer group, (for example, <code>MyDeployers</code>) and add Integration Server user IDs to this group. Then, the user must update the <code>MediatorDeployer</code> ACL by removing the Administrator group and adding the <code>MyDeployers</code> group. Now, in the Username and Password fields on this screen, you can specify any user ID that belongs to the <code>MyDeployers</code> group.</p>
Password	The password for the Integration Server user specified in the Username box.
Sandbox	<p>(Optional). The sandbox category that is to be used to classify this Mediator gateway.</p> <ol style="list-style-type: none"><li data-bbox="584 1302 1338 1491">Click Choose. <p>The Sandbox List is displayed with the predefined sandbox categories: Development, Production, and Test.</p><li data-bbox="584 1491 1338 1533">Select a sandbox category from the list.<li data-bbox="584 1533 1338 1873">Click OK. <p>For information on the Sandbox categories that CentraSite supports out-of-the-box, in CentraSite Control, go to Administration > Taxonomies. In the Taxonomies page, navigate to Sandbox in the list of taxonomies. If you want to use sandbox categories that are not supported by CentraSite, you can define custom categories.</p>

6. Click **Publish**.

A Mediator or Insight Gateway asset instance is created in the specified organization, and registered with the CentraSite Registry Repository. The details page for the Mediator or Insight Gateway asset that you just created is displayed. For each Mediator gateway asset that is successfully created, CentraSite creates a *MyMediatorGateway* Synchronization Group, where *MyMediatorGateway* is the name of the Mediator gateway. You can add users who have access to the Mediator gateway to this Synchronization Group to send metrics for all services. This group gets permissions for all services published to this gateway.

Viewing the Gateway List and Gateway Details

Pre-requisites:

To view a list of available gateways for an organization, you must have one of the following roles:

- CentraSite Administrator: instances of API Gateway, API Portal, Mediator, and Insight Server gateways in any organization in CentraSite.
- Organization Administrator: instances of API Gateway, API Portal, Mediator, and Insight Server gateways in your organization.
- API Gateway Administrator: instances of API Gateway in the specific organization to which your API Gateway Administrator role applies.
- API Portal Administrator: instances of API Portal gateway in the specific organization to which your API Portal Administrator role applies.
- Mediator Administrator: instances of Mediator and Insight Server gateways in the specific organization to which your Mediator Administrator role applies.

In addition to the above, you can also view the details of a gateway instance if you have the Full, Modify, or at least View instance-level permission on the gateway itself.

You view the list of gateway assets using the **Governance Rules** activity in CentraSite Business UI. In addition, you can view gateway details, modify gateway details, publish (register) and unpublish (unregister) gateways, view gateway asset dependencies, and monitor runtime performance metrics in the Gateway Details page.

➤ To view the list of gateways and gateway details

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. To filter the list to display the available API Gateway, API Portal, Mediator, and Insight Server gateways, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, do the following:

- a. Locate **Applicable Scopes**.
- b. Select the required type of gateway asset from the drop-down list:
 - **API Gateway**
 - **API Portal**
 - **Mediator**
 - **Insight**
- c. Click the plus button next to the drop-down box or press Enter to add the scope to the search recipe.

CentraSite displays the list of available API Gateway, API Portal, Mediator, and Insight Server gateways.

The Search Results page provides the following basic information for each gateway:

Column	Description
Name	The name of the gateway.
Description	A descriptive information about the gateway.
Created Date	The date on which the gateway was created in the CentraSite registry. CentraSite automatically sets this attribute when an administrator creates and registers the gateway with CentraSite. Note that once a gateway is created, the value of this field cannot be modified.
Owner	The user to which the gateway belongs.
Organization	The organization to which the gateway belongs.
Version	The system-assigned version identifier for the gateway.

Select an attribute from the **Sort by** box to sort API Gateway assets by name, description, and date last modified in the Search Results page.

Use the **View** menu to selectively display the **Sort by** attributes by specifying them individually.

The Search Results page displays one or more actions that you can perform in CentraSite:

Action	Description
Add Gateway	Create a new API Gateway asset in CentraSite.
Publish	Register an API Gateway asset with CentraSite.
Unpublish	Unregister an API Gateway asset from CentraSite.
Delete	Remove an existing API Gateway asset from CentraSite.

3. In the displayed list, select the gateway whose details you want to examine.

This opens the Gateway Details page. You can also perform the following operations in the Gateway Details page:

Action	Description
Edit	Modify the details of API Gateway asset.
Save	Update the modified details of API Gateway asset.
Delete	Remove the API Gateway asset from CentraSite.
Add to List	Add the API Gateway asset to your favorites list.
View Report	Generate reports that contain the specific information for API Gateway asset.
Asset Navigator	Visualize the dependencies to and from the API Gateway asset.
Permissions	Set instance-level permissions on the API Gateway asset.
Publish	Register an API Gateway asset with CentraSite.
Unpublish	Unregister an API Gateway asset from CentraSite.

Modifying Gateway Asset Details

Pre-requisites:

To modify the information of gateways for an organization, you must have one of the following roles:

- CentraSite Administrator: instances of API Gateway, API Portal, Mediator, and Insight Server gateways in any organization in CentraSite.
- Organization Administrator: instances of API Gateway, API Portal, Mediator, and Insight Server gateways in your organization.
- API Gateway Administrator: instances of API Gateway in the specific organization to which your API Gateway Administrator role applies.
- API Portal Administrator: instances of API Portal gateway in the specific organization to which your API Portal Administrator role applies.
- Mediator Administrator: instances of Mediator and Insight Server gateways in the specific organization to which your Administrator role applies.

In addition to the above, you can also view the details of a gateway instance if you have the Full or at least Modify instance-level permission on the gateway itself.

You modify the existing information of a gateway asset using the **Edit** action in the Gateway Details page.

Note:

If you do not see the **Edit** action in the Gateway Details page, it is probably because you do not have the required role or permission to modify the details of a gateway in CentraSite.

When you modify the details of a gateway, keep the following points in mind:

- To modify the details of a gateway in CentraSite, you must unpublish (unregister) the gateway from CentraSite, make the required changes, and then republish (register) the gateway to CentraSite.
- CentraSite does not allow you to directly modify the endpoint URL of a gateway, and tenant user information of an API Portal gateway, if you have one or more APIs associated to it. If you still continue to modify the information, CentraSite displays a warning message.
- When you make changes to the endpoint of a gateway, for example API Gateway, you must update the details page of the API Gateway with the new configuration, and then republish the API Gateway to CentraSite to put those changes into effect.
- Some attributes accept only specific types of information. For example, for a URL type attribute, you must supply a URL when you edit that attribute.
- Some attributes are designed to be read-only and cannot be edited even if they appear in a gateway on which you have Modify or Full permission.

➤ **To modify the details of a gateway asset**

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. To filter the list to display the available API Gateway, API Portal, Mediator, and Insight Server gateways, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, do the following:
 - a. Locate **Applicable Scopes**.
 - b. Select the required type of gateway asset from the drop-down list:
 - **API Gateway**
 - **API Portal**
 - **Mediator**
 - **Insight**
 - c. Click the plus button next to the drop-down box or press Enter to add the scope to the search recipe.

CentraSite displays the list of available API Gateway, API Portal, Mediator, and Insight Server gateways.

3. In the displayed list, select the gateway whose details you want to modify.

This opens the Gateway Details page. The details include:

- The gateway asset's Basic Information. This includes the gateway's asset type, short description about the gateway, the last modified date, the owning organization, and the user. In addition to the above information, gateway instance includes the following details:
 - The Web application URL of webMethods API Gateway.
 - The Home page URL of webMethods API Portal.
 - Name of an onboarding organization for API Portal users in CentraSite.
- The CentraSite Communication information. This includes the endpoint URL of CentraSite, and the username of a CentraSite user.
- The API Gateway Communication information. This includes the endpoint URL of API Gateway, and the username of a technical user in API Gateway.

A technical user displayed in the **API Gateway Communication** profile is used to publish API assets from CentraSite to API Gateway, and a user who is part of the *MyAPIGateway* Synchronization Group is used to send data from API Gateway to CentraSite.

- The API Portal Communication information. This includes the endpoint URL of API Portal gateway, and the username of a technical user in API Portal.
 - The Mediator Communication information. This includes the endpoint URL of Mediator gateway, and the username of a user who is part of the *MyMediatorGateway* Synchronization Group.
 - The list of API assets published to the gateway.
4. On the actions bar of the Gateway Details page, click **Edit**.
 5. In the **Basic Information** profile, modify the properties of the displayed data fields, as required.

Field	Description
Name	Name of the gateway.
Organization	Name of the organization to which the gateway belongs.
Onboarding Consumer Organization	For API Portal gateway only. Name of the organization that contains the users of API Portal gateway.
	Note: You are not allowed to modify the value of this field if there is at least one API Portal user onboarded in this organization.
Owner	Name of the user to which the gateway belongs.

Field	Description
Description	Description of the gateway.

6. In the **CentraSite Communication** profile, modify the properties of the displayed data fields, as required. This is applicable only for API Gateway, API Portal, and Mediator gateways.

Field	Description
Endpoint	The endpoint URL of CentraSite.
Username	The user ID of the CentraSite user who has one of the following appropriate roles: <ul style="list-style-type: none"> ■ CentraSite Administrator ■ Organization Administrator ■ API Gateway Administrator ■ API Portal Administrator ■ Mediator Administrator ■ Insight Server Administrator
Password	The password of the CentraSite user specified in the above Username field.

7. In the **API Gateway / API Portal / Mediator Communication** profile, modify the properties of the displayed data fields, as required. This is applicable only for API Gateway, API Portal, and Mediator gateways.

Field	Description
Endpoint	<i>Read-only.</i> The endpoint URL of API Gateway, API Portal, or Mediator gateway.
API Gateway WebApp URL	This is applicable only for API Gateway. The Web application URL of API Gateway. For example, <code>http://myHostname:9072</code> .

Note:

Various aspects of the way **API Gateway WebApp URL** functions are as follows:

- If a value is specified for this attribute, then CentraSite uses this URL as the API Gateway Web application URL.
- If a value is not specified for this attribute, then CentraSite automatically populates the dynamic URL based on the value that

Field	Description
	<p>was sent by API Gateway on publishing the API Gateway asset from CentraSite to the API Gateway instance.</p> <ul style="list-style-type: none"> API Gateway sends the Web application URL based on the value specified in the Web application load balancer URL field in API Gateway (go to <Username> > Administration > General > Load balancer) as described in <i>webMethods API Gateway Administrator's Guide</i>. For example, <ul style="list-style-type: none"> if a value is specified for this field, API Gateway sends the defined Web application load balancer URL to CentraSite. if the value is not specified for this field, API Gateway sends the default hostname and port number as Web application URL to CentraSite. For example: <code>http://myHostname:9072</code> <p>Important: If you are modifying the value of this API Gateway WebApp URL attribute, you have to republish the API Gateway asset for the changes to take effect.</p>
Tenant	<p>This is applicable only for API Portal gateway.</p> <p>Name of a tenant user in API Portal gateway.</p> <p>Important: You are not allowed to change the value of Tenant property if there is an API already published to this gateway.</p>
Use CentraSite Credentials	This check box enables the usage of CentraSite credentials for authenticating against this gateway.
Username	The user ID of the API Gateway, API Portal, or Mediator user who has an API Runtime Provider role in CentraSite.
Password	The password of the user specified in the above Username field.
Sandbox	The sandbox category of the gateway.

8. In the **Published APIs** profile, examine the following information.

Attribute	Description
Name	A deep link to open the API asset details page directly in CentraSite Business UI.
Description	A descriptive information of the asset.
Version	The user-assigned version identifier for the asset.
View in API Gateway	This is applicable only for API Gateways.

Attribute	Description
	A deep link to open the API asset details page directly in the API Gateway user interface.
View in API Portal	This is applicable only for API Portal gateways. A deep link to open the API asset details page directly in the API Portal user interface.

- Click **Save** to save the updated gateway.

If you have made changes to the CentraSite communication information, such as the CentraSite **Username** or **Password** field, publish the gateway again.

Setting Instance-Level Permissions on Gateway Asset

Pre-requisites:

To assign instance-level permissions on a gateway, you must have one of the following roles:

- CentraSite Administrator: instances of API Gateway, API Portal, Mediator, and Insight Server gateways in any organization in CentraSite.
- Organization Administrator: instances of API Gateway, API Portal, Mediator, and Insight Server gateways in your organization.
- API Gateway Administrator: instances of API Gateway in the specific organization to which your API Gateway Administrator role applies.
- API Portal Administrator: instances of API Portal gateway in the specific organization to which your API Portal Administrator role applies.
- Mediator Administrator: instances of Mediator and Insight Server gateways in the specific organization to which your Mediator Administrator role applies.

In addition to the above, you can assign instance-level permissions on a gateway instance if you have the Full instance-level permission on the gateway itself.

You assign instance-level permissions on a gateway asset using the **Permission** action in the Gateway Details page.

Note:

If you do not see the **Permission** action in the Gateway Details page, it is probably because you do not have the required role or permission to modify the permission details of the gateway.

When assigning instance-level permissions on a gateway, keep the following points in mind:

- You can assign permissions to any individual user or group defined in CentraSite.
- The groups to which you can assign permissions include the following system-defined groups:

Group Name	Description
Users	All users within a specified organization.
Members	All users within a specified organization and its child organizations.
Everyone	All users of CentraSite <i>including guest users</i> .

- If a user is affected by multiple permission assignments, the user receives the union of all the assignments. For example, if group ABC has Modify permission on an API Gateway and group XYZ has Full permission on the same API Gateway, users that belong to both groups will, in effect, receive Full permission on the API Gateway.

> To assign instance-level permissions on a gateway asset

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. To filter the list to display the available API Gateway, API Portal, Mediator, and Insight Server gateways, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, do the following:
 - a. Locate **Applicable Scopes**.
 - b. Select the required type of gateway asset from the drop-down list:
 - **API Gateway**
 - **API Portal**
 - **Mediator**
 - **Insight**
 - c. Click the plus button next to the drop-down box or press Enter to add the scope to the search recipe.

CentraSite displays the list of available API Gateway, API Portal, Mediator, and Insight Server gateways.
3. In the displayed list, select the gateway you want to assign the user and group permissions.

This opens the Gateway Details page. Also, the actions bar displays a set of actions that are available for working with the displayed gateway.
4. On the actions bar of the Gateway Details page, click **Permission**.

This opens the **Assign Permissions** dialog box.

5. To add users or groups to the **User and Group Permissions** list, do one of the following:
 - a. Type a partial string in the **Add User or Group** text box. CentraSite applies the filter to the users and groups in registry.
 - b. Select the user or group to which you want to assign permissions.
 - c. Click the plus button next to the text box or press Enter to add the user or group to the **User and Group Permissions** list.

-OR-

- a. Click **Choose**.
This opens the **Choose Users and Groups** dialog.
 - b. Type a partial string in the **Add User or Group** text box. CentraSite applies the filter to the users and groups in registry.
 - c. Click the **Search** icon.
 - d. Select the users and groups to which you want to assign permissions.
 - e. Click **OK**.
6. To remove a user or group from the **User and Group Permissions** list, select the **Delete** icon beside the group name or user ID.
 7. For Setting Instance-Level Profile Permissions. Select the **View**, **Modify**, and **Full** check boxes to assign specific permissions to each user and group in the Users and Groups Permissions list as follows:

Permission	Allows the selected user or group to...
View	Examine the details of the gateway.
Modify	Examine and modify the details of the gateway. This permission also allows the selected user or group to publish and unpublish APIs to the gateway.
Full	Examine and modify the details, and delete the gateway. This permission also allows the selected user or group to assign instance-level permissions to the gateway.

8. Click **Save**.

Unregistering Gateways from CentraSite

Pre-requisites:

To unregister gateways from CentraSite, you must have one of the following roles:

- **CentraSite Administrator:** instances of API Gateway, API Portal, and Mediator gateways within any organization in CentraSite.
- **Organization Administrator:** instances of API Gateway, API Portal, and Mediator gateways in the specific organization to which your Manage Organizations permission applies.
- **API Gateway Administrator:** instances of API Gateway in the specific organization to which your API Gateway Administrator role applies.
- **API Portal Administrator:** instances of API Portal gateway in the specific organization to which your API Portal Administrator role applies.
- **Mediator Administrator:** instances of Mediator gateway in the specific organization to which your Mediator Administrator role applies.

In addition to the above, you can unregister a gateway if you have the Full instance-level permission on the gateway itself.

Note:

This functionality is not applicable to Insight Server gateways.

Unregistering a gateway causes CentraSite to suppress interactions with the particular gateway.

You usually unregister a gateway, for example API Portal, as a fallback option for the following reasons:

- The API Portal server is irreversibly down, for example, the machine hosting the API Portal has crashed, or is inaccessible.
- An internal or application error on the CentraSite or API Portal server.
- To edit an API Portal (for example, to modify the hostname of an API Portal).
- To suspend publishing of APIs and handling requests (new, renew, revoke) of access tokens for the API Portal registry (temporarily or permanent).

You unregister a gateway from CentraSite using the **Unpublish** action in the Gateway Details page. The **Unpublish** action is NOT visible in the Gateway Details page for the following conditions:

- If the required role to unregister the particular gateway is not met.
- If the Mediator gateway does not have any API published to it.

When you unregister, for example, an API Portal gateway, CentraSite executes the following operations:

1. Removes any relationship between the APIs (which are already published to API Portal) and the API Portal object in the repository (that is, CentraSite removes any existing association between the API Portal registry and the APIs published to it).
2. Unpublishes all APIs that were published to the API Portal gateway.
3. Deletes onboarded users specific to the API Portal gateway from CentraSite registry.

4. Deletes access tokens of the non-existing API Portal users from the API Gateway or Mediator gateway and the CentraSite registry.

If at the time of unregistration, Mediator gateway is unavailable, the access tokens of the non-existing API Portal users are marked as revoked. At a later time, when the API is republished to Mediator, CentraSite sends a list of valid access tokens to Mediator. Then, the Mediator removes the access tokens that are associated with the non-existing users.

➤ To unregister gateways

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. To filter the list to display the available API Gateway, API Portal, and Mediator gateways, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, do the following:
 - a. Locate **Applicable Scopes**.
 - b. Select the required type of gateway asset from the drop-down list:
 - **API Gateway**
 - **API Portal**
 - **Mediator**
 - c. Click the plus button next to the drop-down box or press Enter to add the scope to the search recipe.

CentraSite displays the list of available API Gateway, API Portal, and Mediator gateways.

3. In the displayed list, select the gateway instances you want to unregister from CentraSite.

If you have selected multiple instances where one or more of them are already unregistered from CentraSite, CentraSite initiates the unregistration mechanism on the selected gateway instances, and ignores the already unregistered instances.

You can also unregister a single gateway instance from its page.

4. On the actions bar of the Search Results page, click **Unpublish**.
5. In the **Unpublish** dialog box, click **Unpublish**.

If you encounter a problem when unregistering gateways, possibly because the machine hosting any gateway instance is inaccessible, CentraSite displays a dialog box describing the problem.

6. Select the **Force Unpublish** check box to unregister the gateway instances forcefully, and then click **Unpublish**.

When this option is selected, CentraSite ignores any failures even if the gateway instance is inaccessible, and clears all data from the CentraSite database.

Deleting Gateways

Pre-requisites:

To delete gateways of an organization, you must have one of the following roles:

- CentraSite Administrator: instances of API Gateway, API Portal, Mediator, and Insight Server gateways in any organization in CentraSite.
- Organization Administrator: instances of API Gateway, API Portal, Mediator, and Insight Server gateways in your organization.
- API Gateway Administrator: instances of API Gateway in the specific organization to which your API Gateway Administrator role applies.
- API Portal Administrator: instances of API Portal gateway in the specific organization to which your API Portal Administrator role applies.
- Mediator Administrator: instances of Mediator and Insight Server gateways in the specific organization to which your Administrator role applies.

Deleting a gateway permanently removes an instance of the gateway asset from the CentraSite registry. When you delete a gateway, CentraSite does not delete a gateway if it has at least one API published to it.

> To delete gateways

1. To filter the list to display the available API Gateway, API Portal, Mediator, and Insight Server gateways, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, do the following:
 - a. Locate **Applicable Scopes**.
 - b. Select the required type of gateway asset from the drop-down list:
 - **API Gateway**
 - **API Portal**
 - **Mediator**
 - **Insight**
 - c. Click the plus button next to the drop-down box or press Enter to add the scope to the search recipe.

CentraSite displays the list of available API Gateway, API Portal, Mediator, and Insight Server gateways.

2. In the displayed list, select the gateways you want to delete in CentraSite.

You can also delete a single gateway from the Gateway Details page.

3. On the actions bar of the Search Results page, click **Delete**.
4. Click **Yes** in the confirmation dialog box.

The selected gateways are permanently removed from the CentraSite Registry Repository.

Managing Gateways through Command Line Interface

This section describes operations you can perform to manage the gateways through the CentraSite Command Line Interface.

Viewing the Gateways List

Pre-requisites:

To list the existing gateways through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `list gateway` for this purpose.

> To list available gateways

- Run the command `list gateway`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list gateways -user <USER-ID> -password <PASSWORD> [-url <CENTRASITE-URL>]`

The input parameters are:

Parameter	Description
CENTRASITE-URLurl	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd|sh list gateways -user
Administrator -password manage -url http://localhost:53307/CentraSite/CentraSite
```

The response to this command could be:

```
-----GATEWAYS-----
Name : DevMediator
Key : uddi:95f79c00-52a9-11e4-969c-d398081308d4
Description : DevMediator
Type : Mediator
Username : Administrator
Gateway URL : http://localhost:5555
CentraSite Username : Administrator

Successfully executed the command : list gateways
```

Adding a Custom Gateway

Pre-requisites:

To create a gateway through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `set gateway` for this purpose.

> To create a gateway

- Run the command `set gateway`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set gateway -user <USER-ID> -password <PASSWORD> [-url <CENTRASITE-URL>] -file <CONFIG-FILE> -gatewayPassword <GATEWAY-PASSWORD> [-csPassword <CS-PASSWORD>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
CONFIG-FILE	Name of the configuration file which contains the gateway parameters. For additional details, see “Configuration File” on page 1393.
GATEWAY-PASSWORD	The password that CentraSite uses to communicate with the gateway.

Parameter	Description
CS-PASSWORD	(Optional). The password that the gateway uses to communicate with CentraSite. If this parameter is not specified, the password of the user logged in is used.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set gateway -user
Administrator -password manage -url http://localhost:53307/CentraSite/CentraSite
-file c:\tmp\mediator-config.xml -gatewayPassword manage
```

The response to this command could be:

```
Executing the command : set gateway
Gateway DevMediator created

Successfully executed the command : set gateway
```

Note:

To update an existing gateway in CentraSite, you can use the `set gateway` command and provide an edited configuration file as input to the command along with the key that identifies the gateway you want to update. The gateway key can be obtained by using the `list gateways` command.

Configuration File

The contents of the configuration file are listed as follows:

Tag Name	Description
-type	(Optional). The gateway type: API Portal or Mediator. The default value is Mediator.
-name	The name of the gateway.
-description	(Optional). Description for the gateway.
-deploymentEndpoint	(Optional). Deployment endpoint of the gateway. The default value is <code>http://localhost:5555</code> .
-user	User ID of the gateway user.
-sandbox	(Optional). The sandbox category to be used to classify the gateway instance.
-organization	(Optional). Name or key of the organization that submits the gateway information. The default value is <code>Default Organization</code> .
-key	(Optional). Key used to update the gateway.

Tag Name	Description
-csUser	(Optional). CentraSite user's credentials. By default, the credentials of the user who has logged in to CentraSite is used.

Sample configuration file:

```
<gateway>
<!-- type of the Gateway -->
<type>Mediator</type>
<!-- name of the gateway -->
<name>DevMediator</name>
<!-- Description of the gateway -->
<description>DevMediator</description>
<!-- Deployment Endpoint for the gateway -->

<deploymentEndpoint>http://localhost:5555</deploymentEndpoint>
<!-- Mediator user Id -->
<user>Administrator</user>
<!-- CentraSite user Id -->
<csUser>Administrator</csUser>
<!-- Mediator user password -->
<password>Administrator</password>
<!-- Name/Key of the sandbox category -->
<!-- <sandbox>Production</sandbox> -->
<!-- Name/Key of submitting organization -->
<!-- <organization>Default Organization</organization> -->

<!-- Key to be used for update -->
<!-- <key></key> -->

</gateway>
```

Deleting Gateway

Pre-requisites:

To delete a gateway through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `remove gateway` for this purpose.

➤ To delete a gateway

- Run the command `remove gateway`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd remove gateway -user <USER-ID> -password <PASSWORD> [-url <CENTRASITE-URL>] -gateway <GATEWAY>`

The input parameters are:

Parameter	Description
CENTRASITE-URLurl	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
GATEWAY	Name of the gateway you want to delete.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd remove gateway -user
Administrator -password manage -url http://localhost:53307/CentraSite/CentraSite
-gateway DevMediator
```

The response to this command can be:

```
Executing the command : remove gateway
Gateway DevMediator removed

Successfully executed the command : remove gateway
```

Publishing APIs to Gateways

Pre-requisites:

To publish APIs to gateways, for example, `webMethods Mediator`, through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `deploy` for this purpose.

» To publish APIs to gateways

- Run the command `deploy`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd deploy -user <USER-ID> -password <PASSWORD> [-url <CENTRASITE-URL>] -virtualService <VIRTUAL-SERVICE> -gateway <GATEWAY>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .

Parameter	Description
PASSWORD	The password for the CentraSite user identified by the parameter USER-ID.
VIRTUAL-SERVICE	The name or key of the virtual service you want to deploy.
GATEWAY	The gateway to which the virtual service identified by the parameter virtualService is to be deployed.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd deploy -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-virtualService TripPinService -gateway DevMediator
```

The response to this command can be:

```
Executing the command : deploy
Successfully executed the command : deploy
```

Unpublishing APIs from Gateways

Pre-requisites:

To unpublish APIs from gateways, for example, webMethods Mediator through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `undeploy` for this purpose.

> To unpublish APIs from gateways

- Run the command `undeploy`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd undeploy -user <USER-ID> -password <PASSWORD> [-url <CENTRASITE-URL>] -virtualService <VIRTUAL-SERVICE> -gateway <GATEWAY>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter USER-ID.

Parameter	Description
VIRTUAL-SERVICE	The name or key of the virtual service you want to unpublish from the gateway.
GATEWAY	The gateway from which the virtual service identified by the parameter <code>virtualService</code> is to be unpublished from the gateway.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd undeploy -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-virtualService TripPinService -gateway DevMediator
```

The response to this command can be:

```
Executing the command : remove gateway
Gateway DevMediator removed

Successfully executed the command : remove gateway
```

Consumer Management

In CentraSite there are three categories of consumers:

- **Registered Consumers:** Refers to the developers who can register to consume the assets available in the catalog. You can provide registered consumers access to asset's metadata or develop processes that notify them when modifications are made.
- **Arbitrary Asset Consumers:** Refers to any arbitrary asset that consumes any other asset in the CentraSite registry for the design-time usage.
- **Consumer Application Consumers:** Refers to the consumer application that consumes (invokes) virtual services at run-time. These consumers are represented in the CentraSite registry by instances of the Application asset which are used by Mediator to determine from which computer application a request for a virtual service originated.

The ability of Mediator to relate a message to a specific consumer application enables Mediator to:

- Indicate the consumer application to which a logged transaction event belongs.
- Monitor a virtual service for violations of a service-level agreement (SLA) for a specified consumer application.
- Control access to a virtual service at run-time (that is, allow only authorized consumer applications to invoke a virtual service).

Definition of Application Assets in CentraSite

The Application asset type is one of the predefined asset types installed with CentraSite. Application assets are used by the policy-enforcement point (PEP) to determine from which consumer application a request for an asset originated. An Application asset defines the precise characteristics by which the PEP can identify or authenticate messages from a specific consumer application at run-time. An application asset has the following attributes for specifying these identifiers:

- *IPv4 Address*: specifies one or more 4-byte IPv4 addresses that identify requests from a particular consumer application. (This attribute is queried when the Identify Consumer action is configured to identify consumer applications by IP address.)

Example: 192.168.0.10

- *IPv6 Address*: specifies one or more 128-bit IPv6 addresses that identify requests from a particular consumer application. See the IPv6 addressing architecture specification for details of this format.

Example: 1234:5678:9ABC:DEF0:1234:5678:9ABC:DEF0

- *Identification Token*: specifies the host names, user names, or other distinguishing strings that identify requests from a particular consumer application. (This attribute is queried when the Identify Consumer action is configured to identify consumer applications by host name, HTTP user name, WSS user name, or a custom token.)
- *Consumer Certificate*: specifies the X.509 certificates that identify requests from a particular consumer. (This attribute is queried when the Identify Consumer action is configured to identify consumer applications by a consumer certificate.)

Identification of Consumer Applications at Run-Time

To determine the consumer application from which a request was submitted, a virtual service must have a run-time policy that includes an Evaluate * action. This action extracts a specified identifier from an incoming request and locates the Application asset defined by that identifier.

For example, if you configure the Evaluate IP Address action to identify consumers by IP address, Mediator extracts the IP address from a request's HTTP header and searches its list of Application assets for the application that is defined by that IP address.

You can configure the following Evaluate * actions to identify consumer applications based on the following information in a request message:

Identifier	Description
IP Address	The IP address from which the request originated.
Host Name	The name of the host machine from which the request originated.
HTTP Authentication Token	The user ID submitted by the requestor when it was asked to provide basic HTTP credentials (user name and password).

Identifier	Description
WS-Security Authentication Token	The WSS username token supplied in the header of the SOAP request that the consumer application submitted to the virtual service.
Consumer Certificate	The X.509 certificate supplied in the header of the SOAP request that the consumer application submitted to the virtual service.
XPath Expression	The custom token supplied using an XPath expression in the header of the request that the consumer application submitted to the virtual service.
OAuth2 Authentication Token	The OAuth2 credentials supplied in the header of the SOAP request that the consumer application submitted to the virtual service.
Kerberos Authentication Token	The Kerberos token supplied in the header of the SOAP request that the consumer application submitted to the virtual service.

When deciding which type of identifier has to be used to identify a consumer application at run-time, consider the following points:

- Identifiers that represent user names are often not suitable because the identified users might submit requests from multiple consumer applications.
- Although identifying applications by IP address or host name is often a suitable choice, it does create a dependency on the network infrastructure. If a consumer application moves to a new machine, or its IP address changes, you must update the identifiers in the Application asset.
- Using X.509 certificates or a custom token that is extracted from the SOAP message itself (using an XPATH expression), is often the most trouble-free way to identify a consumer application.

Registration of Application assets as Consumers of Virtual Service

You use the **Consume** action in CentraSite Business UI to register an Application asset as consumer of a virtual service. This action establishes an association between the Application asset and the virtual service that it consumes. Registering an Application asset with a virtual service also enables you to use the Asset Navigator feature in CentraSite Business UI to quickly determine which virtual services a consumer application consumes using the **Service Versioning And Consumers** usecase.

Additionally, if you use the Authorize User policy action to control access to a virtual service at run-time, only registered consumer applications are allowed to invoke the virtual service. Consequently, when you use this form of access control, the consumer applications that are permitted to use a virtual service *must be registered to the virtual service*.

You register Application assets as consumers of an asset using the following methods:

- CentraSite Business UI
- CentraSite Control

Synchronization of Application assets in CentraSite with Mediator

Mediator maintains a list of consumer applications specified in CentraSite that are authorized to access the API published to Mediator. When Mediator identifies a consumer application at run-time, it searches the list of Application assets that it maintains. This list is synchronized from the CentraSite registry when you synchronize the consumer applications in CentraSite with Mediator.

There are two different lists of consumers in a policy enforcement point (PEP) such as webMethods Mediator:

- **List of Registered Consumers:** List of users and consumer applications (represented as Application assets) who are registered as consumers for an API in CentraSite, and available in Mediator.

When you synchronize applications who are already registered as consumers for an API to Mediator, these synchronized applications are maintained as a list of registered consumers in Mediator.

- **List of Global Consumers :** List of all users and consumer applications (represented as consumers) available in Mediator.

When you synchronize applications who are not yet registered as consumers for an API to Mediator, these synchronized applications are maintained as a list of global consumers in Mediator.

You synchronize consumer applications in CentraSite with Mediator using the following methods:

- CentraSite Business UI
- CentraSite Control
- CentraSite Command Line Tool

Consumer Registration

Note:

Beginning with version 9.9, CentraSite does not support the Consumer Registration functionality in CentraSite Control. As a result:

- The **Register as Consumer** action in the details page of an Application asset is removed from CentraSite Control.
- You cannot modify details of the consumers using the **Consumers** profile of an Application asset in the CentraSite Control. You can only view details of the consumers such as: Name, Description, Type of Consumer, Owner, Organization, Created date, Modified date, and the Lifecycle State.
- The **Consumer Registrations** section that provides functionality for viewing a summary of all consumer registration requests in conjunction with **Pending Registrations** and **Registration Requests** is removed from the **Home > My CentraSite > Inbox** page of CentraSite Control.

- The **My Pending Consumer Registration Requests** data feed that was used to construct a custom portlet for rendering the list of all consumer registration requests in CentraSite Business UI has been deprecated and will be removed in a future release.

Therefore, you cannot register users, groups, or application assets as consumers of an asset using the CentraSite Control. Instead, you can use the enhanced interface, CentraSite Business UI that supports consumer registration for users, groups, and arbitrary assets (in contrast, earlier versions of CentraSite Business UI supported a standardized interface for consumer registration of Application assets only). Documentation of the prior consumer registration interface is available to CentraSite customers who have a current maintenance contract in Empower Product Support Website.

The term *consumer registration* means providing users the ability to consume assets. Consumer registration with CentraSite enables asset providers to establish an enhanced level of protection and access control as to who can consume the assets using configurable approval workflows and thus allows the asset providers to visualize and control the consumption of their assets.

CentraSite's flexible and extensible asset catalog enables the asset providers to expose their reusable assets. At any time, asset consumers can discover the reusable assets in the asset catalog which functions as the central registry, and reuse them in their own applications. Consumer registration functionality provides the mechanism to establish the consumer-provider relationships in CentraSite at both design-time and run-time.

When you execute the consumer registration feature in CentraSite Business UI, CentraSite registers the asset as a consumer of the asset. However, if you have imposed an approval process, it triggers a review and approval process that includes the following steps:

1. CentraSite submits the request to the designated approvers for review and approval.
2. If the request is approved, CentraSite executes the consumer-registration policy. This policy registers the application asset with the virtual service.

Scope of Consumer Registration at Design-Time

At design-time, this functionality allows you to:

- Register any arbitrary asset as consumer of other assets in your organization.
- Fetch details about all of the consumers of a specific asset, also details about all of the assets that a specific asset is consuming.
- Visualize the consumption relationship that exists between asset providers and asset consumers. This helps you to identify the artifacts in the registry that is affected if an asset is not available or must be changed. You can visualize the dependencies between assets using the Asset Navigator page.

Scope of Consumer Registration at Run-Time

At run-time, this functionality lets you:

- Enforce a level of run-time security checks by allowing a consumer to specify details about the consuming applications that in turn is used for identification of that application during invocation of the asset at run-time.
- Determine whether a particular asset consumer is authorized to invoke the target asset.
- Access the asset's metadata (that is, view additional profiles) and receive notifications when modifications are made to the asset that they consume.

Permissions Required to Register Consumers

Any user with a **View** instance-level permission on the asset to be consumed can register consumers for that asset. In addition to registering consumers for a particular asset, a user can also create new assets and register the newly created assets as consumer of that asset, provided that the user has Create Assets permission in CentraSite.

The Design/Change-Time Policy Used for Consumer Registration

The enhanced consumer registration functionality enables the consumer registration process without explicitly creating a consumer-registration policy and also completes the consumer registration process without requiring the owner of the asset to review and accept the user's request.

If you want to impose an approval process on consumer registration, that is, control which consumers can invoke an asset, you can create a design-time policy with one of the CentraSite's built-in approval actions for the **OnConsumerRegistration** event. For example, you can create a design-time policy, named Consumer-Registration Policy, which requires a designated group of approvers to review and approve all consumer requests for invoking the asset. Upon reviewing, the approvers may approve or reject such requests.

When you register as consumers of an asset, the policy is triggered and the Register as Consumer request is submitted to all members of the approval list specified in the Initiate Approval action. Then, the approvers can either approve or decline the request. If the approvers approve the request, the consumers is registered as consumers for the asset. Consider, you have configured the **Set Consumer Permission** action in this policy. CentraSite assigns the appropriate permissions to the specified set of users, groups, and arbitrary assets.

Creating Consumer Registration Policy

To create a consumer registration policy, you must have the Manage Design/Change-Time Policies permission in CentraSite.

The consumer-registration policy is a policy that includes the Register Consumer action and executes on the OnConsumerRegistration event. The OnConsumerRegistration event occurs when you submit the registration request. *CentraSite does not provide a consumer-registration policy out-of-the-box.* You must create this policy for your instance of CentraSite.

Important:

If you are using the Authorize User policy action to control access to a virtual service at run time, you should strongly consider including an approval step in your consumer-registration policy. When you use this form of access control on a virtual service, registering a consumer application with the virtual service grants that consumer application permission to invoke the service. To ensure that only authorized applications are registered with a virtual service, you might want to have a security administrator review and approve this type of registration request.

You can create a consumer registration policy in CentraSite Control using the Add Design/Change-Time Policy page.

➤ To create a consumer registration policy

1. In CentraSite Control, go to **Policies > Design/Change Time**.
2. Click **Add Policy**.
3. In the **Policy Information** panel, specify the following fields:

In this field...	Do the following...
Name	Type a name for the new consumer registration policy. The policy name can contain any character (including spaces).
Description	<i>Optional.</i> Type a description for the new policy. This description appears when a user displays a list of policies in the user interface.
Version	<p><i>Optional.</i> Specify a version identifier for the new policy.</p> <p>Note: The version identifier does not need to be numeric.</p> <p>Examples:</p> <pre>0.0a 1.0.0 (beta) Pre-release 001 V1-2007.04.30</pre> <p>The version identifier you type here is the policy's public, user-assigned version identifier. CentraSite also maintains an internal, system-assigned version number for the policy.</p>
Priority	Type an integer that represents the priority of this policy with respect to other policies that might be triggered by the same event.

4. In the **Scope** panel, specify the object and event types to which the policy applies.

In this field...	Do the following...
Object Types	Choose Asset (of any type).
Event Types	Choose OnConsumerRegistration .
Organization	Specify the organization to which the policy applies or select All if the policy applies to all organizations.

5. Click **Next**.
6. *To impose an approval process on OnConsumerRegistration event.* On the policy's **Actions** tab, add the following actions. Ensure that the approval action *precedes* the Register Consumer action:
 - Initiate Approval —OR— Initiate Group-dependent Approval
 - Register Consumer
7. Click **Finish** to save the new (as yet incomplete) policy.
8. Complete the new policy by configuring the approval action's input parameters.
9. Insert additional actions before and after this pair of actions as necessary. The following example shows an action list that obtains the required approval, executes the registration process, and then grants instance-level permissions to the consumers that the policy registers:

```
Initiate Approval
Register Consumer
Set Consumer Permission
```
10. Activate the policy when you are ready to put it into effect.

Design-Time Consumer Registration Scenario

While registering a design-time consumer, the user can select assets as consumers of an asset. However, these assets selected for consumption should have their asset types configured as consuming types for the asset you want to consume in CentraSite. For more details on configuring the assets as consuming types of assets of a particular type, see [“Adding an Asset Type” on page 253](#).

To impose an approval process for consumer registration process, ensure that you have created the required Consumer Registration design-time policy.

At design-time, you might want to register as consumer an asset instance of a base type, for example, an asset of Service type, and would consume an XML Schema asset. This would help to indicate the design-time asset usage.

➤ To register as consumer an asset in design-time scenario

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link that is located in the upper-left corner of the menu bar.
 - Click the **Search** icon that is located next to the **Scope** list. The default search scope is **Assets**.
2. In the **Additional Search Criteria** list, select **Asset Types**.
 3. To search for the list of asset types, click **Choose**.

This opens the **Choose Asset Types** dialog box.
 4. In the **Choose Asset Types** dialog box, follow these steps:
 - a. Click the chevron next to **Everything** option button.

This displays a list of defined asset types in CentraSite.
 - b. In the list of asset types, select any of the **Service** type for whose instance you want to register the consumers.
 - c. Click **OK**.

This displays a list of the Service assets that were defined using the selected types, and that are available to you in the Search Results page.
 5. Click a Service asset you want to consume.

This opens the Service Details page. Also, the actions bar displays a set of actions that are available for working with the displayed service.
 6. On the actions bar of the Service Details page, click **Consume**.

This opens the **Consume Asset** dialog box for registering a consumer.
 7. To select an existing asset as consumer, do one of the following:
 - Type the name of the asset in the text box.

As you enter the partial text, CentraSite returns the top n assets that meet your search text.
 - Select the asset you want to register as a consumer. Then click the plus button next to the text box or press Enter to add the asset to the list of currently selected assets in the **Consume Asset** dialog box.
 - Click **Choose**.

This opens the **Choose Consumer Assets** dialog box.
 - In the **Choose Consumer Assets** dialog box, do the following:

1. Type the name of the asset that you want to register as a consumer in the text box. Then click the **Search** icon.

CentraSite populates a set of instances whose asset types are defined to be consumers to instances of this asset type.

2. Select the asset or multiple assets that you want to register, from the list displayed.
3. Click **OK**.

8. To create and register a new asset:

- a. Click the **Create asset** link.

This opens the Create New Asset page.

- b. Select the asset type for which an asset is created. The **Types** list reflects its behavior in terms of the type definition for the displayed asset. The list contains the list of top-level asset types that you have defined as consuming type in the asset type definition.

For example, if you have defined Applications, Users, Groups, and XML Schemas as consumers in the REST Service type definition, when trying to create a new asset using the **Consume** action of a displayed REST service, the **Types** list contains only asset types, namely, Applications, and XML Schemas that are the top-level asset types and defined as consuming types of the REST service.

If none of the asset types, for example, Users, Groups, and a Custom Type, defined as consuming types in the type definition of a displayed REST service is a top-level type, the **Create asset** link does not appear in the **Consume Asset** dialog box.

- c. After you specify the value for all of the required attributes, click **Save** to save the new asset.

The Details page for the asset that you just created is displayed.

- d. Display the Service Details page of the asset you want to consume, and then click **Consume**.
- e. Repeat step 7 to select the newly created asset to register as a consumer of the Service asset.

9. To remove an asset that is currently selected for consumption, hover over the asset you want to remove. The **Delete** icon appears.

- a. Click **Delete**.

A confirmation message stating that the selected asset will be removed from the selection list appears .

- b. Click **Yes** to confirm.

10. In the **Reason for Request** field, type the reason for requesting consumption of the asset.

The details entered in this field are visible to all designated approvers who would review the consumer registration request for the asset.

11. Click **Consume**.

If there are any approvals configured for the consumer registration process, then a consumer registration request is sent to the designated list of approvers. Otherwise, the selected consumers are immediately registered with the asset and the registration is reflected through the **Consumers** count in the **Basic Information** profile.

Run-Time Consumer Registration Scenarios

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

Pre-requisite:

To impose an approval process for consumer registration process, ensure that you have created the required Consumer Registration design-time policy.

At run-time, you might want to register as consumer an asset instance of a virtual type, for example, an asset of Virtual Service type, in which case you would have to specify the authentication settings or consumer identifiers or both, and would invoke the asset to query and obtain the data. This would help to indicate the run-time asset usage.

An API provider (owner of the API) specifies the type of authentication (API key or OAuth2 token using the consumption settings) and configures a set of Evaluate * actions (using the run-time policies) to identify and authorize the consumer that request the consumption access to the metadata of the particular API. Based on the specified enforcement definitions for the API, CentraSite offers interactive user interfaces for the following four exemplary scenarios:

Is API Consumption Settings Is an Evaluate Action Configured? For Procedures... Configured?

No	No	See, Scenario A
No	Yes	See, Scenario B
Yes	No	See, Scenario C
Yes	Yes	See, Scenario D

Scenario A: Virtual Service without API Consumption Settings and Evaluate Policy Actions

Prerequisites for scenario A:

- *The asset should be an instance of the virtual type - Virtual Service, Virtual REST Service, Virtual XML Service.*
- Make sure that the API consumption settings or an Evaluate * policy action *is not configured* for the asset.

➤ **To register as a consumer of an asset for Scenario A:**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link that is located in the upper-left corner of the menu bar.
 - Click the **Search** icon that is located next to the **Scope** list. The default search scope is **Assets**.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the list of asset types, click **Choose**.
4. In the **Choose Asset Types** dialog box, follow these steps:
 - a. Click the chevron next to **Everything** option button.
 - b. In the list of asset types, select any of the **Virtual Service** type for whose instance you want to register the consumers.
 - c. Click **OK**.

This displays a list of the Virtual Service assets that were defined using the selected types, and that are available to you in the Search Results page.

5. Click a Virtual Service asset you want to consume.

This opens the Virtual Service Details page. Also, the actions bar displays a set of actions that are available for working with the displayed virtual service.

6. On the actions bar of the Virtual Service Details page, click **Consume**.
7. In the **Consume Asset** dialog box, you can do one of the following:
 - Register an existing asset as consumer of the Virtual Service asset.
 - Create a new asset in CentraSite and then register the newly created asset as consumer of Virtual Service asset.

When you register as a consumer of an asset instance of Virtual Service type, which does not include the consumption settings or evaluate actions, the fields displayed for consumption are the same as for the Service asset instance of base type.

Scenario B: Virtual Service without API Consumption Settings and with Evaluate Policy Actions

Prerequisites for scenario B:

- *The asset should be an instance of the virtual type - Virtual Service, Virtual REST Service, Virtual XML Service.*
- *Make sure that the API consumption settings is not configured for the asset.*
- *An Evaluate * policy action is included in the asset's run-time configuration.*

➤ To register as a consumer of an asset for Scenario B:

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link that is located in the upper-left corner of the menu bar.
 - Click the **Search** icon that is located next to the **Scope** list. The default search scope is **Assets**.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the list of asset types, click **Choose**.
4. In the **Choose Asset Types** dialog box, follow these steps:
 - a. Click the chevron next to **Everything** option button.
 - b. In the list of asset types, select any of the **Virtual Service** type for whose instance you want to register the consumers.
 - c. Click **OK**.

This displays a list of the Virtual Service assets that were defined using the selected types, and that are available to you in the Search Results page.

5. Click a Virtual Service asset you want to consume.

This opens the Virtual Service Details page. Also, the actions bar displays a set of actions that are available for working with the displayed virtual service.

6. On the action bar of the Virtual Service Details page, click **Consume**.
7. In the **Consume Asset** dialog box, you can do one of the following:
 - Register an existing asset as consumer of the Virtual Service asset.

- Create a new Application asset and register the newly created Application asset as consumer of the Virtual Service asset. To do so, follow these steps:

1. Click the **Create Application asset** link.
2. Provide the appropriate information for each of the displayed data fields.

Note:

The list of identifiers is dynamically loaded depending upon the Evaluate policy actions that are defined for the Virtual Service asset.

3. Click **Create** to add the new Application asset to CentraSite.
8. To remove an asset that is currently selected for consumption, hover over the asset you want to remove. The **Delete** icon appears.

- a. Click **Delete**.

A confirmation message stating that the selected asset will be removed from the selection list appears.

- b. Click **Yes** to confirm.

9. In the **Reason for Request** field, type the reason for requesting consumption of the Virtual Service asset.

This comment is visible to all designated approvers who review the consumer registration request for the Virtual Service asset.

10. Click **Consume**.

If there are any approvals configured for the consumer registration process, then a consumer registration request is sent to the designated list of approvers. Otherwise, the selected consumers are immediately registered with the Virtual Service asset and the registration is reflected through the **Consumers** count in the **Basic Information** profile.

Scenario C: Virtual Service with API Consumption Settings and without Evaluate Policy Action

Pre-requisites for scenario C:

- *The asset is an instance of the virtual type - Virtual Service, Virtual REST Service, Virtual XML Service.*
- *Make sure that the API consumption settings is not configured for the asset.*
- *An Evaluate * policy action is not included in the asset's run-time configuration.*

➤ **To register as a consumer of an asset for Scenario C:**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link that is located in the upper-left corner of the menu bar.
 - Click the **Search** icon that is located next to the **Scope** list. The default search scope is **Assets**.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the list of asset types, click **Choose**.
4. In the **Choose Asset Types** dialog box, follow these steps:
 - a. Click the chevron next to **Everything** option button.
 - b. In the list of asset types, select any of the **Virtual Service** type for whose instance you want to register the consumers.
 - c. Click **OK**.

This displays a list of the Virtual Service assets that were defined using the selected types, and that are available to you in the Search Results page.

5. Click a Virtual Service asset you want to consume.

This opens the Virtual Service Details page. Also, the actions bar displays a set of actions that are available for working with the displayed virtual service.

6. On the actions bar of the Virtual Service Details page, click **Consume**.
7. In the **Consume API** dialog box, you can do one of the following:
 - Generate access tokens for an existing Consumer Application asset to make API calls on the Virtual Service asset as follows:
 1. Select the type of authentication token that you want to use to allow the consumer to access the Virtual Service asset.
 - **API Key** - CentraSite generates an API key (a base64-encoded string of the consumer-key:consumer-secret combination) to access and test the Virtual Service asset.
 - **OAuth2 Token** - CentraSite generates the OAuth2 client credentials (a client_id and client_secret) to further request an OAuth2 token to access and test the Virtual Service asset.
 2. Type the name of the Consumer Application asset in the **Consumer Application Name** field.

3. Select **Email me** to automatically generate an email notification about the usage of access token to the user at the email address provided at registration.
4. In the **Reason for Request** field, type the reason for requesting consumption of the Virtual Service asset.

This comment is visible to all designated approvers who review the consumer registration request for the Virtual Service asset.

5. Click **Consume** .

- Register an asset as consumer of the Virtual Service asset as follows:
 1. Click **Consume using other assets**.
 2. In the **Consume Asset** dialog box, you can do one of the following:
 - Register an existing asset as consumer of the Virtual Service asset.
 - Create a new asset in CentraSite and then register the newly created asset as consumer of the Virtual Service asset.

Scenario D: Virtual Service with API Consumption Settings and Evaluate Policy Action

Pre-requisites for scenario D:

- *The asset is an instance of the virtual type - Virtual Service, Virtual REST Service, Virtual XML Service.*
- Make sure that the API consumption settings *is configured* for the asset.
- An Evaluate * policy action *is included* in the asset's run-time configuration.

> To register as a consumer of an asset for Scenario D:

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link that is located in the upper-left corner of the menu bar.
 - Click the **Search** icon that is located next to the **Scope** list. The default search scope is **Assets**.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the list of asset types, click **Choose**.
4. In the **Choose Asset Types** dialog box, follow these steps:
 - a. Click the chevron next to **Everything** option button.

b. In the list of asset types, select any of the **Virtual Service** type for whose instance you want to register the consumers.

c. Click **OK**.

This displays a list of the Virtual Service assets that were defined using the selected types, and that are available to you in the Search Results page.

5. Click a Virtual Service asset you want to consume.

This opens the Virtual Service Details page. Also, the actions bar displays a set of actions that are available for working with the virtual service.

6. On the actions bar of the Virtual Service Details page, click **Consume**.

7. In the **Consume Asset** dialog box, you can do one of the following:

- Generate access tokens to make API calls on the Virtual Service asset.
 1. Provide the required information. The fields displayed are the same as for Scenario C, so for a description of the fields, follow the instructions provided for [Virtual Service with API Consumption Settings and without Evaluate Policy Action](#).
 2. Type the appropriate information for each of the displayed consumer identifier fields. CentraSite automatically populates the list of identifiers depending on the set of Evaluate policy actions that are defined for the Virtual Service asset.
- Register a Consumer Application asset or any asset as consumer of the Virtual Service asset as follows:
 1. Click **Consume using other assets**.
 2. In the **Consume Asset** dialog box, you can do one of the following:
 - Register an existing asset as consumer of the Virtual Service asset.
 - Create a new Application asset and register the newly created Application asset as consumer of the Virtual Service asset.

Viewing Consumer Registration Requests

Note:

Beginning with version 9.9, CentraSite no longer supports the concept of consumer registration requests. This means that there is no procedure for approving the pending consumer registration requests, and reviewing the status of consumer registration requests in both CentraSite Control and CentraSite Business UI.

The **Consumer Registrations** section that provides functionality for viewing a summary of all consumer registration requests in conjunction with **Pending Registrations** and **Registration Requests** is removed from the **Home > My CentraSite > Inbox** page of CentraSite Control.

The **My Pending Consumer Registration Requests** data feed that was used to construct a custom portlet for rendering the list of all consumer registration requests in CentraSite Business UI has been deprecated and will be removed in a future release.

However, if for example, you have an approval process initiated by a consumer registration policy and you are a designated approver, CentraSite places the consumer registration requests for your review and approval. You can review and accept these requests on the **My Pending Approvals** portlet of the Welcome page in CentraSite Business UI. Alternatively, you can review and accept these requests on the **Pending Approvals** tab of your Inbox page in CentraSite Control.

If you have made a register as a consumer request for an asset, on successful registration, you see the Consumers count incremented by one in the asset details page.

Important:

If you migrate CentraSite from a pre-9.9 version, the consumer registration requests that were pending for an approval by the asset owner in the previous version of CentraSite continues to remain in the new CentraSite Registry Repository. To fetch details of these migrated consumer registration requests, you must run the `list Pending Consumer Registrations` command in the command line interface of CentraSite.

Monitoring Consumer Count of an Asset

CentraSite helps monitor the registered consumers for each displayed asset.

The number of consumers is typically represented by a numerical prefix to the attribute **Consumers** in the description area of the Basic Information profile in the asset details page. For example, if you have 5 users registered as consumers for the displayed asset, then the **Basic Information** profile displays **5 Consumers**. If you do not have a registered consumer, then this attribute displays **0 Consumers**.

To see a list of the currently registered consumers for the displayed asset, click on the consumer count. In the displayed list, click the link of the consumer whose details you want to view. This shows the details of the consumer.

In a similar way, you can monitor the number of assets consumed by a particular consumer using the attribute **Consumed Assets** in the description area of the **Basic Information** profile in the consumer asset details page. For example, if this asset has been registered as a consumer of 5 other assets in the registry, then the **Basic Information** profile displays **5 Consumed Assets**. If this asset is not consuming any other asset, then this attribute displays **0 Consumed Assets**.

Modifying Consumer Details

Pre-requisites:

To modify an existing consumer, you must have the Modify Assets permission or at least the Modify instance-level permission on the specified asset or the consumer (which is represented by an individual user, group, or an asset).

> To modify consumer details

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link that is located in the upper-left corner of the menu bar.
 - Click the **Search** icon that is located next to the **Scope** list. The default search scope is **Assets**.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the list of asset types, click **Choose**.
4. In the **Choose Asset Types** dialog box, follow these steps:
 - a. Click the chevron next to **Everything** option button.
 - b. In the list of asset types, select any of the **Service** type for whose instance you want to modify the consumer details.
 - c. Click **OK**.

This displays a list of the Service assets that were defined using the selected types, and that are available to you in the Search Results page.

5. Click a Service asset that contains the required consumer.

This opens the Service Details page. Also, the actions bar displays a set of actions that are available for working with the Service asset.

6. In the **Basic Information** profile of the Service Details page, click the **Consumers** count.

This displays a list of all of the currently registered consumers for the Service asset.

7. Click the name of the consumer whose details you want to examine and modify.

This opens the Consumer Asset Details page. Also, the actions bar displays a set of actions that are available for working with the consumer asset.

8. On the actions bar of the Consumer Asset Details page, click **Edit**.

9. Modify the values for the consumer's fields as required.

10. Click **Save** to update the consumer information.

Unregistering Existing Consumer

To unregister an existing consumer, you must have the Modify Assets permission or at least the Modify instance-level permission on the specified asset or the consumer (which is represented by an individual user, group, or an asset).

Important:

To unregister an existing consumer, the following conditions must be satisfied:

- The user, group, or asset should be currently registered as consumer for the specified asset.
- You are the owner of the asset from that you want to unregister a consumer.

You can unregister an existing consumer using the Asset Details page.

You might consider unregistering a consumer if you want to:

- Suspend consumption of a particular asset (either on a temporary or permanent basis.)
- Delete an asset permanently from the CentraSite registry.

Note:

You can restore the functionality of consumption by re-registering the user or asset with the same asset at any time.

The consequences of unregistering an existing consumer on an asset are as follows:

- The count of consumers in the **Basic Information** profile of the asset is updated.
- Some aspects of runtime such as invocations, performance metrics and other events relating to the operation of a virtual API asset running in Mediator gateway may not work as desired.
- The user, group, or asset should be currently registered as consumer for the specified asset.
- You are the owner of the asset from that you want to unregister the consumer.
- You belong to a role that includes the Modify Assets permission for organization in which the asset resides.
- You have been assigned at least the Modify instance-level permission on the specified asset or the consumer (which is represented by an individual user, group, or an asset).

You can unregister an existing consumer in CentraSite Business UI in the following ways:

- From the Consumer Asset Details page
- From the Consumed Asset Details page

➤ **To unregister an existing consumer from the consumer asset details page**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link that is located in the upper-left corner of the menu bar.
 - Click the **Search** icon that is located next to the **Scope** list. The default search scope is **Assets**.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the list of asset types, click **Choose**.

4. In the **Choose Asset Types** dialog box, follow these steps:
 - a. Click the chevron next to **Everything** option button.
 - b. In the list of asset types, select any of the **Service** type for whose instance you want to unregister the consumer.
 - c. Click **OK**.

This displays a list of the Service assets that were defined using the selected types, and that are available to you in the Search Results page.

5. To unregister a consumer, do one of the following:

- Click the consumer asset that you want to unregister.

This opens the Consumer Asset Details page. Also, the actions bar displays a set of actions that are available for working with the asset.

1. Click the **Consumed Assets** count.

The number of assets consumed by this consumer is typically represented by a numerical prefix to the attribute **Consumed Assets** in the description area of the **Basic Information** profile, for example, **3 Consumed Assets**.

2. In the displayed list, move the pointer over the asset from which you want to unregister this consumer. This causes a **Delete** icon to appear, that you can use for unregistering.
3. Click **Delete**.
4. Click **Yes** in the confirmation dialog box to proceed with unregistration.

Note:

If you unregister an Application asset, you must manually republish the asset to Mediator gateway to put the changes into effect.

- Click the asset that contains the required consumer.

This opens the Consumed Asset Details page. Also, the actions bar displays a set of actions that are available for working with the asset.

1. Click the **Consumers** count.

The number of assets consuming the displayed asset is typically represented by a numerical prefix to the attribute **Consumers** in the description area of the **Basic Information** profile, for example, **3 Consumers**.

2. In the displayed list, hover over the consumer you want to unregister. This causes a **Delete** icon to appear, that you can use for unregistering.
3. Click **Delete**.

4. Click **Yes** in the confirmation dialog box to proceed with unregistration.

Note:

If you unregister an Application asset, you must manually republish the asset to Mediator gateway to put the changes into effect.

Synchronization of Gateway Application from API Gateway to CentraSite

Important:

The service should be an instance of the virtual type - Virtual Service, Virtual REST Service, and Virtual OData Service.

You can create a service in CentraSite and deploy it to API Gateway. In API Gateway, applications can be added to the service to consume it. In CentraSite, the application that consumes the service is persisted as **Gateway Application** asset. The **Gateway Application** asset gets registered as a consumer in the Service Details page of the CentraSite.

The scheduler that runs in the CentraSite fetches the gateway application details from API Gateway based on the interval that you have specified in the API Gateway synchronization settings. When you modify the application details or delete the application in API Gateway, the scheduler synchronizes those changes back to the corresponding gateway application in CentraSite. For more information on configuring synchronization settings, refer to ["Configuring the API Gateway Synchronization Settings" on page 1478](#).

Viewing Gateway Application Details

To view gateway application as a consumer from the Service Details page:

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link that is located in the upper-left corner of the menu bar.
 - Click the **Search** icon that is located next to the **Scope** list. The default search scope is **Assets**.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the list of asset types, click **Choose**.
4. In the **Choose Asset Types** dialog box, follow these steps:
 - a. Click the chevron next to **Everything** option button.
 - b. In the list of asset types, select any of the **Service** type for the instance that you want to view the gateway application consumer details.
 - c. Click **OK**.

This displays a list of the Service assets that were defined using the selected types, and that are available to you in the Search Results page.

5. Click a Service asset that contains the gateway application consumer.

This opens the Service Details page.

6. In the **Basic Information** profile of the Service Details page, click the **Consumers** count.

This displays a list of currently registered consumers such as Gateway Application for the service asset.

Note:

As API Gateway is the source of truth for gateway application, the gateway application is set as read-only. Hence, you cannot **Revoke** or **Delete** the application from CentraSite.

7. Click the name of the gateway application consumer whose details you want to examine .

This opens the Gateway Application Details page. Also, the actions bar displays a set of actions that are available for working with the gateway application asset.

The following table describes the profiles that are available in Gateway Application Details page.

Profile	Description
Basic Information	Displays the general information about the asset, such as the asset's version, last modified date, asset type, owning organization, owning user, a description of the asset, and the number of watchers, consumers, consumed assets, and pending approvals.
Gateway Details	<p>Displays the following details of the API Gateway instance(s) from which the application is registered as consumers:</p> <ul style="list-style-type: none"> ■ Gateways: Lists the name of the API Gateway instance along with a deep link to open the API gateway instance in the CentraSite user interface. ■ For APIs: Lists the name of the services that consumes this gateway application along with the deep link to open the Service Details page. ■ View Application in API Gateway: Lists the API Gateway instances and its corresponding link. Click this link to view the Application page directly in the API Gateway UI.

Searching Gateway Application

> To search gateway application

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets for which you have View permission in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the gateway application, enter the asset type as *Gateway Application* and click



This displays a list of gateway application asset types for which you have View permission in the Search Results page.

4. Click the name of the gateway application whose details you want to examine .

This opens the Gateway Application Details page. Also, the actions bar displays a set of actions that are available for working with the gateway application asset.

Deleting Gateway Application

To delete the gateway application from CentraSite, either you have to

- Delete or unregister the application in API Gateway . As and when the scheduler runs the gateway application is removed from CentraSite.
- **Unpublish** the service that consumes the gateway application.
- **Unpublish** the API Gateway instance.

Note:

When you **Publish** the service to the Mediator gateway, the gateway application details will not be published.

Synchronizing Consumer Applications through CentraSite Business UI

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if

required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

Pre-requisites:

To synchronize consumer applications in CentraSite with Mediator, you must have the following roles:

- **CentraSite Administrator:** to synchronize consumers of any organization with an instance of Mediator belonging to any organization.
- **Organization Administrator:** to synchronize consumers with an instance of Mediator to which your Manage Organizations permission applies.
- **Mediator Administrator:** to synchronize consumers with an instance of Mediator belonging to the organization to which your Mediator Administrator role applies.
- **Mediator Publisher:** to synchronize consumers with an instance of Mediator belonging to the organization to which your Mediator Publisher role applies.

You synchronize, also termed publish, consumer applications to a policy enforcement point (PEP) such as webMethods Mediator to perform client validation.

You synchronize consumer applications in CentraSite with Mediator for the following reasons:

- To update any changes, for example, modification to the range of valid IP addresses, that was performed in the details page of a consumer application in CentraSite.
- To update the list of consumers in Mediator when a consumer application is deleted in CentraSite.
- To update the list of registered consumers in Mediator when a consumer application is unregistered from the API in CentraSite.

Note:

When synchronizing consumer applications with Mediator, if the application has an API key or OAuth 2.0 token, then CentraSite does not allow the synchronization of that particular application with Mediator. In this case, you have to manually republish the application that includes the API key or OAuth 2.0 token to Mediator.

➤ To synchronize consumer applications

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. To filter the list of the available Mediator gateways, do the following:
 - a. Go to the advanced search panel.
 - b. In the **Narrow Your Results** section, do the following:
 - a. Locate **Applicable Scopes**.

- b. Select **Mediator** from the drop-down list.

CentraSite displays the list of available Mediator gateways in the Search Results page. Also, the actions bar displays a set of actions that are available for working with the displayed **Mediator** gateways.

3. Select the Mediator gateways to which you want to synchronize the consumers.

4. On the actions bar of the Search Results page, click **Sync Consumers** ().

You can also synchronize consumers to a single Mediator gateway through the **Sync Consumers** actions in the Mediator Gateway Details page.

The deployment process is carried out by a synchronous mechanism between CentraSite and Mediator:

- a. CentraSite invokes the Mediator's deployer service and pushes the consumer applications that are ready for deployment to the Mediator.
- b. Instantly, Mediator deploys the consumer applications that were received from CentraSite and notifies CentraSite when the deployment process is complete.

Synchronizing Consumer Applications through CentraSite Control

Pre-requisites:

To synchronize consumer applications in CentraSite with Mediator, you must have the following roles:

- **CentraSite Administrator:** to synchronize consumers of any organization with an instance of Mediator belonging to any organization.
- **Organization Administrator:** to synchronize consumers with an instance of Mediator to which your Manage Organizations permission applies.
- **Mediator Administrator:** to synchronize consumers with an instance of Mediator belonging to the organization to which your Mediator Administrator role applies.
- **Mediator Publisher:** to synchronize consumers with an instance of Mediator belonging to the organization to which your Mediator Publisher role applies.

You synchronize, also termed deploy, consumer applications to a policy enforcement point (PEP) such as webMethods Mediator to perform client validation.

> To synchronize consumer applications

1. In CentraSite Control, go to **Operations > Deployment**.

2. On the **Deploy Consumers** tab, click **Synchronize**.
3. In the **Select a Target and Consumer Applications to be Deployed** dialog box, perform a keyword or advanced search to display the list of consumer applications and targets that are ready for deployment.
4. Click **OK**.

The deployment process is carried out by a synchronous mechanism between CentraSite and Mediator:

- a. CentraSite invokes the Mediator's deployer service and pushes the consumer applications that are ready for deployment to the Mediator.
- b. Instantly, Mediator deploys the consumer applications that were received from CentraSite and notifies CentraSite when the deployment process is complete.

The **Deploy Consumers** tab includes the following information:

Column	Description						
Pending Changes	Icons indicating the deployment status of the consumer applications.						
	<table border="1"> <thead> <tr> <th>Icon</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>The consumer application is deployed to the target.</td> </tr> <tr> <td></td> <td>The consumer application is pending deployment to the target.</td> </tr> </tbody> </table>	Icon	Description		The consumer application is deployed to the target.		The consumer application is pending deployment to the target.
Icon	Description						
	The consumer application is deployed to the target.						
	The consumer application is pending deployment to the target.						
Rule ID	The synchronization rule ID of the target (that is, webMethods Mediator or Insight).						
Target	The name of the target on which the consumer application is deployed.						
Last Sync Date	The date and time that the deployment occurred.						
Status	The deployment status of the consumer application (for example, Deployed or Failed).						

Important:

If the status shown in the **Status** column does not automatically switch to **Deployed** or **Failed**, click the **Refresh** button to determine whether CentraSite was able to deploy the consumer applications successfully. If the deployment process failed, identify and correct the error and then try deploying the consumer application again.

Managing Consumers through Command Line Interface

This section describes operations you can perform to manage consumers and consumer registration requests through the CentraSite Command Line Interface (CLI).

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

Fetching Details of Migrated Pending Consumer Registrations

Pre-requisites:

To fetch details of migrated consumer registration requests through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

If you migrate CentraSite from a pre-9.9 version, the consumer registration requests that were pending for an approval by the asset owner in the previous version of CentraSite will continue to remain in the new CentraSite Registry Repository.

Once you have the list of pending consumer registrations from the old Registry Repository, you can do one of the following:

- Issue email notifications to specified groups or individuals requesting them to register as consumers for the specified assets.
- Manually register the individual consumers for the specified assets.

CentraSite provides a command tool named `list Pending Consumer Registrations` for this purpose.

➤ To fetch details of pending consumer registrations

- Run the command `list Pending Consumer Registrations`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list Pending Consumer Registrations -user <USER-ID> - password <PASSWORD> [-url <CENTRASITE-URL>] -file <LOG-FILE>`

Note:

This command will not synchronize consumer application, if the CentraSite run-time aspects are not enabled. By default, run-time policies configured from CentraSite are disabled. To enable the CentraSite run-time policies, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

The input parameters are:

Parameter	Description
CENTRASITE-URLurl	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .

Parameter	Description
PASSWORD	The password for the CentraSite user identified by the parameter USER-ID.
LOG-FILE	Name of an output file that will contain the migrated consumer registration details.

The command response is of the following syntax:

```
Requested By:
<Name={Name_of_the_Requestor}> <{UDDI_Key_of_the_Requestor}> <Organization=
{Organization_to_which_Requestor_belongs}>
```

```
For the Asset:
<Type={Type_of_the_consumed_asset}> <Name={Name_of_the_consumed_asset}>
<{UDDI_key_of_the_consumed_asset}>
<Organization={Submitting_organization_of_the_consumed_asset}>
<Version={version_of_the_consumed_asset}>
```

```
Has Consumers:
<Type={Type_of_the_consuming_asset}> <Name={Name_of_the_consuming_asset}>
<Key={UDDI_key_of_the_consuming_asset}>
<Organization={Submitting_organization_of_the_consuming_asset}>
<Version={version_of_the_consuming_asset}>
...
```

Example (all in one line):

The command for viewing the pending consumer registration requests with an administrator account INTERNAL\Admin and password AdminPW would be:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd list Pending Consumer
Registrations -user INTERNAL\Admin -password AdminPW -url
http://localhost:53307/CentraSite/CentraSite -file c:\temp
\PendingConsumerRegistrations.txt
```

The response to this command could be:

```
Executing the command: list Pending Consumer Registrations
=====
Details of the 'Pending' Consumer Registration Requests
=====
Requested By:
<Name=INTERNAL\Cameron> <Key=uddi:c89dcf73-f25d-11e4-8fd1-b6bf1262f3c2>
<Organization=Default Organization>
For the Asset:
<Type=Virtual Service> <Name=RealTimeStockQuotes_VS>
<Key=uddi:5f8b0a0c-f26a-11e4-b8e7-849ec8059cef>
<Organization=Default Organization> <Version=1.0>
Has Consumers:
<Type=Application> <Name=Cam's Application 3>
<Key=uddi:ea1eeb10-f26f-11e4-b8e7-86fdbb748830>
<Organization=Default Organization> <Version=1.0>
...
Successfully executed the command: list Pending Consumer Registrations
```

Purging Consumer Registration Requests

Pre-requisites:

To purge consumer registration requests through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

In some circumstances, you may not be able to delete a Virtual Service asset because there could be one or more consumer registration requests that are internally related to the asset. You can purge such consumer registration requests if their status is Approved or Rejected.

CentraSite provides a Java tool named `PurgeConsumerRegistrationRequests.jar` for this purpose. This command tool purges all consumer registration requests that are related to a specified asset or all consumer registration requests that are available in the CentraSite registry.

- Run the Java tool `PurgeConsumerRegistrationRequests.jar`.

The syntax is of the format:

- `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd
PurgeConsumerRegistrationRequests.jar <CentraSite URL> <admin user id> <password>`
- `C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd
PurgeConsumerRegistrationRequests.jar <CentraSite URL> <admin user id> <password>
<objectID>`

The input parameters are:

Parameter	Description
CentraSite URL	The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
admin user id	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
password	The password for the CentraSite user identified by the parameter <code>admin user id</code> .
objectID	The ID of the asset whose related consumer registration requests you want to delete.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteToolbox.cmd
PurgeConsumerRegistrationRequests.jar http://localhost:53307/CentraSite/CentraSite
DOMAIN\admin pAsSw0rD
```

Synchronizing Consumer Applications

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

Pre-requisites:

To synchronize consumer applications in Mediator gateway through the Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `sync consumers` for this purpose.

➤ To synchronize consumer applications in Mediator gateway

- Run the command `sync consumers`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd sync consumers [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -gateway <GATEWAY>`

The input parameters are:

Input Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password of the user identified by the parameter <code>USER-ID</code> .
GATEWAY	The gateway on which to synchronize the consumer applications.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd sync consumers -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-gateway Gateway1
```

Assigning Consumer Association with a Virtual Service

Pre-requisites:

To associate an Application asset (represented as a consumer application) with a Virtual Service asset through the CentraSite Command Line Interface (CLI), you must have the CentraSite Administrator role.

When a virtual service associated with a consumer application is imported from a source instance to a target instance of CentraSite, the imported virtual service does not include the associated consumer application in the target instance.

CentraSite provides a command tool named `set Consumer` for this purpose. This tool associates a consumer application with a virtual service.

➤ To associate a consumer application

- Run the command `set Consumer`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd | sh set Consumer [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -application <CONSUMER-APPLICATION> -service <VIRTUAL-SERVICE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
CONSUMER-APPLICATION	The name or key of an Application asset (consumer application) you want to associate with a virtual service.
VIRTUAL-SERVICE	The name or key of the virtual service to which you want to associate the consumer application identified by the parameter <code>CONSUMER-APPLICATION</code> .

Example (all in one line):

The command for associating a consumer application with a virtual service with an administrator account `INTERNAL\Admin` and password `AdminPW` would be:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set Consumer -user  
INTERNAL\Admin -password AdminPW -url http://localhost:53307/CentraSite/CentraSite  
-application ConsumerApplication -service SampleVirtualService
```

Removing Consumer Association from a Virtual Service

Pre-requisites:

To remove an existing association between an Application asset (represented as a consumer application) and a Virtual Service asset through the CentraSite Command Line Interface (CLI), you must have the CentraSite Administrator role.

CentraSite provides a command tool named `reset Consumer` for this purpose. This tool removes a Consumer association that is existing between a consumer application and a virtual service.

> To remove a consumer association

- Run the command `reset Consumer`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd | sh reset Consumer [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -application <CONSUMER-APPLICATION> -service <VIRTUAL-SERVICE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
CONSUMER-APPLICATION	The name or key of an Application asset (consumer application) you want to disassociate from a virtual service.
VIRTUAL-SERVICE	The name or key of a virtual service from which you want to disassociate the consumer application identified by the parameter <code>CONSUMER-APPLICATION</code> .

Example (all in one line):

The command for removing an association that exists between a consumer application and a virtual service with an administrator account `INTERNAL\Admin` and password `AdminPW` would be:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd reset Consumer -user
INTERNAL\Admin -password AdminPW -url http://localhost:53307/CentraSite/CentraSite
-application ConsumerApplication -service SampleVirtualService
```

Reassigning Consumer Association to a Virtual Service

Pre-requisites:

To reassign an Application, API-Key, or OAuth2 Client asset (represented as a consumer application) to a Virtual Service asset through the CentraSite Command Line Interface (CLI), you must have the CentraSite Administrator role.

When a virtual service associated with an instance of the asset type Application, API-Key, or OAuth2 Client is imported from a source instance to a target instance of CentraSite, the imported virtual service does not include the associated asset in the target instance.

CentraSite provides a command tool named `reassign Consumer` for this purpose. This tool reassigns an Application, API-Key, or OAuth2 Client asset from one virtual service to another service. During

the process, an association of the API key or OAuth2 Client token of the consumer application is moved from one virtual service to another.

> To reassign a consumer association

- Run the command `reassign Consumer`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd | sh reassign Consumer [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -application <CONSUMER-APPLICATION> -service <VIRTUAL-SERVICE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .
CONSUMER-APPLICATION	The name or key of an Application, API-Key, or OAuth2 Client asset (consumer application) you want to reassociate with a virtual service.
VIRTUAL-SERVICE	The name or key of the virtual service to which you want to reassociate the consumer application identified by the parameter <code>CONSUMER-APPLICATION</code> .

Example (all in one line):

The command for reassigning an Application, API-Key, or OAuth2 Client asset with a virtual service with an administrator account `INTERNAL\Admin` and password `AdminPW` would be:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd reassign Consumer -user  
INTERNAL\Admin -password AdminPW -url http://localhost:53307/CentraSite/CentraSite  
-application MyAPIKey -service SampleVirtualService-2
```

Access Token Management

CentraSite provides a simple token-based authentication for providers to flexibly secure applications (APIs) and for consumers to easily access the data of the secured applications. The token-based authentication secures an application based on a security token that is generated for the user on authentication and then stores the token secure on the client.

CentraSite supports two types of token-based authentication:

- **API Keys:** The type of access key authorization grant that Mediator supports is *API Keys*. API keys are used as an authorization grant when the client is requesting API to protected resources based on an authorization previously arranged with the authorization server. That is, the client application gains authorization when it successfully registers with CentraSite as a consumer.

The API Provider (and users with an instance-level Modify permission on an API at a minimum) can enforce API key authentication by configuring the API Asset Details page. Such users can configure the following characteristics about client requests for API keys:

- Specify the approval requirements for clients requesting API keys.
 - You can specify that requests must be approved by approver groups of your choosing, or you can specify that requests will be automatically approved.
- Configure email messages to be sent to:
 - The approver groups when requests are submitted for approval.
 - The clients to inform them of their approval status.
- Specify the expiration of the API key.

Clients that want to use the API key to call (consume) an API in CentraSite must:

- Register as a consumer for the API.
 - When the client registration request is approved, the client receives an API key (a base64-encoded string of the consumer-key:consumer-secret combination). It works for both SOAP and REST calls.
- To call the API, the client must pass the API key in an HTTP request header or as a query string parameter. The use of this key establishes the client's identity and authentication.
- **OAuth 2.0 Tokens:** The type of OAuth2 authorization grant that Mediator supports is *OAuth 2.0 Client Credentials*. Client credentials are used as an authorization grant when the client is requesting API to protected resources based on an authorization previously arranged with the authorization server. That is, the client application gains authorization when it successfully registers with CentraSite as a consumer.

The API Provider (and users with an instance-level Modify permission on an API at a minimum) can enforce OAuth 2.0 authentication by configuring the API Asset Details page. Such users can configure the following characteristics about the approval process of granting OAuth2 client credentials:

- Specify the approval requirements for client requests for client credentials.
 - You can specify that requests must be approved by approver groups of your choosing, or you can specify that requests will be automatically approved.
- Configure email messages to be sent to:
 - The approver groups when requests are submitted for approval.
 - The clients to inform them of their approval status.

Clients that want to use the OAuth2 protocol to call APIs in CentraSite must:

- Register as a consumer for the API.

When the client registration request is approved, the client receives client credentials (a `client_id` and `client_secret`).

- Request an OAuth2 access token by passing the client credentials to the Mediator-hosted REST service `mediator.oauth2.getOAuth2AccessToken`. This service will provide an OAuth2 access token to the client.
- To call the API, the client must pass their OAuth access token in an HTTP request header.

An OAuth2 token is a unique token that a client uses to invoke APIs using the OAuth 2.0 protocol. The token contains an identifier that uniquely identifies the client. The use of a token establishes the client's identity, and is used for both the authentication and authorization.

Note:

Instructions throughout the remainder of this section use the term *access tokens* when referring to API keys and OAuth 2.0 tokens in general.

Access Token Request for an API Through API Portal Gateway

An API Provider restricts the access to an API by enforcing the appropriate access tokens. If the API that is exposed in an API Portal gateway enforces an access token, any user who requests access to the data of the exposed API gets an option to request for the access token of one of the type - API key or OAuth 2.0.

The access token request for an API through the API Portal gateway is a three step process in CentraSite.

1. **Client creation process:** Whenever a client requests an access token for an API in API Portal, CentraSite receives the request for the API access token, and processes the request. CentraSite checks if the client who made the access token request already exists in the CentraSite registry. If the client already exists in the registry, then CentraSite generates an access token entry in the registry. However, if the client does not exist in the registry, CentraSite performs the client creation process. During this process CentraSite registers the client as a member of the consumer organization configured for the registered API Portal in the CentraSite registry.
2. **Access token generation process:** After a client (API Portal user) is successfully created in the registry, CentraSite generates the access token and usage details for the API.

If an approval process is configured for access token generation, CentraSite initiates the approval process and submits the client's request to the designated group of approvers. Approvers receive the approval request in the Pending Approval Requests of the API details page. Approvers whose user account includes a valid email address also receive an email message informing them that a request is awaiting for their approval. CentraSite does not execute the client's requested operation until it obtains the necessary approvals. If an approver rejects the request, the requested access token is not generated.

3. **Notification process:** In the event of a success of the access token request, CentraSite returns a success message with details about the newly generated access token to API Portal, and notifies the client (including data that is pertinent to the access token validity and usage of the API) through email. In the event of a failure of the access token request, CentraSite notifies the client about the failed request.

Similarly, when those clients subsequently request for renewal or revocation of the access key, CentraSite verifies the client credentials, performs the requested operation, and notifies the API Portal and client.

Points to keep in mind when API Portal is used with CentraSite:

- When a client requests for an access token from API Portal, CentraSite generates an User object entry in the registry that describes the client, and then stores the user entry in the repository. This user is not allowed to log into CentraSite or perform any operation in CentraSite.
- CentraSite automatically associates the users with the API Portal's **Consumer Organization**. This **Consumer Organization** property, which is configured during the registration of an API Portal with CentraSite, specifies the organization to which the new user is added.
- The consumer organization owns the users from an API Portal. You cannot delete this consumer organization, unless you belong to a CentraSite Administrator role.
- You cannot delete an API Portal user from the registry, unless you belong to a CentraSite Administrator role.
- If you are the owner of the API asset or the access token itself, you have permission to renew and revoke access token that is available to you. If you are the CentraSite Administrator, you have the permission to renew and revoke any access token on the CentraSite server.

Managing Access Tokens through CentraSite Business UI

This section describes operations you can perform to manage access tokens through CentraSite Business UI.

Note:

You cannot manage access tokens from CentraSite, if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

The API Consumption Model Using API Keys

To enable a consumer to access and use an API using API access key, the following events must occur:

1. The consumer sends a request to consume an API. The request must include the consumer's authentication credentials.
2. CentraSite generates the API key for consumption of the API (the specific key generation steps depend on the configuration settings defined by the Provider (owner) of the API). Later,

CentraSite prepares the API for publishing and invokes the API Key Generation policy on the Mediator.

3. The API Key Generation policy publishes the API key to Mediator.
4. If publish of the API key is successful, the API Key Generation policy returns a success message with details including API Key, Expiration Date, and Usage Notes of the access key for consuming the API. If publish of the API key is unsuccessful, the deployer service returns a failure message.
5. The consumer accesses the URL for API consumption, sends the API key as an integral part of the HTTP/SOAP request header or as a query string in the URI, and upon validation of the API key consumes the API.
6. If the consumption is successful, the consumer uses the API. If the consumption is unsuccessful for some reasons of authorization, a 500 fault is returned.

The API Consumption Model Using OAuth 2.0 Tokens

To enable a consumer to access and use an API using OAuth 2.0 access token, the following events must occur:

1. The consumer sends a request to consume an API. The request must include the consumer's authentication credentials.
2. CentraSite generates the OAuth 2 Client Id and Client Secret for consumption of the API (the specific OAuth 2.0 token generation steps depend on the configuration settings defined by the Provider (owner) of the API). Later, CentraSite prepares the API for publishing and invokes the OAuth 2 Client Generation policy in Mediator.
3. The OAuth 2 Client Generation policy publishes the OAuth 2.0 token to Mediator.
4. If publish of the OAuth 2.0 token is successful, the OAuth 2 Client Generation policy returns a success message with details including OAuth 2 Client Id, Client Secret, and URL to obtain the access token for consuming the API. If publish of the OAuth 2.0 token is unsuccessful, the deployer service returns a failure message.
5. CentraSite generates the OAuth 2.0 token using the OAuth 2 Client Id, Client Secret, and access token URL.
6. The consumer accesses the URL for API consumption, sends the OAuth 2.0 token as an integral part of the HTTP/SOAP request header, and upon validation of the OAuth 2.0 token consumes the API.
7. If the consumption is successful, the consumer uses the API. If the consumption is unsuccessful for some reasons of authorization, a 500 fault is returned.

Mediator Evaluating Consumers at Run-Time

After you have successfully registered as a consumer for a particular API, in order to call an API you must provide your API key or OAuth2 access token in your HTTP request header.

- If you use an API key to call the API, the client must provide the API key in the HTTP request header or as a query string parameter. The use of this key establishes the client's identity and authentication.
- If you use an OAuth2 access token to call the API, the client must provide the OAuth2 access token as an integral part of the HTTP request header. An OAuth2 token is a unique token that a client uses to invoke APIs using the OAuth 2.0 protocol. The token contains an identifier that uniquely identifies the client. The use of a token establishes the client's identity and is used for both the authentication and authorization.

In addition, the API provider can include run-time security actions in the run-time governance rules for APIs. Security actions can validate clients' request and response messages (through WSS X.509 certificates, WSS username tokens, and so on) or identify clients (through IP address or hostname). To enforce client validation, Mediator maintains a list of consumer applications specified in CentraSite that are authorized to access the API published to Mediator.

Fetching and Using API Access Keys for Consumption

APIs are often exposed over the open internet for consumption. API Providers use API access keys as authentication tokens to prevent unauthorized access to an API.

CentraSite automatically generates the API keys when consumers request for consumption of the secured APIs. The API Providers can view, approve, and set expiration for the generated API keys. This ensures that no consumer can access a protected API without a valid key.

API access keys are verified at run-time to ensure that:

- The access key presented is valid and has not expired.
- The access key passed as a parameter in the URI or the HTTP/SOAP request for an API is approved to consume the API.

If an API is configured for the API key authentication, and you have successfully registered as a consumer for that particular API, then you would receive your API access key details through an email message.

Using the Generated API Key

CentraSite allows you to set API keys as part of the HTTP header for a REST API, the SOAP header for a SOAP API, or as the query component of a request URI.

Important:

In the case where a consumer is sending a request with both credentials (HTTP/SOAP header) and (query string), the HTTP/SOAP header takes precedence over the query string when the Mediator is determining which credentials it should use for the consumption.

Request Header

The API keys are passed as the HTTP/SOAP header component of an API consumption request. The HTTP/SOAP header corresponds to an array of header names to include for that particular API consumption.

The following example demonstrates a typical HTTP/SOAP request with API keys that form the header value of the API Access URL:

```
x-CentraSite-APIKey:a4b5d569-2450-11e3-b3fc-b5a70ab4288a
```

Query String

Note:

Query string is only applicable for REST APIs.

The API keys are passed as the query component of a REST API consumption request.

The following example demonstrates a typical HTTP GET request with API keys that form a query string of the API Access URL:

```
http://localhost:5555/ws/RestAPI?APIKey=a4b5d569-2450-11e3-b3fc-b5a70ab4288a
```

Notice that the API keys are added to the path after a “?” and specified as key-value pair.

When you request a REST API for consumption using the Access URL and the generated API key, CentraSite automatically validates the API's run-time actions to ensure that any "Evaluate" action that appears in the policy governance rule of an API is validated with the consumer requesting for the API.

Example:

If a REST API consumption encounters a problem due to one or more of the following reasons, a 500 status code is returned:

- If the API key value in the HTTP header or the query string is authenticated as invalid.

The sample message looks like this:

```
The request is authenticated as invalid.
```

- If the HTTP header is not present in the request.

The sample message looks like this:

```
A required header is missing in the request.
```

- If the API key value in the HTTP header is expired.

The sample message looks like this:

```
The API key has expired.
```

Fetching and Using OAuth 2.0 Client Tokens for Consumption

If you are using the OAuth 2.0 protocol and you have successfully registered as a consumer for an API, you should have received your OAuth 2.0 client credentials (a `client_id` and `client_secret`).

Now you need to obtain an OAuth 2.0 access token by providing your client credentials to the Mediator-hosted REST service `mediator.OAuth 2.0.getOAuth 2.0AccessToken`. This service will

provide an OAuth 2.0 access token that you can subsequently include in your requests to call the API.

The service's input parameters are:

- `client_id`
- `client_secret`
- `scope` (optional). The scope value is the name of the virtual service. If the scope value is valid, Mediator obtains the access token. If no scope value is provided, Mediator provides the access token to the scope in which the client is allowed and adds the scope to the response. To provide the scope, include it in the request body.

Ways for Clients to Provide the Inputs

There are three ways in which a client can provide the inputs for this service:

- Provide inputs in the Basic authentication header (recommended).

The client can provide the client credentials (`client_id` and `client_secret`) in the Authorization header using the following form:

```
Authorization: Basic <base-64-encoded client_id:password, client_secret>
```

If you want to provide the scope, include it in the request body.

- Provide JSON inputs for the service.

The client can send a JSON request to the service in the following form:

```
{
  "client_id" : "",
  "client_secret": "",
  "scope" : ""
}
```

Note:

The client should contain the header `Content-type:application/json` in the request.

- Provide inputs in the request body

The OAuth 2.0 specifications do not support sending the client credentials over the URL as URL-Encoded. However, you can send the client credentials in the request body using the following form:

```
client_id=<client_id>&client_secret=<client_secret>&scope=<scope>
```

Note:

- The client should contain the header `Content-type:application/x-www-form-urlencoded` in the request.
- If a client provides the `client_id` and `client_secret` in both the Authorization header and the request body, the credentials given in the Authorization header are used.

Using HTTPS for Granting Access Tokens

For security reasons Software AG recommends using HTTPS in your production environment. If you are using HTTPS as the transport protocol over which the OAuth 2.0 access tokens are granted authorization, you must set the parameters `pg.OAuth 2.0.isHTTPS` and `pg.OAuth 2.0.ports` as described in *Administering webMethods Mediator*.

Responses Returned to Clients

Following are sample responses that are returned to the client:

- Sample XML response:

```
<Response
xmlns="https://localhost/rest/pub.mediator.OAuth 2.0.getOAuth 2.0AccessToken">
<access_token>db95b40095f31439a1cd8f411e64abe8</access_token>
<expires_in>3600</expires_in>
<token_type>Bearer</token_type>
</Response>
```

- Sample JSON response:

```
{
"access_token": "db95b40095f31439a1cd8f411e64abe8",
"token_type": "Bearer",
"expires_in": 3600
}
```

Viewing Access Token Details

API keys and OAuth 2.0 access tokens allow you to use an API published in CentraSite or API Portal gateway, and grant access to data which is restricted or secured to your user account.

An API key or OAuth access token acts as a type of *secret key* or *secret token*. As long as the consumer is in possession of this access token, the consumer can access both the publicly available and secured data of the API. A provider can revoke this access token at any time. Furthermore, all access tokens expire after the specified days. Once the access token is revoked or has expired, the user can access only the publicly available data.

In CentraSite, the default expiry limit for API keys is unlimited.

You can view details of API keys and OAuth 2.0 access tokens in the following ways:

- Through the email notification messages that were auto-generated by CentraSite.
- Through the Virtual Service Details page.
- Through the User Preferences page.

Viewing Access Token Details Through Email Notifications

Once your API registration request is approved, CentraSite sends an automated email message containing details of the API key value and its usage to both the approver and consumer.

If you want to receive email messages of the access tokens, make sure:

- You have the notification option set as **Email** in the User Preferences page.
- You have specified a valid email address.
- Access your mailbox.

The information contained in the email message depends on whether you access the asset as a provider or consumer.

- If you are the provider of the asset, you see the following information:

API Key Details

- API key
- Expiration date
- API Usage Information (If provided)

OAuth 2.0 Token Details

- OAuth 2 Client ID
- Client Secret
- The URLs to obtain the access tokens
- Expiration date
- API Usage Information (If provided)

- If you are the consumer of the asset, you see the following information:

API Key Details

- API key
- Expiration date
- API Usage Information (If provided)

OAuth 2.0 Token Details

- OAuth 2 Client ID
- Client Secret
- The URLs to obtain the access tokens
- Expiration date
- API Usage Information (If provided)

Viewing Access Token Details Through Service Details Page

> To view the details of an access token

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.This displays a list of assets for which you have View permission in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of any type of Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the list of asset types, select the required type of Service.
 - c. Click **OK**.
5. In the displayed list of Service assets, click the asset for which you want to display the access token details.

This opens the Service Details page.

6. Select the **Consumer Overview** profile.

You can also view the access token details using the **Consumers** attribute in the **Basic Information** profile. You can click on the count to show the consumers of the asset and navigate to the details page of the access token.

The information contained in the details page of the asset depends on whether you access the asset as a provider or consumer.

- If you are the provider of the asset, you see the following information:

API Key Details

- API key
- Expiration date
- API Usage Information (If provided)

OAuth 2.0 Token Details

- OAuth 2 Client ID
- Client Secret
- The URLs to obtain the access tokens
- Expiration date
- API Usage Information (If provided)
- If you are the consumer of the asset, you see the following information:

API Key Details

- Expiration date
- API Usage Information (If provided)

OAuth 2.0 Token Details

- The URLs to obtain the access tokens
- Expiration date
- API Usage Information (If provided)

Viewing Access Token Details Through User Preferences Page

> To view the details of an access token

1. Open a web browser and navigate to the CentraSite Business UI.
2. In CentraSite Business UI, click your user name that is located in the upper-right corner of header area.

This opens the User Preferences page.

3. Locate the section **My Access Keys**.

Note:

The **My Access Keys** section *is not visible* unless you have at least one access token.

4. In the displayed list of access tokens, hover over the access token for which you want to examine the details.

For each API key or OAuth 2.0 token, CentraSite displays the value, and the expiration date.

When you hover over an access token, CentraSite displays one or more actions that you can perform on that particular token:

- Renew the API key, provided the key has a limited usage period.
- Revoke the API key or OAuth 2.0 token temporarily from the CentraSite Registry Repository.

Viewing Undelivered Access Tokens

Access tokens that are not delivered during the retry attempts to API Portal are flagged and persisted in the database. To view the list of undelivered access tokens, add the portlet named **GetUndeliveredAccessTokens** to your Welcome page.

➤ To add the **GetUndeliveredAccessTokens** portlet to your Welcome page

1. In CentraSite Business UI, click the **Welcome** link that is located in the upper-right corner of the header area.

The Welcome page displays a list of portlets that are configured for your view.

2. Click the **Configure** link (below the label **Welcome to CentraSite Business UI**).

This opens the **Configure your Welcome Page** dialog box with the list of portlets that are available to you.

3. In the **Configure Your Welcome Page** dialog box, click the **create a portlet** link.
4. In the **Create Portlet** dialog box, provide the required information for each of the displayed data fields:

Field	Description
Name	Name of the portlet. For example, <code>Undelivered Access Tokens</code> . A portlet name can contain any character (including spaces). A portlet name does not need to be unique within the Welcome page. However, to reduce ambiguity, you should avoid giving multiple portlets of the same type the same name. As a best practice, we recommend that you adopt appropriate naming conventions to ensure that portlets are distinctly named within the Welcome page.
Description	<i>Optional.</i> The description for the portlet.
Type	The portlet display type, Text .
Number of Entries	<i>Optional.</i> The number of entries to display in the portlet. If the number of entries is set to 0, then all the available entries are displayed.
Data Feed	Select Undelivered Access Tokens .
Attributes	The display attributes for the portlet. The attributes are dynamically displayed depending on the selected search query.

- Click the chevron to expand the **Advanced Settings** panel. Provide the required information for each of the displayed data fields:

Field	Description
Actions	<p><i>Optional.</i> A set of actions that are available for the configuration of the Text portlet.</p> <p>The default actions include:</p> <ul style="list-style-type: none"> ■ Refresh ■ Configure
Icon URL	<p><i>Optional.</i> The path to an image file that is used to represent this portlet in the Welcome page.</p> <p>Prerequisite: The image file must be in PNG format. To ensure proper alignment when it is displayed in the user interface, the image must be 16 x 16 pixels in size.</p> <p>The image must reside in the folder <code><CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\images\system.</code></p> <p>The path for the image should be specified, for example, as <code>images/system/icon.png</code></p>
Refresh Interval	<p><i>Optional.</i> The time interval (in seconds) after which a the portlet content is refreshed..</p> <p>If a value is not specified (or if the value 0 is specified), refresh will not happen.</p>
Suppress Zero Values	<p>The Suppress Zero Values option button determines how rows containing zero values are handled in the portlet. When this option button is selected, any row that contains a value equal to zero is hidden in the portlet.</p> <p>By default, the Suppress Zero Values property is set to Yes.</p>

- Click **OK**.

The newly created **Undelivered Access Tokens** portlet is added to the CentraSite Registry Repository, and you is redirected to the **Configure Your Welcome Page** dialog box.

By default, the newly created portlet is disabled and is not displayed in the Welcome page.

- In the **Configure Your Welcome Page** dialog box, select the **Undelivered Access Tokens** portlet to add to your Welcome page.
- Click **OK**.

The **Undelivered Access Tokens** portlet is displayed in your personalized Welcome page.

Retry Mechanism for Guaranteed Access Token Delivery

Retry mechanism ensures the guaranteed delivery of messages from CentraSite to API Portal, and protects requests from transient failures that might occur while sending messages from CentraSite to API Portal.

When an API developer requests an API access token (access token of the type, API key or OAuth2 token) in API Portal, CentraSite receives the request for the access token, and processes the request. CentraSite then generates the access token and sends the access token details to API Portal. If a transient failure occurs while sending the access token to API Portal, CentraSite uses the retry mechanism to resend the token. Requests for renewal and revocation of access tokens are also handled in a similar manner.

Retry Mechanism for CentraSite 9.12 Configured with API Portal 9.12

Beginning with version 9.12, API Portal is configurable to work as a standalone component. This means that an instance of API Portal can be integrated with any third-party API Management tool (other than CentraSite or Mediator) for the access token retrieval functions of APIs deployed in the third-party environments.

For CentraSite instances configured to API Portal 9.12, CentraSite uses default scheduler to resend the undelivered access tokens to API Portal. The `PollingInterval` of the default scheduler is set to 120 minutes. CentraSite resends the undelivered access tokens at a regular time interval of 2 hours (120 minutes).

You can view and change the default scheduler settings using the `centrasite.xml` configuration. For more information, see [Changing the Default Scheduler for Resending Undelivered Access Tokens](#) later in this chapter.

Retry Mechanism for CentraSite 9.12 Configured with API Portal 9.10

CentraSite uses the global values set for `numberOfRetries` and `retryIntervals` to resend the undelivered tokens to API Portal. For n retries, specify n number of values (comma separated) for retry intervals. By default, `numberOfRetries` = 5 and `retryIntervals` = 10,300,900,1800,3600 seconds. In case of a transient failure, CentraSite tries to resend the token 5 times at time intervals of 10, 300, 900, 1800 and 3600 seconds respectively.

You can view and change the default values of `numberOfRetries` and `retryIntervals` using the command line interface `CentraSiteCommand.cmd` (Windows) or `CentraSiteCommand.sh` (UNIX) of CentraSite Command. For usage description of the command line, see [Managing Access Tokens through Command Line Interface](#) later in this chapter.

Changing the Default Scheduler for Fetching Access Token Requests

Beginning with version 9.12, API Portal is configurable to work as a standalone component. This means that an instance of API Portal can be integrated with any third-party API Management tool (other than CentraSite or Mediator) for the access token requests of APIs deployed in the third-party environments.

When an API developer requests an API access token (access token of type API key or OAuth2) in API Portal, the API access token request is sent to CentraSite. CentraSite then receives the request for the access token, generates the access token, and sends the access token details to API Portal. If a transient failure occurs while sending the access token request to CentraSite, as a fallback option for the transient failure, the access token request is flagged and stored in an events table in the API Portal's internal database. Requests for renewal and revocation of access tokens are also handled in a similar manner.

When CentraSite is configured to API Portal 9.12, CentraSite uses a default scheduler that is scheduled to run once every two hours (120 minutes) to pick up such access token requests from the API Portal's internal database. CentraSite then processes the access token requests, and sends the access token details to API Portal.

To reschedule the scheduler to run at a different time interval (other than the default 120 minutes), change the global value of `PollingInterval` in the **centrasite.xml** file.

➤ To change the default scheduler for fetching access token requests

1. Open the customization file, `centrasite.xml`, in a rich text editor.

You can find the **centrasite.xml** file on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\custom\conf`.

2. Locate the property named `<AccessTokenFetchSettings>`

The property `<AccessTokenFetchSettings>` is commented and would look like the following:

```
<AccessTokenFetchSettings>
  <!-- API Keys and OAuth request New, Renew, Revoke
    is fetched periodically -->
  <!-- Interval duration is minutes -->
  <!-- Default fetching interval is 120 minutes -->
  <FetchingInterval>120</FetchingInterval>
</AccessTokenFetchSettings>
```

3. Uncomment the property `<AccessTokenFetchSettings>`.
4. Change the default value set for the property `PollingInterval`.
5. Save the configuration file.
6. Restart Software AG Runtime for the changes to take effect.

Post-requisites:

If at a later time, you want to switch to the default scheduler settings, comment the property `<AccessTokenFetchSettings>`, and then restart Software AG Runtime.

Changing the Default Scheduler for Resending Undelivered Access Tokens

If access tokens are not delivered during the retry attempts, the undelivered tokens are flagged and stored in the CentraSite's internal database.

When CentraSite is configured to API Portal 9.12, CentraSite uses a default scheduler that is scheduled to run once every two hours (120 minutes) to pick up the undelivered tokens, and resend them to API Portal.

To reschedule the scheduler to run at a different time interval (other than the default 120 minutes), change the global value of `PollingInterval` in the **centrasite.xml** file.

➤ To change the default scheduler for resending undelivered access tokens

1. Open the customization file, `centrasite.xml`, in a rich text editor.

You can find the **centrasite.xml** file on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\custom\conf`.

2. Locate the property named `<AccessTokenPublishSettings>`.

The property `<AccessTokenPublishSettings>` is commented and would look like the following:

```
<!-- <AccessTokenPublishSettings>
  <!-- Undelivered API Keys and OAuth tokens is
    delivered periodically -->
  <!-- Interval duration is minutes -->
  <!-- Default polling interval is 120 minutes -->
  <PollingInterval>120</PollingInterval>
</AccessTokenPublishSettings> -->
```

3. Uncomment the property `<AccessTokenPublishSettings>`.
4. Change the default value set for the property `PollingInterval`.
5. Save the configuration file.
6. Restart Software AG Runtime for the changes to take effect.

Post-requisites:

If at a later time, you want to switch to the default scheduler settings, comment the property `<AccessTokenPublishSettings>`, and then restart Software AG Runtime.

Renewing API Access Keys

To renew an API access key, you must belong to the API Runtime Provider role.

After an API access key is generated, sometimes you might have to renew the old key due to expiration or security concerns. You can also change expiration period for the access key or set it so that the key never expires.

Note:

The **Renew** functionality is not supported for the OAuth 2.0 access tokens.

To renew an API access key, make sure that:

- You have configured the API key authentication using the **API Consumption Settings** action in the details page of the asset.
- A gateway instance (for example, Mediator) is up and running.

➤ **To renew an API key**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.This displays a list of assets for which you have View permission in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of any type of Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select the required type of Service.
 - c. Click **OK**.
5. In the displayed list of Service assets, click the asset for which you want to display the API access key details.

This opens the Service Details page.
6. Locate the hyperlinked text **N** next to **Consumers** attributes in the **Basic Information** profile.
7. Click on the hyperlinked consumer name whose API key you want to renew.
8. In the displayed list of API keys/OAuth 2.0 tokens, hover over the API key you want to renew.

This displays one or more actions you can perform on the API key.
9. Click the **Renew** icon.

Important:

If an API key has unlimited expiration period, then the **Renew** icon will not be displayed for that particular key.

Revoking Access Tokens as API Provider

An API Provider or an administrator will use the Asset Details page to revoke an access token.

After issuing an access token, you might want to revoke the token if you find a serious error in the virtual instance of an asset.

When you revoke an access token, access to the associated virtual asset, and its resources is blocked when you try to access them using that particular access token.

- You have configured the API key authentication or OAuth 2.0 token authentication using the **API Consumption Settings** action in the details page of the asset.
- A gateway instance (for example, Mediator) is up and running.

> To revoke access token as an API Provider

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.This displays a list of assets for which you have View permission in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of any type of Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select the required type of Service.
 - c. Click **OK**.
5. In the displayed list of Service assets, click the asset for which you want to display the API access key details.

This opens the Service Details page.

6. Locate the hyperlinked text **N** next to **Consumers** attributes in the **Basic Information** profile.
7. Click on the hyperlinked consumer name whose access token you want to revoke.

8. In the displayed list of access tokens, hover over the access token you want to revoke.

CentraSite displays one or more actions you can perform on the access token.

9. Click the **Delete** icon.

A confirmation message appears that the access token is revoked from the CentraSite Registry Repository.

10. Click **Yes** in the confirmation dialog box.

Once the access token revocation is processed, CentraSite sends an email message to the API Consumer informing that the request has been processed successfully.

CentraSite provides predefined email template for the access token revocation. By default, this template is configured in the **centrasite.xml** file. But, if you do not want to use the predefined email template, you can create and add your own email template to CentraSite, and configure the **centrasite.xml** file, as required.

Revoking Access Tokens as API Consumer

An API Consumer will use the User Preferences page to revoke an access token.

After issuing an access token, you might want to revoke the token if you find a serious error in the virtual instance of an asset.

When you revoke an access token, access to the associated virtual asset, and its resources is blocked when you try to access them using that particular access token.

- You have configured the API key authentication or OAuth 2.0 token authentication using the **API Consumption Settings** action in the details page of the asset.
- A gateway instance (for example, Mediator) is up and running.

➤ To revoke access token as an API Consumer

1. Open a web browser and navigate to the CentraSite Business UI.
2. In CentraSite Business UI, click your user name that is located in the upper-right corner of header area.

This opens the User Preferences page.

3. Locate the section **My Access Keys**.
4. In the displayed list of access tokens, hover over the access token you want to revoke.

CentraSite displays one or more actions you can perform on the access token.

5. Click the **Delete** icon.

A confirmation message appears that the access token is revoked from the CentraSite Registry Repository.

6. Click **Yes** in the confirmation dialog box.

Once the access token revocation is processed, CentraSite sends an email message to the API Consumer informing that the request has been processed successfully.

CentraSite provides predefined email template for the access token revocation. By default, this template is configured in the **centrasite.xml** file. But, if you do not want to use the predefined email template, you can create and add your own email template to CentraSite, and configure the **centrasite.xml** file, as required.

Deleting Access Tokens

Deleting an access token permanently removes the entry for the access token (that is, it removes the asset instance for the access token from the CentraSite's object database). Deleting an access token does not remove the API asset that is associated with it.

When you delete an API key, CentraSite removes an entry for the API key (that is, it removes the instance of the API key asset from CentraSite's object database). Also note that:

- You could delete an access token only if it is already revoked.
- You cannot delete an access token that is in the Pending mode (for example, awaiting a renew approval).
- You must be a user with Manage Assets permission to delete the access token (that is, available as the API Key asset or the OAuth2 Client asset in CentraSite).

An API Provider and users with the appropriate permissions can delete access tokens.

➤ To delete access tokens

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of assets for which you have View permission in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the access tokens, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.

- b. In the displayed list of asset types, select **API Key** or **OAuth2 Client**.
 - c. Click **OK**.
5. In the displayed list of API keys or OAuth 2.0 access tokens, select the check box for one access token, or select the check boxes for multiple access tokens you want to delete.
 6. On the actions bar of the Search Results page, click the **Delete** icon.

You can also delete a single access token from the actions bar of that particular Access Token Details page.
 7. Click **Yes** in the confirmation dialog box.

The selected access tokens are permanently removed from the CentraSite Registry Repository.

Configuring Email Templates

Notifications informing the various activities of the API key and OAuth 2.0 token are generated and sent to both the API Provider and API Consumer.

CentraSite supports the following types of notifications for the access tokens:

- An access token request is pending for approval - an information type message to the API Provider.
- An access token request is approved - an information type message to the API Consumer.
- An access token request is rejected - an information type message to the API Consumer.

CentraSite provides predefined email template for the access token generation. By default, this template is configured in the `centrasite.xml` file. But, if you do not want to use the predefined email template, you can create and add your own email template to CentraSite, and configure the `centrasite.xml` file as necessary.

Configuring Email Templates for Access Token Generation

Notifications about the usage of new API key or OAuth 2.0 access token are generated and sent to both the API Provider and API Consumer.

CentraSite supports the following types of notifications for the access token generation:

- An access token request is pending for approval - an information type message to the API Provider.
- An access token request is approved - an information type message to the API Consumer.
- An access token request is rejected - an information type message to the API Consumer.

➤ **To configure notifications for access token generation**

1. Open the customization file, `centrasite.xml`, in a rich text editor.

You can find the **centrasite.xml** file on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\custom\conf`.

2. Locate the property `KeyGenerationSettings`.

The notification settings for the API key and OAuth 2.0 token generation would look like the following:

```
<KeyGenerationSettings>
  <Approve
    subject="CS_MSG_INMBU_DEFAULT_EMAIL_SUBJECT_ACCESS_KEY_GENERATION_SUCCESS_OR_APPROVE"
    template="APIKeyGenerationSuccess.html" />
  <ApprovalRequest
    subject="CS_MSG_INMBU_DEFAULT_EMAIL_SUBJECT_ACCESS_KEY_GENERATION_PENDING"
    template="PendingApprovalNotification.html" />
  <Reject
    subject="CS_MSG_INMBU_DEFAULT_EMAIL_SUBJECT_ACCESS_KEY_GENERATION_REJECTED"
    template="RejectionNotification.html" />
</KeyGenerationSettings>
```

3. Uncomment the section `API Key Settings` to enable the access token generation notifications.
4. Use the property, `Approve`, to set the subject and body of the notification message for the API Consumer whose access token generation request is approved.
5. Use the property, `ApprovalRequest`, to set the subject and body of the notification message for the API Provider who has an access token generation request pending for approval.
6. Use the property, `Reject`, to set the subject and body of the notification message for the API Consumer whose access token generation request is rejected.
7. Save and close the file.
8. Restart Software AG Runtime.

Configuring Email Templates for Access Token Renewal

Notifications informing the new validity of the API key and OAuth 2.0 token are generated and sent to both the API provider and API consumer.

CentraSite supports the following types of notifications for the access token renewal:

- An API key renewal request is pending for approval - an information type message to the API provider.
- Your API key renewal request is approved - an information type message to the API consumer.

- Your API key renewal request is rejected - an information type message to the API consumer.

➤ To configure email templates for access token renewal

1. Open the customization file, `centrasite.xml`, in a rich text editor.

You can find the **centrasite.xml** file on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\custom\conf`.

2. Locate the property `KeyRenewalSettings`.

The notification settings for the API key and OAuth 2.0 token renewal would look like the following:

```
<KeyRenewalSettings>
<Approve
  subject="CS_MSG_INMBU_DEFAULT_EMAIL_SUBJECT_ACCESS_KEY_RENEWAL_SUCCESS_OR_APPROVE"
  template="APIKeyRenewalSuccess.html" />
<ApprovalRequest
  subject="CS_MSG_INMBU_DEFAULT_EMAIL_SUBJECT_ACCESS_KEY_RENEWAL_PENDING"
  template="APIKeyRenewalPendingNotification.html" />
<Reject
  subject="CS_MSG_INMBU_DEFAULT_EMAIL_SUBJECT_ACCESS_KEY_RENEWAL_REJECT"
  template="RejectionNotification.html" />
</KeyRenewalSettings>
```

3. Uncomment the section `API Key Settings` to enable the access token renewal notifications.
4. Use the property, `Approve`, to set the subject and body of the notification message for the API Consumer whose access token renewal request is approved.
5. Use the property, `ApprovalRequest`, to set the subject and body of the notification message for the API Provider who has an access token renewal request pending for approval.
6. Use the property, `Reject`, to set the subject and body of the notification message for the API Consumer whose access token renewal request is rejected.
7. Save and close the file.
8. Restart Software AG Runtime.

Configuring Email Templates for Access Token Revocation

Once the API key or the OAuth 2.0 token revocation is processed, CentraSite sends an email message to the API Consumer informing that the request has been processed successfully.

➤ To configure email templates for access key revocation

1. Open the customization file, `centrasite.xml`, in a rich text editor.

You can find the **centrasite.xml** file on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\custom\conf`.

2. Locate the property `KeyRevocationSettings`.

The notification settings for the API key and OAuth 2.0 token revocation would look like the following:

```
<KeyRevocationSettings>
  <Approve
    subject="CS_MSG_INMBU_DEFAULT_EMAIL_SUBJECT_ACCESS_KEY_REVOCATION_
      NOTIFICATION" template="APIKeyRevocationSuccess.html"/>
</KeyRevocationSettings>
```

3. Uncomment the section `APIKeySettings` to enable the access token revocation notifications.
4. Use the property, `Approve`, to provide a suitable subject if you do not want to use the default subject, and specify the name of the custom email template.
5. Save and close the file.
6. Restart Software AG Runtime.

Configuring Email Templates for Access Token Expiration

Notifications informing about the upcoming API key and OAuth 2.0 token expirations, and the newly generated access tokens are sent to the API Consumer.

CentraSite supports the following types of notifications for the access token expiration:

- Access token has expired - a critical event type message.
- Access token expires soon - a warning type message. It is generated `n` days before the token expiration date and displayed every day before the token actually expires.

➤ To configure email templates for access token expiration

1. Open the customization file, `centrasite.xml`, in a rich text editor.

You can find the **centrasite.xml** file on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\custom\conf`.

2. Locate the property `ExpiryNotificationSettings`.

The notification settings for the API key and OAuth 2.0 token expiration would look like the following:

```
<ExpiryNotificationSettings>
  <ExpiredNotification
    subject="Access key has expired!"
    template="APIKeyExpiredNotification.html" />
```

```
<AdvanceNotification
  subject="Access key about to expire!"
  template="APIKeyExpirationNotification.html" />
<SchedulerExecutionFrequency>12h</SchedulerExecutionFrequency>
<AdvanceNotificationInterval>5d</AdvanceNotificationInterval>
</ExpiryNotificationSettings>
```

3. Uncomment the section `API KEY EXPIRATION CONFIGURATION` to enable the access token expiration notifications.
4. Use the property, `ExpiredNotification`, to set the subject and body of the notification message for the API Consumers whose access token has expired.
5. Use the property, `AdvanceNotification`, to set the subject and body of the notification message for the API Consumers whose access tokens are due to expire.
6. Use the property, `SchedulerExecutionFrequency`, to specify how frequently to check for the expiration status of access tokens. Enter the time interval in the following format: years (y), months (m), days (d), hours (h), minutes (min).
7. Use the property, `AdvanceNotificationInterval`, to specify how many days should the consumers be notified before the access token expiration (in days). The consumers receive a notification message as configured in the `AdvanceNotification` property. Enter the time interval in the following format: years (y) months (m) days (d) hours (h) minutes (min).
8. Save and close the file.

Important:

If you have set up a Software AG Runtime cluster with load balancing, locate the `CENTRASITE ACCESS URL CONFIGURATION` element, and ensure that the `lb_or_reverse_proxy_url` attribute in the following property points to the load balancer's IP/Port.

```
<CentraSite url="http://localhost:53307/CentraSite/CentraSite"
  lb_or_reverse_proxy_url="http://localhost:53307"/>
```

9. Restart Software AG Runtime.

Managing Access Tokens through Command Line Interface

This section describes operations you can perform to manage access tokens through Command Line Interface.

Fetching the Default Values for Retry Mechanism

Pre-requisites:

To fetch the default values for retry configuration through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite uses the default values set for the global properties *numberOfRetries* and *retryIntervals* for the guaranteed delivery of messages from CentraSite to API Portal. By default, *numberOfRetries* = 5 and *retryIntervals* = 10,300,900,1800,3600 seconds. In case of a transient failure, CentraSite tries to resend the undelivered access tokens 5 times at time intervals of 10,300,900,1800 and 3600 seconds respectively.

These default values for resending the undelivered access tokens are stored in the CentraSite's internal database. You can view the default values for retry configuration when required.

CentraSite provides a command tool named `get ApiPortalConfig` for this purpose. The tool displays the default values set for *numberOfRetries* and *retryIntervals* in the console and immediately writes these default values for retry configuration in the XML file identified by the `-file` parameter.

➤ To fetch the default values for retry configuration

- Run the command `get ApiPortalConfig`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get ApiPortalConfig [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -file <CONFIG-FILE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
CONFIG-FILE	Name of the XML file where you want to write the default values of <i>numberOfRetries</i> and <i>retryIntervals</i> .

Note:

If the file is in a different location other than `<CentraSiteInstall_Directory>/utilities`, provide the absolute file path.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd get ApiPortalConfig -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-file C:\CentraSite\configuration\test.xml
```

The response to this command could be:

```
Executing the command : get ApiPortalConfig
API Portal configuration was successfully written to C:\config.xml file.
```

API Portal CONFIGURATIONS

```
-----
com.centrasite.apiportal.config.retry.RetryIntervals : 10,300,900,1800,3600
com.centrasite.apiportal.config.retry.NumberOfRetries : 5
Successfully executed the command : get ApiPortalConfig
```

Changing the Default Values for Retry Mechanism

Pre-requisites:

To change the default values for retry configuration through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite uses the default values set for the global properties *numberOfRetries* and *retryIntervals* for the guaranteed delivery of messages from CentraSite to API Portal. By default, *numberOfRetries* = 5 and *retryIntervals* = 10,300,900,1800,3600 seconds. In case of a transient failure, CentraSite tries to resend the undelivered access tokens 5 times at time intervals of 10,300,900,1800 and 3600 seconds respectively.

The default values for resending the undelivered access tokens are stored in the CentraSite's internal database. You can change the default values for retry configuration to suit your individual requirements.

CentraSite provides a command tool named `set ApiPortalConfig` for this purpose. The tool displays the updated values for *numberOfRetries* and *retryIntervals* in the console.

When changing the default values set for the global properties *numberOfRetries* and *retryIntervals*, keep the following points in mind:

- If *numberOfRetries* is more than the number of values specified for *retryIntervals*, CentraSite uses the last time interval specified in *retryIntervals* for the remaining retry attempts. For example, if *numberOfRetries* = 5 and *retryIntervals* = 10,300,900,1800 seconds, CentraSite tries to resend the undelivered access tokens 5 times at time intervals of 10,300,900,1800,1800 seconds.
- If *numberOfRetries* is less than the number of values specified for *retryIntervals*, CentraSite ignores the values that are beyond the value set for *numberOfRetries*. For example, if *numberOfRetries* = 4 and *retryIntervals* = 10,300,900,1800,3600,7200 seconds, CentraSite tries to resend the undelivered access tokens 4 times at time intervals of 10,300,900,1800 seconds. Time intervals beyond 1800 seconds are ignored.

➤ To change the default values for retry configuration

- Run the command `set ApiPortalConfig`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set ApiPortalConfig [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -file <CONFIG-FILE>`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
CONFIG-FILE	Name of the XML file where <i>numberOfRetries</i> and <i>retryIntervals</i> values are changed.

Note:

If the file is in a different location other than `<CentraSiteInstall_Directory>/utilities`, provide the absolute file path.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd set ApiPortalConfig -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-file C:\CentraSite\configuration\test.xml
```

The response to this command could be:

```
Executing the command : set ApiPortalConfig
API Portal CONFIGURATIONS
-----
com.centrasite.apiportal.config.retry.RetryIntervals : 10,10,10,10,10
com.centrasite.apiportal.config.retry.NumberOfRetries : 5
Successfully executed the command : set ApiPortalConfig
```

Resending Undelivered Access Tokens

Pre-requisites:

To resend the undelivered access tokens through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

If access tokens are not delivered during the retry attempts, the undelivered tokens are flagged and persisted in the database.

CentraSite provides a command tool named `send AccessTokens` for this purpose.

- Run the command `send AccessTokens`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd send AccessTokens [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> [-api <API>] [-apiPortal <APIPORTAL>] [-accessToken <ACCESSTOKEN>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
API	Name or ID (uuid) of the Virtual Service (API). For multiple services, use a comma to separate the values. All values should either be the names or the IDs of the services, not a combination of both.
APIPORTAL	Name or ID (uuid) of the access tokens to be resent. To resend multiple tokens to the API Portal, use a comma to separate the values. All values should either be the names or the IDs of the access tokens, not a combination of both.
ACCESSTOKEN	Name or ID (uuid) of the API Portal to which the tokens have to be resent. To resend tokens to multiple API Portals, use a comma to separate the values. All values should either be the names or the IDs of the API Portals, not a combination of both.

Note:

To resend the undelivered access tokens using `send AccessTokens` command, provide values for any one of the parameters:

- `-api`
- `-apiPortal`
- `-accessToken`

If you do not specify values for any of the three parameters, CentraSite resends all the undelivered tokens to the respective API Portals.

Examples (all in one line):

Providing multiple IDs

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd send AccessTokens -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-api uddi:5f0ad20e-9bdd-11e4-9184-dc550a8f1855, 9f0ad41e-9crd-11e4-9172-dc550a8f2345
```

Providing multiple service names

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd send AccessTokens -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-api ApprovalVirtualService,SearchVirtualService
```

If the service, access tokens, or the API Portal names contain white spaces, enclose the names within "".

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd send AccessTokens -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-api "Approval Virtual Service, Search Virtual Service"
```

Purging Expired or Inactive Access Tokens

Pre-requisites:

To purge expired, revoked, or inactive access tokens through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `purge accesstokens` for this purpose.

➤ To purge expired or inactive access tokens

- Run the command `purge accesstokens`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd purge accesstokens [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> [-keyType <KEY-TYPE>] [-beforeDate <BEFORE-DATE>] [-batchSize <BATCH-SIZE>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
KEY-TYPE	(Optional). The type of access tokens you want to purge: "APIKey", "OAuth", or "All". If you omit this parameter, CentraSite assumes the default "All" configuration.
BEFORE-DATE	(Optional). Purges the access tokens that were generated before the specified date. The date format defaults to <code>yyyy-mm-dd</code> . When specifying the date, enclose the date value within "".
BATCH-SIZE	(Optional). The maximum number of access tokens you want to purge in a batch. The default value for the batch size is 500.

Note:

To purge the expired access tokens using the `purge accesstokens` command, you can optionally specify values for the following parameters:

- `keyType`
- `untilDate`
- `batchSize`

If you do not specify values for any of these optional parameters, CentraSite purges all the available expired access tokens.

Examples (all in one line):

Specifying access tokens of type API Key:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd purge accesstokens -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-keyType APIKey
```

Specifying access tokens of type OAuth 2.0:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd purge accesstokens -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-keyType OAuth
```

Providing the date:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd purge accesstokens -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
-keyType OAuth -beforeDate "2014-12-30"
```

Providing the batch size:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd purge accesstokens-url
http://localhost:53307/CentraSite/CentraSite-user Administrator -password
manage-keyType OAuth -batchSize 1000
```

Restoring Expired Access Tokens

Pre-requisites:

To restore the expired access tokens through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `restore accesstokens` for this purpose.

> To restore expired access tokens

- Run the command `restore accesstokens`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd restore accesstokens [-url <CENTRASITE-URL>] -user <USER-ID> -password <PASSWORD> -serviceKey <SERVICE-KEY> [-beforeDate <BEFORE-DATE>] [-afterDate <AFTER-DATE>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the registered CentraSite user identified by the parameter <code>USER-ID</code> .
SERVICE-KEY	Name or ID (uuid) of the service for which the access tokens need to be restored. For multiple services, use a comma to separate the values. All values should either be the names or the IDs of the services, not a combination of both. If the service name contains white spaces, enclose the name within <code>""</code> .
BEFORE-DATE	(Optional). Restores the access tokens that were generated until the specified date. The date format defaults to <code>yyyy-mm-dd</code> . When specifying the date, enclose the date value within <code>""</code> .
AFTER-DATE	(Optional). Restores the access tokens that were generated after the specified date. The date format defaults to <code>yyyy-mm-dd</code> . When specifying the date, enclose the date value within <code>""</code> .

Note:

To restore the expired access tokens using the `restore accesstokens` command, you can optionally specify values for the following parameters:

- `beforeDate`
- `afterDate`

If you do not specify values for any of these optional parameters, CentraSite restores all the available access tokens for the corresponding API identified by the parameter `serviceKey`.

Examples (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd restore accesstokens -user Administrator -password manage -servicekey uddi:5f0ad20e-9bdd-11e4-9184-dc550a8f1855
```

Providing multiple service keys:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd restore accesstokens -url http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage
```

```
-serviceKey uddi:5f0ad20e-9bdd-11e4-9184-dc550a8f1855,  
uddi:9f0ad41e-9crd-11e4-9172-dc550a8f2345
```

Providing the date range:

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd restore accesstokens -url  
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage  
-serviceKey uddi:5f0ad20e-9bdd-11e4-9184-dc550a8f1855 -beforeDate "2014-12-30"
```

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd restore accesstokens -url  
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage  
-serviceKey uddi:5f0ad20e-9bdd-11e4-9184-dc550a8f1855 -afterDate "2014-12-30"
```

Run-Time Alias Management

Note:

This section is not applicable if the CentraSite run-time aspects are not enabled. By default, run-time aspects configured from CentraSite are disabled. However, you can enable them if required. To enable the CentraSite run-time aspects, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

Mediator Runtime Aliases

You develop, test, and publish a virtual service in Development stage, Test stage, and Production stage respectively. When you promote a virtual service from one stage to the next, you have to change environment settings such as:

- The runtime (routing) endpoint for the virtual service.
- The routing endpoint connection properties. For example, connection timeout settings and endpoint client certificates.
- Outbound authentication tokens.

You can promote a virtual service in CentraSite from one stage to the next by:

- Promoting the same virtual service from one stage to the next.

Using this method, you have to export or import the virtual service to the next stage, change its environment settings, and then deploy it. In addition, this method of promotion requires a dedicated CentraSite instance for each stage.

- Creating a separate virtual service for each stage.

Using this method only requires one instance of CentraSite, however, you have to use a unique name for each virtual service.

An easier way to promote a virtual service is to define separate routing runtime aliases to be used in each stage. These aliases are deployed to the runtime gateways along with the virtual services, and are referenced by the virtual services at runtime.

Defining the Runtime Aliases

Before you define the runtime (routing) endpoint aliases:

- Ensure that the gateways for the aliases have already been created. For example, Development, Testing, and Production gateways.
- Ensure that you have the API Run-time Provider role for creating an alias.
- Ensure that you have the Mediator Publisher role or Publishing permissions on the Mediator gateway for publishing the aliases to Mediator.

> To define runtime endpoint aliases

1. In the CentraSite Business UI activity bar, click **Governance Rules**.
2. Click **Add Runtime Alias**.
3. Provide the required information in the wizard 1 of the Add a New Runtime Alias dialog box.

Field	Description
Runtime alias type	<p>Select the type of runtime alias:</p> <ul style="list-style-type: none"> ■ Simple Alias: A simple routing URL or host and port name. ■ webMethods Integration Server Alias: Contains the webMethods Integration Server service name. <div style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;"> <p>Note: The webMethods Integration Server service must be available in the Integration Server, to which the aliases are deployed.</p> </div> <ul style="list-style-type: none"> ■ Secure Alias: Contains client's authentication credentials, and domain values. The password is hashed and put into secure storage so that it is not visible in clear text. ■ Transformation Alias: Accepts the XSLT style sheet to be used for the request or response transformation. ■ Endpoint Alias: A name and value pair which can also contain endpoint properties such as Connection Timeout, Read Timeout, and so on.
Authentication Scheme	<p>For the Secure Alias type only, select one of the following:</p> <ul style="list-style-type: none"> ■ HTTP Basic Authentication

Field	Description
	<ul style="list-style-type: none"> ■ NTLM Authentication ■ OAuth2 Authentication ■ Kerberos Authentication
Name	Name of the alias.
Description	(Optional) . The description of the alias.

4. Click **Next**.

5. Provide the required information in the wizard 2 of the Add a New Runtime Alias dialog box.

Field	Description
Default Value	<p>The value used by default when no stage-specific values are entered. This default value is overwritten at deployment time by stage-specific values, if they exist.</p> <p>For Simple or Endpoint Alias , type a default URL or components of the URL such as service name.</p> <p>For webMethods Integration Server Alias, type the webMethods Integration Server service name.</p> <p>For Transformation Alias, select an XSLT style sheet which is considered as a default value.</p>
Stage-specific Values	Stage-specific values are used to define the values that are specific to a particular Mediator. You can select an instance of Mediator and type the value to be used for that Mediator.
Endpoint Properties	<p>For the Endpoint Alias type only. Click the endpoint properties icon next to Default Value to configure endpoint properties.</p> <ul style="list-style-type: none"> ■ SOAP Optimization Method: Optional, This setting is not applicable for REST services. Mediator can accept the following methods to optimize the payloads of SOAP requests: <ul style="list-style-type: none"> ■ None (the default). ■ MTOM: Indicates that Mediator expects to receive a request with a Message Transmission Optimization Mechanism (MTOM) attachment, and forwards the attachment to the native service.

Field	Description
	<ul style="list-style-type: none"> <li data-bbox="516 258 1229 359">■ SwA: Indicates that Mediator expects to receive a SOAP with Attachment (SwA) request, and forwards the attachment to the native service. <li data-bbox="516 386 1229 667">■ Connection Timeout: The time interval (in seconds) after which a connection attempt timeouts. If a value is not specified (or if the value 0 is specified), Mediator uses the value of the global property <code>pg.endpoint.connectionTimeout</code> located in the file <code>Integration Server_directory/packages/WmMediator/config/resources/pg-config.properties</code>. The default of that property is 30 seconds. <li data-bbox="516 695 1229 760">■ Read Timeout: The time interval (in seconds) after which a socket read attempt will timeout.

The precedence of the Read Timeout configuration is as follows:

1. If a value is specified for this Read Timeout field, Mediator will use this value for the socket read attempt. The read timeout value defined at an alias level takes precedence over the timeout values defined at an API level and the global configuration.
2. If a value 0 is specified (or if the value is not specified) for this Read Timeout field, then Mediator will use the value specified in the Read Timeout field of the routing action. The read timeout value defined at an API level takes precedence over the global configuration.
3. If a value 0 is specified (or if the value is not specified) for the Read Timeout field in the routing action (at an API level), then Mediator will use the value of the global property `pg.endpoint.readTimeout` located in the file `Integration Server_directory/packages/WmMediator/config/resources/pg-config.properties` (in the Mediator Administration console, go to **> Settings > Extended Settings > pg.endpoint.readTimeout** property.).

Note:

If a value for the Read Timeout configuration is not specified in any of the above configuration parameters, then Mediator will use the default 30 seconds.

Field	Description
	<p data-bbox="552 252 1328 430">SSL Options: To enable SSL client authentication for the endpoint, you must specify values for both the Keystore Alias field and the Client Certificate Alias field. If you specify a value for only one of these fields, a deployment error occurs.</p> <div data-bbox="552 451 1328 577" style="background-color: #f0f0f0; padding: 5px;"> <p data-bbox="552 462 1328 567">Note: SSL client authentication is optional; you may leave both fields blank.</p> </div> <ul data-bbox="552 598 1328 1113" style="list-style-type: none"> <li data-bbox="552 598 1328 745">■ Keystore Alias: The keystore alias of the instance of Integration Server on which Mediator is running. This value (along with the value of Client Certificate Alias) is used for performing SSL client authentication. <li data-bbox="552 766 1328 1113">■ Client Certificate Alias: The client's private key to be used for performing SSL client authentication. If you specify a client certificate alias, you must also include in the virtual service's policy the Require SSL action and select that action's Client Certificate Required option. The Client Certificate Required option specifies whether client certificates are required for the purposes of: 1) Verifying the signature of signed SOAP requests or decrypting encrypted SOAP requests, and 2) Signing SOAP responses or encrypting SOAP responses. <p data-bbox="552 1134 1328 1239">WS-Security Header Customization: Indicates whether Mediator should pass the WS-Security headers of the incoming requests to the native service.</p> <ul data-bbox="552 1260 1328 1711" style="list-style-type: none"> <li data-bbox="552 1260 1328 1543">■ Remove processed security headers: Removes the security header if it is processed by Mediator (that is, if Mediator processes the header according to the virtual service's security run-time policy). Mediator does <i>not</i> remove the security header if <i>both</i> of the following conditions are true: 1) Mediator did not process the security header, and 2) the <code>mustUnderstand</code> attribute of the security header is 0 or false). <li data-bbox="552 1564 1328 1711">■ Pass all security headers: Passes the security header, even if it is processed by Mediator (that is, even if Mediator processes the header according to the virtual service's security run-time policy). <div data-bbox="552 1732 1328 1858" style="background-color: #f0f0f0; padding: 5px;"> <p data-bbox="552 1743 1328 1848">Note: If the virtual service does not contain a security run-time policy, and the <code>mustUnderstand</code> attribute of</p> </div>

Field	Description
	the security header is 0 or false, then Mediator <i>always</i> forwards the security header to the native service.
OAuth2 Token	A valid OAuth 2.0 token. The specified token will be used by Mediator in the outbound request.

Examples:

Endpoint Type	Field Values
Alias	<ul style="list-style-type: none">■ Name: ProdSandbox■ Default Value: http://myhost:5555
webMethods Integration Server Alias	<ul style="list-style-type: none">■ Name: ProdService■ Default Value: fault:postProcess
Secure Alias	<ul style="list-style-type: none">■ Name: ProdAuthToken■ Password: ****■ Domain: mysever.sag
Transformation Alias	<ul style="list-style-type: none">■ Name: ProdTrans■ Default Value: sample.xsl
Endpoint Alias	<ul style="list-style-type: none">■ Name: SearchServiceProductionEndpoint■ Description: The clustered production endpoint for the search service.■ Default Value: http://prodcluster:6666/Search■ Endpoint Properties:<ul style="list-style-type: none">■ Connection Timeout: 5■ Read Timeout: 15

6. Click **OK**.

7. Click **Publish**.

Referencing the Runtime Aliases in Virtual Services

Once you have defined the runtime (routing) aliases, you need to reference them in the virtual services. Before you do this, the following prerequisites must be met:

- Ensure that the gateways for the aliases have already been created. For example, Development, Testing, and Production gateways.
- Ensure that you have the API Run-time Provider role for creating an alias.
- Ensure that you have the Mediator Publisher role for publishing the aliases to Mediator.

➤ **To reference an alias in a virtual service**

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link in the upper-left corner of the menu bar.
- Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.

This displays a list of defined assets for which you have the View permission in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Virtual Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **Virtual Service**.
 - c. Click **OK**.
5. In the displayed list of Virtual Services, click the Virtual Service for which you want to reference the runtime alias.

This opens the Virtual Service Details page. Also, the actions bar displays a set of actions that are available for working with the virtual service.

6. On the actions bar of the Virtual Service Details page, click **Virtualize**.

This opens the **Virtualize <Service_Name> (Step 2 of 3)** wizard.

7. For Simple and Endpoint aliases. In the **Policy Actions** window, **Policy Enforcement > Routing** section, select one of the following routing actions and use `${<aliasname>}` (`<aliasname>` is the name of the alias specified in the previous step) in **Route to**, to form the complete routing URL:

Routing Action	Description
Straight Through Routing	Routes the requests directly to a native endpoint that you specify.
Context Based Routing	Route requests to different endpoints based on specific values that appear in the request message.
Content Based Routing	Route requests to different endpoints based on specific criteria that you specify.
Load Balancing and Failover Routing	Routes the requests across multiple endpoints.
Dynamic Routing	Route requests to a dynamic URL based on specific criteria that you specify.

8. For Secure alias only. In the **Policy Actions** window, **Policy Enforcement > Outbound Authentication**:
 - a. Select one of the following authentication actions:

Routing Action	Description
HTTP Basic Authentication	Use when native API enforces basic authentication. Based on the modes selected, Mediator either uses configured basic authentication credentials to invoke a native service or it uses credentials from the authorization header of the incoming request to access the native API.
NTLM Authentication	Used when native API enforces NTLM authentication. Based on the modes selected, Mediator either uses configured authentication credentials to obtain the NTLM token to invoke the native service or it uses credentials from the authorization header of the incoming request to obtain the NTLM token to access the native API.
OAuth2 Authentication	Used when native API enforces OAuth authorization. Based on the modes selected, Mediator either uses configured OAuth token to invoke the native service or it uses the OAuth token of the incoming request to access native service.
Kerberos Authentication	Used when service provider wants a web service client that does not have the ability to generate the Kerberos token to access the service enforced with the Kerberos policy. It is also used when service provider wants a

- | Routing Action | Description |
|----------------|---|
| | web service client to access a service enforced with the Kerberos policy. |
- b. For **Authenticate Using**, select **Secure Alias**.
 - c. Type the **Alias name**.
9. For **webMethods Integration Server** only. In the **Policy Actions > Request Handling** or in the **Response Processing** window, select **Invoke webMethods Integration Server**.
To configure **Invoke webMethods Integration Server**:
 - a. Select **webMethods IS Service Alias** in the **Select Type** option.
 - b. Type the alias name (already created) in the **webMethods IS Service Alias** field.
 10. For **Transformation Alias** only, select one of the following:
 - **Policy Actions > Request Handling > Request Transformation**
 - **Policy Actions > Response Processing > Response Transformation**
 To configure **Request** or **Response Transformation**:
 - a. Select **Transformation Alias** in the **Select Type** option.
 - b. Type the alias name (already created) in the **Transformation Alias** field.
 11. Click **Virtualize**.

Publishing a Virtual Service with Runtime Alias

> To publish a virtual service that references the runtime alias

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link in the upper-left corner of the menu bar.
 - Click the **Search** icon next to the **Scope** list. The default search scope is **Assets**.
 This displays a list of defined assets for which you have the View permission in the Search Results page.
2. In the **Additional Search Criteria** list, select **Asset Types**.

3. To search for the assets of type, Virtual Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and then follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the displayed list of asset types, select **Virtual Service**.
 - c. Click **OK**.

5. In the displayed list of Virtual Services, click the Virtual Service you want to publish with its referenced alias.

This opens the Virtual Service Details page. Also, the actions bar displays a set of actions that are available for working with the virtual service.

6. On the actions bar for the Virtual Service, click **Publish**.
7. In the **Publish** dialog box, select the gateway(s) you want to publish.
8. Click **Publish**.

To unpublish the virtual service, select the service, select **Gateways**, and click **Unpublish**.

Endpoint Management

Multiple Endpoints

In an SOA environment, tracking and management of service endpoints is a key task. Not only is your SOA environment likely to have many individual services, but many of those services are deployed on multiple endpoints.

In general, a service has multiple endpoints for the following reasons:

- It acquires additional endpoints as it moves through its lifecycle. For example, by the time a service goes into production, it is usually available at three different endpoints in your environment: at development endpoint, test endpoint, and production endpoint. To track the endpoints that a service acquires as it moves through its development cycle, add the endpoints to the metadata for the service in the registry.
- It has different endpoints to accommodate the needs of different consumer applications. For example, you may offer the same service over multiple entry protocols (for example, both JMS and HTTP) and with different security mechanisms. In CentraSite, you accommodate these needs by exposing the service over multiple virtual services.

Endpoint information is used by developers who write programs that bind to services.

At run time, a consumer application can perform a static bind or a dynamic bind to a service.

- When a consumer application is written to use static binding, the program binds to a specified address at run time (that is, an endpoint that is already known by the program). When a developer creates a program that uses static binding, he or she gets the endpoint of the service from CentraSite at design time. Typically, this end point is added to a configuration file or a parameter setting that the consumer program reads at run time. If the actual endpoint of the service is modified, the consumer program must be configured to access the service at its new endpoint.
- When a consumer application is written to use dynamic binding, the program looks up the service's address at run time and binds to that address. To use this type of binding, a developer must write a query to retrieve the service's endpoint from CentraSite. At run time, the consumer program executes the query and binds to the endpoint that the query returns. If the endpoint of the service changes, one only has to update the service's endpoint information in CentraSite. Nothing has to be changed in the consumer application.

Representation of Service Endpoints in CentraSite Business UI

Within CentraSite Business UI, the endpoints for a service are shown in the **Provider Overview** profile on the Service Details page. In this panel, an endpoint is represented as a specific **Access URI** (that is, address where the service is deployed).

Representation of Service Endpoints in CentraSite Control

Within CentraSite Control, the endpoints for a service are shown in the **Operations** panel on the service's **Summary** profile. In this panel, an endpoint is represented as a **Binding** that identifies a specific **Access URI** (that is, address where the service is deployed).

The following example shows the **Operations** panel for a service that is deployed at two endpoints: the endpoint represented by the **ExpenseReporting_DEV** binding and the endpoint represented by the **ExpenseReporting_TEST** binding. Note that the **Access URI** column provides the exact address of each endpoint:

The service shown here is deployed at two endpoints, as represented by the two bindings listed in the **Operations** panel.

Operations		
Name	Binding	Access URI
GetReport	ExpenseReporting_DEV	http://DEva_A16:5333/AP/ExpenseReportingService.asmx
CreateNewReport	ExpenseReporting_DEV	http://DEva_A16:5333/AP/ExpenseReportingService.asmx
UpdateReport	ExpenseReporting_DEV	http://DEva_A16:5333/AP/ExpenseReportingService.asmx
GetReport	ExpenseReporting_TEST	http://TEva_N50:5333/AP/ExpenseReportingService.asmx
CreateNewReport	ExpenseReporting_TEST	http://TEva_N50:5333/AP/ExpenseReportingService.asmx
UpdateReport	ExpenseReporting_TEST	http://TEva_N50:5333/AP/ExpenseReportingService.asmx

The **Binding** and **Access URI** information that appears in the **Operations** panel is derived from the <port> definitions in the service WSDL. Specifically, the **Binding** name is derived from the name of the port and the **Access URI** is derived from the port's <address> element. The Bindings on the Operations panel are derived from the <port> definitions in the service WSDL as described in the following image:

```

    .
    .
    <usdl:service name="ExpenseReportingService">
      <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">Provides services for..
      <usdl:port name="ExpenseReporting_DEV" binding="tns:ExpenseReportingSoap">
        <soap:address location="http://DEva_A16:5333/AP/ExpenseReportingService.asmx" />
      </usdl:port>
      <usdl:port name="ExpenseReporting_TEST" binding="tns:ExpenseReportingSoap">
        <soap:address location="http://TEva_N50:5333/AP/ExpenseReportingService.asmx" />
      </usdl:port>
    </usdl:service>
  </usdl:definitions>

```

Operations

Name	Binding	Access URI
GetReport	ExpenseReporting_DEV	http://DEva_A16:5333/AP/ExpenseReportingService.asmx
CreateNewReport	ExpenseReporting_DEV	http://DEva_A16:5333/AP/ExpenseReportingService.asmx
UpdateReport	ExpenseReporting_DEV	http://DEva_A16:5333/AP/ExpenseReportingService.asmx
GetReport	ExpenseReporting_TEST	http://TEva_N50:5333/AP/ExpenseReportingService.asmx
CreateNewReport	ExpenseReporting_TEST	http://TEva_N50:5333/AP/ExpenseReportingService.asmx
UpdateReport	ExpenseReporting_TEST	http://TEva_N50:5333/AP/ExpenseReportingService.asmx

Managing Native Service Endpoints During its Lifecycle

When a native service moves through its lifecycle, it usually gains additional endpoints. For example, during development, a developer generally deploys the service somewhere in the development environment. When the service moves to the testing phase, it is generally deployed at another endpoint for testing. Finally, when the service is placed in production, the operations organization deploys the service at an endpoint in the production environment.

Each time you deploy a native service to an additional endpoint, you must add the new endpoint to the service in CentraSite. To do this you:

1. Download the service WSDL from the CentraSite registry.
2. Add the endpoint to the WSDL (as an additional port definition).
3. Reattach the updated WSDL to the service in CentraSite.

When you attach the updated WSDL to the service, CentraSite automatically updates the binding information on the service's **Operations** panel.

Note:

Although it is possible to represent the development, test, and production endpoints as individual services in the registry, Software AG recommends that you avoid doing this. Such an approach produces a large amount of duplicated metadata and does not return any real benefits. Instead, maintain just one catalog entry for a native service and add the service endpoints to this entry as the service progresses through its lifecycle.

Bindings for a Service in a Single-Stage Registry

If you are using a single-stage deployment of CentraSite, services remain in the same registry for their entire lifecycle. Therefore, in this type of registry, the catalog entry for a service includes the service's development, test, and production endpoints.

The following shows an example of a service that has endpoints in the development, test, and production environments. Note that the naming scheme that has been used to identify the bindings for this service clearly indicates the environment in which the endpoint resides. In a single-stage environment, the development, test, and production endpoints are listed for the service as described in the following image:

Operations		
Name	Binding	Access URI
GetReport	ExpenseReporting_DEV	http://DEva_A16:5333/AP/ExpenseReportingService.asmx
CreateNewReport	ExpenseReporting_DEV	http://DEva_A16:5333/AP/ExpenseReportingService.asmx
UpdateReport	ExpenseReporting_DEV	http://DEva_A16:5333/AP/ExpenseReportingService.asmx
GetReport	ExpenseReporting_TEST	http://TEva_N50:5333/AP/ExpenseReportingService.asmx
CreateNewReport	ExpenseReporting_TEST	http://TEva_N50:5333/AP/ExpenseReportingService.asmx
UpdateReport	ExpenseReporting_TEST	http://TEva_N50:5333/AP/ExpenseReportingService.asmx
GetReport	ExpenseReporting_PROD	http://PRva_X26:5333/AP/ExpenseReportingService.asmx
CreateNewReport	ExpenseReporting_PROD	http://PRva_X26:5333/AP/ExpenseReportingService.asmx
UpdateReport	ExpenseReporting_PROD	http://PRva_X26:5333/AP/ExpenseReportingService.asmx

Important:

The endpoints for a service are visible to any user who has View permission on the service in CentraSite. To prevent unauthorized access to the services themselves, ensure that appropriate security measures are in place at these endpoints.

Example of the Endpoints for a Service in a Two-Stage Registry

In a multi-stage deployment, the set of endpoints that you publish to the registry varies according to needs of the registry's audience. In a two-stage deployment, for example, you would list the service's development and test endpoints on the creation CentraSite and you would list the service's test and production endpoints on the consumption CentraSite.

Example of the bindings you would see if you were to view a service in the creation CentraSite and in the consumption CentraSite.

The creation CentraSite will hold the endpoints in development and test...

Operations		
Name	Binding	Access URI
GetReport	ExpenseReporting_DEV	http://DEva_A16:5333/AP/ExpenseReportingService.asmx
CreateNewReport	ExpenseReporting_DEV	http://DEva_A16:5333/AP/ExpenseReportingService.asmx
UpdateReport	ExpenseReporting_DEV	http://DEva_A16:5333/AP/ExpenseReportingService.asmx
GetReport	ExpenseReporting_TEST	http://TEva_N50:5333/AP/ExpenseReportingService.asmx
CreateNewReport	ExpenseReporting_TEST	http://TEva_N50:5333/AP/ExpenseReportingService.asmx
UpdateReport	ExpenseReporting_TEST	http://TEva_N50:5333/AP/ExpenseReportingService.asmx

...and the consumption CentraSite will hold the endpoints in production and test...

Operations		
Name	Binding	Access URI
GetReport	ExpenseReporting_PROD	http://PRva_X26:5333/AP/ExpenseReportingService.asmx
CreateNewReport	ExpenseReporting_PROD	http://PRva_X26:5333/AP/ExpenseReportingService.asmx
UpdateReport	ExpenseReporting_PROD	http://PRva_X26:5333/AP/ExpenseReportingService.asmx
GetReport	ExpenseReporting_TEST	http://TEva_N50:5333/AP/ExpenseReportingService.asmx
CreateNewReport	ExpenseReporting_TEST	http://TEva_N50:5333/AP/ExpenseReportingService.asmx
UpdateReport	ExpenseReporting_TEST	http://TEva_N50:5333/AP/ExpenseReportingService.asmx

Naming Convention for Binding Names

When a service has multiple endpoints, the binding names give users a hint as to the endpoint's function. As a best practice, consider adopting a naming convention for bindings that identifies service endpoints in a clear and consistent manner. (This practice is especially important if your consumer applications queries the registry to obtain a service endpoint run time). Each binding name includes a suffix to indicate the environment in which the endpoint resides.

Binding names are derived from the port names in the service WSDL, hence, to produce bindings whose names conform to the particular naming scheme that you have adopted, you must assign the appropriate names to port definitions in the WSDL.

For example, to produce the binding names shown in the single-stage example described in [“Bindings for a Service in a Single-Stage Registry” on page 1474](#), the port definitions in the WSDL must look as follows. Naming conventions for bindings must be applied to the port names in the WSDL:

```

:
<wsdl:service name="ExpenseReportingService">
  <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">Services for expense reports. </documentation>
  <wsdl:port name="ExpenseReporting_DEV" binding="tns:ExpenseReportingSoap">
    <soap:address location="http://DEva_A16:5333/AP/ExpenseReportingService.asmx"></soap:address>
  </wsdl:port>
  <wsdl:port name="ExpenseReporting_TEST" binding="tns:ExpenseReportingSoap">
    <soap:address location="http://TEva_N50:5333/AP/ExpenseReportingService.asmx"></soap:address>
  </wsdl:port>
  <wsdl:port name="ExpenseReporting_PROD" binding="tns:ExpenseReportingSoap">
    <soap:address location="http://PEva_X26:5333/AP/ExpenseReportingService.asmx" />
  </wsdl:port>
</wsdl:service>
:

```

Managing Endpoints of a Virtual Service During its Lifecycle

When you create a virtual service, CentraSite generates a WSDL file for the virtual service. Initially, Mediator generates this WSDL file as an abstract WSDL file, and is represented as an empty WSDL in the `<protocol>://` format. However, when you deploy the virtual service, CentraSite replaces the port definitions in this WSDL file with a port definition that specifies the virtual service's endpoint on the Mediator. At this time, it also updates the binding information that appears on the virtual service's **Operations** panel.

CentraSite automatically updates the port definitions in the virtual service WSDL and regenerates the corresponding bindings any time you deploy, undeploy, or redeploy the virtual service.

As the endpoint information for virtual services is generated and updated by CentraSite, you must not manually edit the endpoint information for virtual services. In other words, unlike native services, you must not manually add endpoints to the WSDL of a virtual service. Instead, simply allow CentraSite to generate and manage the endpoints for the virtual services that you deploy.

Publishing the Test Endpoint for a Virtual Service on the Consumption Registry

In the consumption CentraSite, the catalog entry for a native service provides consumers with bindings to the test instance of the service and to the production instance of the service. However, with a *virtual service*, you cannot do this. The set of bindings for a virtual service are generated and managed automatically by CentraSite, and you cannot manually add bindings to this set.

If you have a test instance of a virtual service deployed on the Mediator in your test environment, and you would like to disclose that endpoint to users when they view the virtual service in the registry, you can identify the endpoint in a separate attribute (that is, as additional metadata) within the virtual service on the consumption CentraSite.

Note:

If you have consumer applications that dynamically bind to a virtual service, be aware that those applications have to bind against the *creation CentraSite* during the testing phase of their development and against the *consumption CentraSite* when they enter the production phase.

Deploying Multiple Virtual Services for a Single Native Service

Often you may want to deploy a service on multiple endpoints to make the service available over multiple transports and security mechanisms. For example, you may want to extend the same service over JMS and HTTP transports or you may want to allow internal users to access a service using basic HTTP user name or password credentials and you may require other users to submit digital certificates.

To accommodate these kinds of operational requirements for a native service, you deploy multiple virtual services for a single native service. For example, to make a particular native service available to consumers over both HTTP and JMS, you would create two virtual services for the native service: one that accepts requests over HTTP and another that accepts requests over JMS. Both virtual services would route requests to the same native service at the back end.

The following shows a registry in which a native service (SalesReportingService) has been exposed over two virtual services. Virtual Services provide two transports for one native service:

One native service...
...exposed over two transport protocols using virtual services...

<input type="checkbox"/>	 SalesReportingService	Service	1.0
	Provides services for submitting, revising, and viewing Sales reports.		
<input type="checkbox"/>	 SalesReportingService_VS_JMS	Virtual service	1.0
	Provides services for submitting, revising, and viewing Sales reports.		
<input type="checkbox"/>	 SalesReportingService_VS_HTTP	Virtual service	1.0
	Provides services for submitting, revising, and viewing Sales reports.		

Note:

To make it easier to manage virtual services, consider adopting a naming convention like the one shown in this section. Doing so makes it easier to identify virtual services and the native service with which they are associated. The names of virtual services cannot contain spaces or special characters (except `_` and `-`). Consequently, if you adopt a convention that involves using the name of the native service as part of the virtual service name, then the names of the native services themselves must not contain characters that are invalid in virtual service names.

Using the Asset Navigator Tool to Find the Virtual Services for a Native Service

When you create a virtual service, CentraSite establishes a relationship between the virtual service and the native service from which you created it. If you create multiple virtual services for a native service, each of the virtual services has an established relationship to the native service.

The relationships that CentraSite creates between a native service and a virtual service enables you to use the Asset Navigator feature to easily navigate and visualize a native service and quickly locate all of its virtual services. Example of the Asset Navigator tool lists the virtual services associated with a native service.

**Important:**

To ensure that a relationship is established between a native service and virtual service, always use the **Search for Endpoint** button to set the **Route To** address in a virtual service's **Routing Protocols** processing step. Do not manually type this address into the **Route To** field. If you type the address manually, the relationship to the appropriate native service is not be created.

Configuring the API Gateway Synchronization Settings

You can specify the interval to refresh the following details of APIs that are displayed in generic profiles of CentraSite Business UI:

- List of Endpoints defined for an API in API Gateway.
- Status (Active/ Inactive) of an API in API Gateway.
- List of access tokens and applications defined for an API in API Gateway.

➤ To configure the API Gateway synchronization settings

1. Open the customization file, *centrasite.xml*, in a rich text editor.

You can find the **centrasite.xml** file on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\custom\conf`.

2. Locate the `APIGatewaySyncSettings` section in the file.

```

<APIGatewaySyncSettings>
  <!-- Undelivered Endpoints and API Status (Activated/Deactivated) will be polled
  periodically
  from API Gateway -->
  <!-- Interval duration is minutes -->
  <!-- Default polling interval is 60 minutes -->
  <FetchingInterval>60</FetchingInterval>
</APIGatewaySyncSettings>
  
```

3. Specify the interval, in minutes, to synchronize in the `FetchingInterval` property. The default value of this property is 60 minutes.
4. Save and close the file.
5. Restart Software AG Runtime for the change to take effect.

Runtime Events and Key Performance Indicator (KPI) Metrics

CentraSite can receive run-time events and Key Performance Indicator (KPI) metrics from gateways such as webMethods Mediator. A run-time event is an event that occurs while services are actively deployed on a gateway. Examples of run-time events are:

- Successful or unsuccessful SOAP and REST requests.
- Policy violation events that are generated upon violation of service's runtime policy.
- Service monitoring events that are generated by the service-monitoring actions in the runtime policy.

KPI metrics are used to monitor the run-time execution of virtual services. Metrics include the maximum response time, average response time, fault count, availability of virtual services, and so on. If you include run-time monitoring actions in your run-time policies, the actions monitors the KPI metrics for virtual services and send alerts to various destinations when user-specified performance conditions for a service are violated.

CentraSite provides predefined event types that can be used with any supported runtime gateway, such as webMethods Mediator. In addition, you can also create custom event types.

The runtime event data are collected by the gateway and published to CentraSite through SNMP. The gateway publishes runtime data for all instances.

You can view the runtime events and metrics for all targets, for a particular target, or for a particular virtual service using both CentraSite Business UI and CentraSite Control.

Runtime Event Types

The types of run-time events that Mediator can publish are:

Event Type	Description
Lifecycle	A Lifecycle event occurs each time Mediator is started or shut down.
Error	An Error event occurs each time an invocation of a virtual service results in an error.
Policy Violation	A Policy Violation event occurs each time an invocation of a virtual service violates a run-time policy that was set for the virtual service.
Transaction	A Transaction event occurs each time a virtual service is invoked (successfully or unsuccessfully).
Monitoring	Mediator publishes key performance indicator (KPI) metrics, such as the average response time, fault count, and availability of all virtual services.

The Key Performance Indicator (KPI) Metrics

For the Monitoring event type, the types of KPI metrics that Mediator can publish are:

Metric	Reports on...
Availability	The percentage of time that a virtual service was available during the current interval. A value of 100 indicates that the service was always available. Only the time when the service is unavailable counts against this metric. If invocations fail due to policy violations, this parameter could still be as high as 100.
Average Response Time	The average amount of time it took the service to complete all invocations in the current interval. This is measured from the moment Mediator receives the request until the moment it returns the response to the caller.
Fault Count	The number of failed invocations in the current interval.
Maximum Response Time	The maximum amount of time it took the service to complete an invocation in the current interval.
Minimum Response Time	The minimum amount of time it took the service to complete an invocation in the current interval.
Successful Request Count	The number of successful service invocations in the current interval.
Total Request Count	The total number of requests for each service running in Mediator in the current interval.

Note:

By default, Average Response Time, Minimum Response Time, and Maximum Response Time do not include metrics for failed invocations. You can include metrics for failed invocations by setting the `pg.PgMetricsFormatter.includeFaults` parameter to true. For more information on advanced settings, see *Administering webMethods Mediator*.

The Event Notification Destinations

Mediator can publish data about the run-time events and metrics to the following destinations:

- An SNMP server. You can use any one or both servers:
 - CentraSite's SNMP server, which uses SNMPv3 user-security model.
For the procedure to configure Mediator to send SNMP traps to the CentraSite SNMP server, see *Administering webMethods Mediator*.
 - A third-party SNMP server, which uses either the SNMPv1 community-based security model or the SNMPv3 user-based security model.

For the procedure to configure Mediator to send SNMP traps to a third-party SNMP server, see *Administering webMethods Mediator*.

- An API Portal Destination.
- An Elasticsearch Destination.
- An EDA/Database destination. Mediator can publish events and KPI metrics to an EDA endpoint or a database (that is, a JDBC connection pool associated with the function alias named, Mediator). You have to define the JDBC connection pool in the Integration Server.

For the procedure to configure Mediator to send this data to an EDA/Database destination, see *Administering webMethods Mediator*.

Destinations for Monitoring and Transaction Events

For the Monitoring and Transaction event types, in addition to the SNMP, Elasticsearch, and EDA/Database destinations, there are additional event notification destinations to select from.

Monitoring events are generated by the following run-time actions that you can configure for your virtual services in CentraSite:

- Monitor Service Performance
- Monitor Service Level Agreement
- Throttling Traffic Optimization

Transaction events are generated by the run-time action Log Invocations.

Destinations for Monitoring and Transaction events are as follows and you selects these destinations when you configure your virtual services in CentraSite :

- An EDA/Database destination
- An Elasticsearch destination
- The CentraSite SNMP server or a third-party SNMP server
- An API Portal destination

The additional destinations for monitoring and transactional events are:

SMTP Email

To specify an SMTP email destination, you must:

- Select Email as a destination when you configure the run-time actions.
- Set the Email Configuration parameters in Integration Server (go to **Solutions > Mediator > Administration > Email**) as described in *Administering webMethods Mediator*.

The Integration Server's Local Log

To specify the Integration Server's local log as a destination, you must:

- Select Local Log as a destination when you configure the run-time actions. When configuring actions, you must also specify the severity of the messages to be logged (the logging level).
- Set the Integration Server Administrator's logging level for Mediator to match the logging levels specified for the run-time actions (go to **Settings > Logging > Server Logger**) as described in *Administering webMethods Mediator*. For example, if a Log Invocation action is set to the logging level of Error, you must also set Integration Server Administrator's logging level for Mediator to Error. If the action's logging level is set to a low level (Warning-level or Information-level), but Integration Server Administrator's logging level for Mediator is set to a higher level (Error-level), then only the higher-level messages are written to the log file.

Entries posted to the local log are identified by a product code of MED.

The Integration Server's Audit Log

You can select the Integration Server Audit Log as a destination for the Log Invocation action only. If you expect a high volume of invocations in your system, it is recommended that you select the Audit Log destination. For more information, see *webMethods Audit Logging Guide*.

Managing Collection of Metrics

webMethods Mediator collects performance data (for example, average response time, total request count, fault count) for the virtual services that it hosts. It publishes this data to the destinations (as an example, CentraSite, API Portal, Elasticsearch, and EDA/Database) at regular intervals. When you install and configure the Mediator, you must specify whether you want it to collect performance data and, if so, how often you want it to publish the data to CentraSite.

We recommend that you always enable the collection of performance data on your Mediator. A publication interval of 15 minutes is appropriate for most environments. However, if Mediator handles a very high volume of traffic, consider increasing this interval to 30 or 60 minutes. CentraSite stores the performance data that it receives from the Mediator in the performance log. You can look at the performance information for a particular virtual service by viewing its **Runtime Metrics** profile in CentraSite Business UI, and **Performance** profile in CentraSite Control.

The performance data that CentraSite collects from Mediator can cause the log to grow quite rapidly. When the log grows very large, queries to the log can significantly affect CentraSite's performance. To prevent this from happening, we suggest that you routinely purge old entries from the log.

CentraSite provides a log-purging utility that you can use to automatically purge the log on a scheduled basis. We suggest that you use this utility to keep no more than one month of performance data in the log (adjust this recommendation as necessary to accommodate your particular needs). When you configure the log-purging utility, you can specify whether you want to delete the purged log entries or export them to an archive file (in case you want to retain them for future reference).

Note:

The performance metrics that Mediator collects enable service consumers (and potential service consumers) to determine whether a virtual service is performing at a required level. However, Mediator does not collect data at the granularity that a network administrator would need in order to analyze performance problems (for example, to determine why the response time for a particular virtual service drops at a particular time of day).

Managing Collection of Events

In addition to performance metrics, webMethods Mediator can also log *event data*. Event data supplies information about activities or conditions that occur on Mediator.

Mediator logs two basic kinds of events: 1) data relating to the operation of Mediator itself and 2) data relating to the execution of virtual services.

The event data that Mediator collects about itself are referred to as *lifecycle events*. These events represent activities or conditions that occur during the general operation of Mediator. Lifecycle events are reported for the completion of significant processes (for example, Mediator server start-up and shut down). Mediator logs lifecycle events if you configure it to do so.

Mediator collects the following types of event data relating to the execution of virtual services. Be aware that Mediator does not collect this type of information automatically. If you want to capture these types of events, you must deploy run-time policies to do so:

- *Transaction Events* report information about the requests that the Mediator processes. This type of event is produced by the execution of a logging action in a run-time policy. For example, you might configure a run-time policy to log all of the request and response messages submitted to a particular virtual service.
- *Monitoring Events* report transgressions relating to performance metrics. This type of event is produced by the execution of a monitoring action in a run-time policy. For example, you might configure a run-time policy to report occasions when the response time for a virtual service exceeds a specified threshold.

Mediator publishes event data as Simple Network Management Protocol (SNMP) traps. When you install Mediator, you can configure it to publish this data to CentraSite, to another third-party SNMP server or to both. If you select to log event data to CentraSite, you can view the events using the CentraSite Control user interface.

Like the performance log, the event log grows larger over time. If it becomes very large, queries to the log causes performance issues with CentraSite. To manage the size of the event log, we suggest that you occasionally purge old entries from it. As a general guideline, consider maintaining only three months of event data in the log. (Adjust this recommendation as necessary to accommodate your particular needs. If you routinely log request and response messages, for example, you might need to purge more often.)

You can configure CentraSite's log-purging utility to purge the event log on a scheduled basis. When you configure the purging facility, you can specify whether you want to delete the purged entries or export them to an archive file (in case you want to retain them for future reference).

Using CentraSite with Other Policy Enforcement Points

Instead of (or in addition to) using webMethods Mediator for mediation and policy enforcement, you can use other third-party products with CentraSite. Support for third-party policy-enforcement and run-time governance tools is available through integrations that are provided by members of the CentraSite Community. These tools are made available through the CentraSite Community Web site at Software AG TECHcommunity Website .

The Metrics Tracking Interval

Mediator tracks performance metrics by intervals. Interval is a period of time you set in Mediator, during which metrics are collected for reporting to CentraSite. You set the interval in the Publish Interval field on the **Mediator > Administration > CentraSite Communication** page in the Integration Server Administrator. For details, see *Administering webMethods Mediator*.

Mediator only tracks metrics for the current interval. At the end of the interval, Mediator aggregates the metrics and reports them to CentraSite. Once the metrics are reported, Mediator resets its counters for the new interval. Mediator does not calculate and aggregate metrics across intervals. If Mediator is shut down or the virtual service is undeployed before the current interval expires, the performance data is discarded.

Note:

To avoid the need for Mediator to store metrics during periods of inactivity, Mediator stores only first and last zero value metrics that occurs during an interval, and discards the remaining consecutive zero value metrics. Doing this reduces the storage space consumed by the metrics, and speeds the queries you perform in the dashboard. Skipping the in-between zero metrics does not affect in the performance graphs shown in the dashboard.

For more information about the metrics tracking interval, see *Administering webMethods Mediator*.

CentraSite Configuration to Receive Run-Time Events and Metrics

Prerequisites:

- Ensure that Mediator is configured for publishing events to an SNMP server, as described in the *Administering webMethods Mediator* guide.
- If you use a target type other than Mediator or webMethods Insight, ensure that you configure CentraSite to publish events by providing the MIB file in your target type's definition file, as described in [“Gateway Management” on page 1364](#). (CentraSite provides a MIB file for Mediator and Insight Server.)
- Modify CentraSite's default settings for logging run-time events. By default, CentraSite logs all predefined event types, but you may disable any type.

CentraSite provides an Event Receiver, which is a data collector that collects the run-time event data. The Event Receiver listens for run-time events from the target instances through the SNMP

(Application-Layer) protocol, and contains the logic to parse and store event data in the Event Receiver's data store.

Components of the Event Receiver

The Event Receiver contains the following components:

■ The SNMP Listener

CentraSite's SNMPv3 Trap Listener that supports SNMP4J. This Listener starts automatically when CentraSite starts.

■ The Intermediate Queue

The queue from the SNMP Listener to the Event Processor. This queue decouples the SNMP Listener threads from the Event Processor to improve throughput. The modes supported are:

- **FileSystem:** Incoming Traps are stored temporarily in the file system.
- **InMemory:** Incoming Traps are stored temporarily in memory.
- **NoQueue:** Incoming Traps are not be stored in any intermediate queue. The SNMP Listener threads are processed.

■ The Event Processor

The Event Processor (SOALinkSNMPEventsListener) transforms incoming SNMPv3 Traps into an XML file (Events.xml) that complies with the schema in the RuntimeEvents Collection component. The Event Processor transforms an SNMPv3 Trap to the Events.xml file as follows:

1. Determines the Event Type (and Target Type) to which the Trap belongs, and gets the corresponding UUIDs. This involves searching all Event Type-to-Trap mappings in all the defined target types, using the Trap's OID. Since this is an expensive search, the Event Type-to-Trap mapping is cached to improve performance.
2. Parses the Trap attributes and obtains the Service (UUID), the Target (Name), the TimeStamp, and the SessionId. The processor then searches the registry or repository and obtains the corresponding UUID for the Target Name. This mapping is also cached to improve performance.
3. Collects the remaining attributes from the Trap.
4. Constructs the Events.xml file using the Event Type UUID, Target Type UUID, Service UUID, Target UUID, TimeStamp, SessionId, and other collected attributes.

■ The Batch Condition

The Batch Condition is a set of OR conditions used by the Event Processor. The Event Processor supports two modes of event storage into CentraSite: BatchMode and NoBatchMode. BatchMode is available only for FileSystem and InMemory queues. When BatchMode is enabled, the Event Processor continues to accumulate Events.xml documents until one of the conditions is evaluated as true. Then it inserts all the documents as a single batch into CentraSite.

■ The RuntimeEvents Collection

The run-time events are stored in the RuntimeEvents Collection as non-registry objects.

Configuring the Event Receiver

The Event Receiver is bundled in the installation as a web-application named SOALinkSNMPEventsListener supporting the JavaEE standard. The web.xml configuration file contains all the Event Receiver configuration properties. The web.xml configuration file for the Event Receiver is in the `<CentraSiteInstall_Directory>/cast/cswebapps/SOALinkSNMPEventsListener/WEB-INF` directory.

Note:

Restart CentraSite after setting the Event Receiver configuration properties in the web.xml configuration file.

Set Database Configuration Properties

In the Event Receiver's configuration file, set the following properties related to the RuntimeEvents Collection database:

Database Property

Description

`com.softwareag.centrasite.soalink.events.dbUrl` The URL of the RuntimeEvents Collection database. All run-time events persist to this database.

`com.softwareag.centrasite.soalink.events.dbUserId` The user name that the Events Listener used for authentication before persisting event data to the RuntimeEvents Collection database. The default value of this property is the predefined user EventsUser.

Optionally, you can change the value EventsUser to any login user who has the following privileges:

- Write access on the Tamino collection RuntimeEvents.
- Read access on TargetTypes, Targets, RuntimeEventTypes, and LogUnit, which are under the Tamino collection CentraSite.

If you want to change the value to a login user, type that login user's name in the form `<hostName>\<userName>`.

Important:

The predefined password of EventsUser is EventsManager4CS (there is no need to specify the password in this file). If you want

Database Property	Description
	to change this password or if you have changed the value EventsUser to a login user, <i>you must change the password</i> . Whenever you change the password, you must restart CentraSite.
<code>com.softwareag.centrasite.soalink.events.db.NonActivityTimeOut</code>	The non-activity timeout in seconds for the RuntimeEvents Collection database (default 2592000 seconds (30 days)).

Set SNMPv3 Transport Configuration Properties

In the Event Receiver's configuration file, set the following properties related to the RuntimeEvents Collection database:

SNMPv3 Transport Property	Description
<code>com.softwareag.centrasite.soalink.events.snmp.transport</code>	Wire transport protocol that is used by the SNMP Listener. Supported values are: TCP and UDP.
<code>com.softwareag.centrasite.soalink.events.snmp.host</code>	The CentraSite host name or IP address to which the SNMP listener binds.
	<p>Note: If the machine on which CentraSite server is installed only supports IPv4, then the <code>com.softwareag.centrasite.soalink.events.snmp.host</code> property must be configured to point to the hostname of the machine. The default value works fine for the dual IP stacks.</p>
<code>com.softwareag.centrasite.soalink.events.snmp.port</code>	The port to which the SNMP listener binds. The default is 8181. If Microsoft Internet Information Services (IIS) is installed (or can be installed) on the same machine hosting Integration Server/Mediator, then you may want to change the default SNMP port of 8181 to something else, to avoid any potential runtime conflicts when sending SNMP packets.
<code>com.softwareag.centrasite.soalink.events.snmp.inboundMsgSizeBytes</code>	Maximum inbound message size in bytes (an integer). Traps that exceed this size limit is rejected. Default value is 256Kb.

SNMPv3 Transport Property	Description
<code>com.softwareag.centrasite.soalink.events.snmp.dispatcherPoolSize</code>	The SNMP Listener's Worker-Thread pool size (default is 10). This determines the throughput of the Listener.

Setting the SNMPv3 USM Configuration Properties

In the Event Receiver's configuration file, set the following properties related to SNMPv3 USM:

SNMPv3 USM Property	Description
<code>com.softwareag.centrasite.soalink.events.snmp.engineId</code>	EngineId to be used by the SNMP Listener. If the parameter is left blank, the SNMP Listener auto-generates the engineId.
<code>com.softwareag.centrasite.soalink.events.snmp.securityName</code>	The SecurityName to be used by the SNMP Listener.
<code>com.softwareag.centrasite.soalink.events.snmp.securityLevel</code>	The Maximum SecurityLevel to be supported by SNMP Listener. Supported values in order are: NOAUTH_NOPRIV, AUTH_NOPRIV, and AUTH_PRIV. For example, AUTH_PRIV provides the highest level of security but also supports the other two levels. Similarly, AUTH_NOPRIV supports NOAUTH_NOPRIV.
<code>com.softwareag.centrasite.soalink.events.snmp.authProtocol</code>	Authorization protocol to be used by the SNMP Listener for decoding the incoming trap. Supported values are: MD5 and SHA.
<code>com.softwareag.centrasite.soalink.events.snmp.authPassPhraseKey</code>	The PassPhrase key to be used by the Authorization protocol. The PassPhrase key length should be ≥ 8 . The key is stored in this file; the PassPhrase value is stored securely in passman.

Note:

The value should be the same as the authPassphrasKey that you have set in the `web.xml` file. You can also use the `CentraSiteCommand.cmd` tool to

SNMPv3 USM Property**Description**

reset this key at a later stage as required.

`com.softwareag.centrasite.soalink.events.snmp.privProtocol` The Privacy protocol to be used by the SNMP Listener for decoding the incoming trap. Supported values are: DES, AES128, AES, AES192, AES256, 3DES, and DESEDE.

`com.softwareag.centrasite.soalink.events.snmp.privPassPhraseKey` The PassPhrase key to be used by the PrivacyProtocol. The PassPhrase length should be ≥ 8 . The key is stored in this file; the PassPhrase value is stored securely in passman.

Note:

The value must be the same as the PassphrasKey that you have set in the `web.xml` file. You can also use the `CentraSiteCommand.cmd` tool to reset this key at a later stage as required.

Setting the Events Queue Implementation Property

In the Event Receiver's configuration file, set the following property related to the implementation of the events queue:

Events Queue Property**Description**

`com.softwareag.centrasite.soalink.events.eventsQueueImpl`

Supported values are:

- **FileSystem:** Incoming Traps are stored temporarily in the file system
- **InMemory:** Incoming Traps are stored temporarily in memory
- **NoQueue:** Incoming Traps are not stored in any intermediate queue and the SNMP Listener threads are processed one by one.

Setting the Properties for FileSystem or InMemory

When the `eventsQueueImpl` property is set to either `FileSystem` or `InMemory`, you must also set the following properties:

FileSystem or InMemory Property	Description
<code>com.softwareag.centrasite.soalink.events.enableBatchInsertion</code>	<p>Enables or disables batch insertion of events into the database. Supported values are true and false.</p> <p>If true, events are batched according to the batching rules properties and the batch is stored to the database. If false, events are stored to the database one by one.</p>
<code>com.softwareag.centrasite.soalink.events.maxNumOfEventsPerBatch</code>	<p>Maximum number of events in a batch. This must be an integer value. A value ≤ 0 disables this rule. This rule is evaluated only on arrival of a new Trap.</p>
<code>com.softwareag.centrasite.soalink.events.maxSizeOfBatch</code>	<p>Maximum size (in bytes) of a batch. Default value is 512KB. This must be an integer value. A value ≤ 0 disables this rule. This rule is evaluated only on arrival of a new Trap.</p>
<code>com.softwareag.centrasite.soalink.events.maxTimeIntervalBetweenBatches</code>	<p>Maximum time interval (in milliseconds) between two subsequent batch storages. This must be an integer value. A value ≤ 0 disables this rule. Unlike the other two rules, this rule is evaluated periodically. Hence this rule prevents any trap stuck in the batch for ever if inflow of traps stops (acts as a batch-timeout). A very low value for this rule reduces batch efficiency and introduces unnecessary looping.</p>
<code>com.softwareag.centrasite.soalink.events.fileSystemQueueDir</code>	<p>(Only applies when the <code>eventsQueueImpl</code> property is set to <code>FileSystem</code>.) The directory that must be used as <code>FileSystem Queue</code>. Incoming traps are stored in this directory temporarily and hence should have write permission. The path can be absolute or relative. It is advisable to provide the absolute path. Relative paths are considered relative to one of the following, based on availability in the same order:</p> <ol style="list-style-type: none">1. <code>SOALinkSNMPEventsListener/WEB-INF</code> directory for exploded deployments.

FileSystem or InMemory Property	Description
	2. javax.servlet.context.tempdir for zipped deployments.
	3. java.io.tmpdir if none of the above are available.

Event Type Modeling

Event types are modeled as registry objects. The String, Date, Integer, and Boolean event attributes are stored in the registry or repository as slots. The File-Type attributes (representing payloads or binary-data) are stored as HasExternalLink associations. For example, consider the predefined event type such as Transaction, if you go to the **Target Type** details page, you see the Transaction event type attributes (which are obtained from the webMethodsESB.mib file) as follows:

Attribute Name	Object ID	Type
Service	1.3.6.1.4.1.1783.201.1.1.1	String
Target	1.3.6.1.4.1.1783.201.1.1.2	String
Timestamp	1.3.6.1.4.1.1783.201.1.1.3	Date
Consumer	1.3.6.1.4.1.1783.201.1.1.4	String
RequestStatus	1.3.6.1.4.1.1783.201.1.1.5	String
ResponsePayload	1.3.6.1.4.1.1783.201.1.1.6	File
RequestPayload	1.3.6.1.4.1.1783.201.1.1.7	File
ProviderRoundTripTime	1.3.6.1.4.1.1783.201.1.1.8	Integer
TotalRoundTripTime	1.3.6.1.4.1.1783.201.1.1.9	Integer
SessionID	1.3.6.1.4.1.1783.201.1.1.16	String
ConsumerIP	1.3.6.1.4.1.1783.201.1.1.17	String
OperationName	1.3.6.1.4.1.1783.201.1.1.21	String
NativeEndpoint	1.3.6.1.4.1.1783.201.1.1.22	String

All these attributes except the File-Type attributes, RequestPayload, and ResponsePayload are stored as registry object slots as follows:

Slot Key	Slot Type	Slot Value (Attribute)
uddi_16d34470-9a92-11dd-9b43-e319c2a6593c	xs:string	Service
uddi_f18b5a40-9a91-11dd-b95e-b4758b17b88b	xs:string	Target

Slot Key	Slot Type	Slot Value (Attribute)
uddi_c798d3c0-9a91-11dd-889e-b999c87ba6b7	xs:datetime	TimeStamp
uddi_a7476ff0-a108-11dd-9c38-d8fd010529cc	xs:string	Consumer
uddi_a7476ff0-a108-11dd-9c38-eac6d60fc855	xs:string	RequestStatus
uddi_a7476ff0-a108-11dd-9c38-f3f84c6111f0	xs:integer	ProviderRoundTrip Time
uddi_a7476ff0-a108-11dd-9c38-d02170b3aae3	xs:integer	TotalRoundTripTime
uddi_21b67010-9a92-11dd-926a-991c4c180c79	xs:string	SessionID
uddi_a7476ff0-a108-11dd-9c38-d34f346cb3d5	xs:string	ConsumerIP
uddi_f1c8a185-4b18-4974-a360-6c70756a174a	xs:string	OperationName
uddi_524d05f5-d526-4605-b594-ace1cb750d33	xs:string	NativeEndpoint

The File-Type attributes, ResponsePayload, and RequestPayload are stored as HasExternalLink associations as follows:

Association Key	Association Name (Attribute)
uddi:a747704b-a108-11dd-9c38-fde9d932116a	ResponsePayload
uddi:a745265b-a108-11dd-9c38-bf43eee17363	RequestPayload

The Target Type to Event Type Association Object

A target type is associated with an event type (represented as a registry object) by a Target Type to Event Type Association object, which defines the UUID to MIB OID mapping.

The table shows the contents of a sample object that associates the target type webMethods Mediator with the event type Transaction. The table's columns are described as follows:

- **Attribute:** The Attribute column is not part of the object, it is included here for your reference.
- **Slot Key:** Contains the UUID that is obtained from the event type registry object.
- **Slot Type:** Contains the slot type that is obtained from the event type registry object.
- **Slot Value:** Contains the event type attribute's Object Identifier (OID) that is obtained from the MIB file.

Attribute	Slot Key (Event Type UUID)	Slot Type	Slot Value (Event Attribute OID)
Service	uddi_16d34470-9a92-11dd-9b43-e319c2a6593c	xs:string	1.3.6.1.4.1.1783.201.1.1.1
Target	uddi_f18b5a40-9a91-11dd-95eb-4758b1788b	xs:string	1.3.6.1.4.1.1783.201.1.1.2
TimeStamp	uddi_c798d3c0-9a91-11dd-89eb-199c87ba67	xs:datetime	1.3.6.1.4.1.1783.201.1.1.3
Consumer	uddi_a7476ff0-a108-11dd-9c38-d8fd010529cc	xs:string	1.3.6.1.4.1.1783.201.1.1.4
RequestStatus	uddi_a7476ff0-a108-11dd-9c38-eac6d60fc855	xs:string	1.3.6.1.4.1.1783.201.1.1.5
ResponsePayload	uddi_a747704b-a108-11dd-9c38-fde9d932116a	xs:anyURI	1.3.6.1.4.1.1783.201.1.1.6
RequestPayload	uddi_a745265b-a108-11dd-9c38-bf43eee17363	xs:anyURI	1.3.6.1.4.1.1783.201.1.1.7
ProviderRound TripTime	uddi_a7476ff0-a108-11dd-9c38-f3f84c6111f0	xs:integer	1.3.6.1.4.1.1783.201.1.1.8
TotalRoundTrip Time	uddi_a7476ff0-a108-11dd-9c38-d02170b3aae3	xs:integer	1.3.6.1.4.1.1783.201.1.1.9
SessionID	uddi_21b67010-9a92-11dd-926a-991c4c180c79	xs:string	1.3.6.1.4.1.1783.201.1.1.16
ConsumerIP	uddi_a7476ff0-a108-11dd-9c38-d34f346cb3d5	xs:string	1.3.6.1.4.1.1783.201.1.1.17
OperationName	uddi_f1c8a185-4b18-4974-a360-6c70756a174a	xs:string	1.3.6.1.4.1.1783.201.1.1.21
NativeEndpoint	uddi_524d05f5-d526-4605-b594-ace1cb750d33	xs:string	1.3.6.1.4.1.1783.201.1.1.22

Event Modeling

An event is an instance of an event type. Events are modeled in a separate schema from the event type schema. CentraSite models events as non-registry objects (to avoid storing large amounts of unwanted event data in the registry or repository) and instead stores event data in a database collection within the Event Receiver. CentraSite maps events to their corresponding event types using the event types' UUIDs. Similarly, events are mapped to target types, targets and services using UUIDs, and the event type attributes.

The stored event data contains:

- The event Trap ID (MIB OID).
- The event Trap value that consists of:
 - The attribute key (MIB OID).
 - The attribute value.

The event data is stored in the Event Receiver as an events doctype.

If an event contains payloads (for example, File-Type attributes such as ResponsePayload and RequestPayload), the payloads are stored in the Event Receiver as a payloads doctype, and is referenced by the event stored under the event doctype, using ino:id. This is used to reduce de-serialization of the usually large payloads and to improve performance of queries on the stored events.

Managing Runtime Events and Metrics through CentraSite Business UI

This section describes operations you can perform to manage the runtime events and metrics through CentraSite Business UI.

Displaying Event Information for Assets (APIs)

Pre-requisites:

The following general guidelines apply when displaying the events:

- Ensure that webMethods Mediator is configured to collect and report runtime events.

You must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > CentraSite Communication**). For procedures, see *Administering webMethods Mediator*.
- Ensure that CentraSite is configured to receive runtime events from Mediator.
- Ensure that you have at least an instance-level View permission on the **Runtime Events** profile of the asset.

- Ensure that the type's definition (for example, Virtual Service) includes the **Runtime Events** profile to view runtime event information for its asset instance.

The **Events** profile displays the runtime events of an asset. You can filter the list by target, event type and time period.

The **Events** profile contains information about runtime events that have occurred in a target (that is, a policy-enforcement point (PEP) or a runtime monitoring component).

The target publishes to CentraSite the runtime events that have occurred (assuming that the target type contains a MIB file in its target type definition file).

CentraSite provides predefined event types for use with Mediator, or any third-party policy-enforcement point (PEP), or runtime monitoring component such as Insight that is integrated with CentraSite.

➤ To display the runtime events of an asset (API)

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:

- Click the **Browse** link that is located in the upper-left corner of the menu bar.
- Click the **Search** icon that is located next to the **Scope** list. The default search scope is **Assets**.

A list of defined assets in CentraSite is displayed in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Virtual Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the list of asset types, select **Virtual Service**.
 - c. Click **OK**.

A list of defined Virtual Service assets is displayed in the Search Results page.

5. Click the Virtual Service whose runtime events you want to examine.

This opens the Virtual Service details page.

6. Select the **Runtime Events** profile.
7. Use the following fields to filter the event list you want to view:

In this field...	Specify...
Gateway	<p>A gateway to which the asset is deployed or select All to view the event information of all gateways to which the API is deployed.</p> <p>CentraSite displays None by default.</p>
Consumer	<p>A consumer of the asset or select All to view the run-time event information of all consumers of the asset.</p> <p>CentraSite displays All by default. However, if you do not have at least one consumer registered in the registry, CentraSite displays None by default.</p>
Event Type	<p>A particular event type or select All to view all event types.</p> <p>CentraSite displays All by default.</p>
Date Range	<p>A range of dates from which to view the events (for example, Last 1 hour, Last 12 hours, Last 1 day, Last 5 days, Last 10 days, Last 20 days, Last 1 month, Custom, and so on).</p> <p>CentraSite displays Last 1 month by default.</p>
Start Date/End Date	<p>If chosen Custom in the previous field, then the time period for which to view the metrics.</p> <p>Start Date: Click the calendar and select a starting date and time.</p> <p>End Date: Click the calendar and select an ending date and time.</p>
Display Interval	<p>A running count events of the service displayed at regular time intervals.</p> <p>The interval is specified in the format 3m 2d 6h; wherein m indicates the month, d indicates the day and h indicates the hour.</p>

8. Click **Refine**.
9. Expand the **Graphical** node to display a graphical view of the run-time event information.
10. Expand the **Tabular** node.

CentraSite displays a tabular view of the event information in the left pane.

Field	Description
Date/Time	The date/time that the event occurred. Click this hyperlinked value to view the Event Detail page, which will contain the event's SOAP request or response name in the Attribute column. Click the hyperlinked request or response name to display the full SOAP request or response.

Field	Description
Event Type	(Read-only). The type of event (for example, Monitoring, Policy Violation, Error, and so on).
Gateway	(Read only). The gateway on which the event occurred.

11. To access the details of an event, click on the link for the event.

The **Event Details** dialog in the right pane shows a detailed information about the event that you select in the left pane.

Displaying Performance Metrics for Assets (APIs)

Pre-requisites:

The following general guidelines apply when displaying the Key Performance Indicator (KPI) metrics:

- Ensure that webMethods Mediator is configured to collect and report runtime metrics.
You must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > CentraSite Communication**). For procedures, see *Administering webMethods Mediator*.
- Ensure that CentraSite is configured to receive runtime metrics from Mediator.
- Ensure that you have at least an instance-level View permission on the **Runtime Metrics** profile of the asset.
- Ensure that the type's definition (for example, Virtual Service) includes the **Runtime Metrics** profile to view runtime metric information for its asset instance.

Gateways capture runtime metrics for assets. If you are using the Mediator gateway, Mediator's data collector captures KPI metrics for each asset and publishes them to CentraSite at regular intervals. If you are using a runtime monitoring component such as Insight, the monitoring component captures the KPI metrics of all rogue assets and publishes them to CentraSite at regular intervals.

Note:

If you receive a Javascript error when trying to display the **Performance** profile, please install the latest versions of the Adobe Flash Player/Shockwave Player plug-ins on your Microsoft Internet Explorer.

➤ To display the runtime performance of an asset (API)

1. In CentraSite Business UI, access the Advanced Search panel in one of the following ways:
 - Click the **Browse** link that is located in the upper-left corner of the menu bar.

- Click the **Search** icon that is located next to the **Scope** list. The default search scope is **Assets**.

A list of defined assets in CentraSite is displayed in the Search Results page.

2. In the **Additional Search Criteria** list, select **Asset Types**.
3. To search for the assets of type, Virtual Service, click **Choose**.
4. In the **Choose Asset Types** dialog box, select the **Assets** option button, and follow these steps:
 - a. Click the chevron next to **Assets** option button.
 - b. In the list of asset types, select **Virtual Service**.
 - c. Click **OK**.

A list of defined Virtual Service assets is displayed in the Search Results page.

5. Click the Virtual Service whose runtime metrics you want to examine.

This opens the Virtual Service details page.

6. Select the **Runtime Metrics** profile.
7. Expand the **Filters** node.
8. Specify the exact set of attributes you want to use to filter the metrics list.

In this field...	Do the following...
Gateway	Select a particular gateway to which the virtual service is published, or select All to view the metrics of all gateways to which the virtual service is published. CentraSite displays None by default.
Date Range	Specify a range of dates to view the metrics for the virtual service. For example, Last 1 hour, Last 12 hours, Last 1 day, Last 5 days, Last 10 days, Last 20 days, Custom. CentraSite displays Last 10 days by default.
Start Date/End Date	If you have selected the Custom option in the previous field, specify the time period to view the metrics list. Start Date: Click the calendar and select a starting date and time. End Date: Click the calendar and select a ending date and time.

In this field...	Do the following...
Display Interval	Specify the running count metrics of the virtual service at the displayed time intervals. The interval is specified in the format 3m 2d 6h; wherein "m" indicates the month, "d" indicates the day and "h" indicates the hour.

9. Click **Refine**.

CentraSite displays a graphical view of the run-time metrics for all performance categories as shown below:

- **Multi-line Chart.** The chart shows the Minimum Response Time, Maximum Response Time, and Average Response Time of the API.
- **Pie Chart.** The chart shows the Success Request Counts, Total Request Counts, and Fault Counts of the API.
- **Gauge Chart.** The chart shows the availability of the API.

Managing Runtime Events and Metrics through CentraSite Control

This section describes operations you can perform to manage the runtime events and metrics through CentraSite Control.

Displaying Event Information for Targets

Pre-requisites:

If you are using the Mediator target, ensure that Mediator is configured to send event notifications to the destination(s) that are applicable for each event type. For details, see *Administering webMethods Mediator*.

You must have the permissions to manage targets.

➤ To view a list of runtime events for targets

1. In CentraSite Control, go to **Operations > Events > Event List**.
2. Provide the following information to filter the event list:

In this field...	Specify...
Target Type	The type of the target whose events you want to view.

In this field...	Specify...
Target	The target whose events you want to view (or select All to view events of all targets).
Event Type	A particular event type, or select All to view all event types.
Service Type	Select All or Virtual Service . Note: CentraSite does not provide out-of-the-box policy-enforcement for web services.
Date Range	A range of dates from which to view the events.
Start Date	Alternatively, select the check box next to this field and click the calendar and select a starting date and time.
End Date	Click the calendar and select an ending date and time.

3. Click **Search**.

The generated event list displays the following information:

Field	Description
Date/Time	The date and time that the event occurred. Click this hyperlinked value to view the Event Detail page that contains the event's SOAP request or response name in the Attribute column. Click the hyperlinked request or response name to display the full SOAP request or response.
Session ID	(Read-only). The session ID that generated the event.
Event Type	(Read-only). The type of event (for example, Monitoring, Policy Violation, Error, and so on).
Service Name	(Read-only). The name of the service that caused the event.
Service Type	(Read-only). The service's type.
Target	(Read-only). The target on which the event occurred.
Target Type	(Read-only). The type of the target on which the event occurred.

Displaying Event Information for Assets

Pre-requisites:

The following general guidelines apply when displaying the events:

- Ensure that webMethods Mediator is configured to collect and report runtime events.

You must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > CentraSite Communication**). For procedures, see *Administering webMethods Mediator*.

- Ensure that CentraSite is configured to receive runtime events from Mediator.

The **Events** profile displays the runtime events of an asset. You can filter the list by target, event type and time period.

The **Events** profile contains information about runtime events that have occurred in a target (that is, a policy-enforcement point (PEP) or a runtime monitoring component).

The target publishes to CentraSite the runtime events that have occurred (assuming that the target type contains a MIB file in its target type definition file).

CentraSite provides predefined event types for use with Mediator, or any third-party policy-enforcement point (PEP), or runtime monitoring component such as Insight that is integrated with CentraSite. In addition, you can create custom event types using CentraSite Control.

Users with proper permissions can perform the following additional tasks:

- View a log of all runtime events that have occurred in a particular gateway or in all gateways.
- Create and manage custom run-time event types for use with Mediator.

➤ To display the runtime events of an asset

1. In CentraSite Control, go to **Asset Catalog > Browse**.
2. In the **Assets** pane, right-click an asset whose runtime events you want to examine, and click **Details**.

You can also select multiple assets, click the **Actions** menu, and click **Details**.

3. Click the **Events** tab.
4. In the **Events** profile, type the appropriate information for each of the displayed fields:

In this field...	Do the following...
Target Type	The target type (for example, Mediator or a run-time monitoring component such as Insight).
Target	The target name.
Event Type	A particular event type or select All to view all event types.
Date Range	A range of dates from which to view the events.
Start Date	Alternatively, select the check box next to this field, click the calendar and select a starting date and time.

In this field...	Do the following...
End Date	Click the calendar and select an ending date and time.

- Click **Search**.

CentraSite displays the following information for each of the event:

Field	Description
Date/Time	The date and time that the event occurred. Click this hyperlinked value to view the Event Detail page, which will contain the event's SOAP request or response name in the Attribute column. Click the hyperlinked request or response name to display the full SOAP request or response.
Session ID	(Read-only). The session ID that generated the event.
Event Type	(Read-only). The type of event (for example, Monitoring, Policy Violation, Error, and so on.).
Service Name	(Read-only). The name of the service that caused the event.
Service Type	(Read-only). The service's type.
Target	(Read-only). The target on which the event occurred.
Target Type	(Read-only). The type of the target on which the event occurred.

Displaying Performance Metrics for Assets

Pre-requisites:

The following general guidelines apply when displaying the Key Performance Indicator (KPI) metrics:

- Ensure that webMethods Mediator is configured to collect and report runtime metrics.
You must configure Mediator to communicate with CentraSite (in the Integration Server Administrator, go to **Solutions > Mediator > Administration > CentraSite Communication**). For procedures, see *Administering webMethods Mediator*.
- Ensure that CentraSite is configured to receive runtime metrics from Mediator.

Gateways capture runtime metrics for assets. If you are using the Mediator gateway, Mediator's data collector captures KPI metrics for each asset and publishes them to CentraSite at regular intervals. If you are using a runtime monitoring component such as Insight, the monitoring component captures the KPI metrics of all rogue assets and publishes them to CentraSite at regular intervals.

Note:

If you receive a Javascript error when trying to display the **Performance** profile, please install the latest versions of the Adobe Flash Player/Shockwave Player plug-ins on your Microsoft Internet Explorer.

➤ **To display the runtime performance of an asset**

1. In CentraSite Control, go to **Asset Catalog > Browse**.

A list of currently defined asset types is displayed in the **Types** pane.

2. Select the **Virtual Service** check box.

A list of the Virtual Service assets are displayed in the **Assets** pane.

3. In the **Assets** pane, right-click an asset whose runtime metrics you want to examine, and then click **Details**, or select the check boxes for multiple assets, click the **Actions** menu, and then click **Details**.

4. Click the **Performance** tab.

5. To switch between the tabular view or graphical view of the runtime metrics, click the **Switch to** button.

6. In the Tabular view of the runtime metrics, type the appropriate information for each of the displayed fields:

In this field...	Do the following...
Select Target	Specify a target of the asset, or select All to view the metrics of all targets to which the virtual service is deployed.
Start Date/End Date	Specify the time period to view the metrics.

7. Click **Search**.

The table displays the asset's performance metrics in the available categories (Success Request Count, Total Request Count, Fault Count, and so on.).

Creating Custom Run-Time Events

You must have the Manage Runtime Event Types permission. By default, the predefined CentraSite Administrator role and Operations Administrator role include this permission.

Important:

To enable CentraSite to recognize custom event types, ensure that your MIB file (which is contained in your target type definition file) contains the SNMP Traps metadata and Object Identifiers for the custom events.

> To create custom event types

1. In CentraSite Control, go to **Operations > Events > Event Types** to display the **Event Types** page.

The page displays all the predefined event types (Monitoring, Policy Violation, Transaction, Error, and Lifecycle) and any custom event types that have been defined.

2. To view the details of any event type, click its hyperlinked name.

The list of attributes for the event type is displayed. You can edit the attributes of custom event types but not the predefined event types.

3. To create a custom event type, click **Add Event Type**. In the **Add/Edit Event Type** page specify a name and description for the event type. Event type names can contain any character (including spaces) and are not case-sensitive.
4. In the Event Type Attribute panel, the following default attributes are displayed. These attributes are required and cannot be deleted:

Attribute	Data Type
TimeStamp	Date
Target	String
Service	String
SessionID	String

To create additional attributes, perform the following steps:

- a. Click the plus button at the bottom of the attribute list.
- b. Specify a name in the **Name** column and a value in the **Data Type** column (Boolean, File, Date, Integer, or String). Attribute names can contain any character (including spaces).
- c. To add another attribute, click the plus button at the bottom of the list.
- d. Click **Save**.

Modifying Run-Time Events

You can modify the custom run-time events as required:

> To modify a custom run-time event

1. In CentraSite Control, go to **Operations > Events > Event Types**.

The **Event Types** page displays all event types that have been defined.

Note:

To delete a custom event type, select the check box next to the event type and click **Delete**.

2. To edit the attributes of a custom event type:
 - a. Click its hyperlinked name to display the **Add/Edit Event Type** page.
 - b. You can modify the value of an attribute's data type but not its name. Data types can be Boolean, File, Date, Integer, or String.
 - c. To add another attribute, use the plus button at the bottom of the list.
 - d. To delete an attribute, click the minus button next to the attribute.
 - e. Click **Save**.

Managing Runtime Event Store

CentraSite offers you an enhanced storage mode, also called as `flat`, that requires a minimal disk space in the database and provides a very fast and reliable storage for the run-time events.

Beginning with version 10.0, CentraSite is configured to use the flat storage mode by default. You can reconfigure CentraSite to use the legacy storage mode at any point in time.

Note:

If you migrated from a previous version of CentraSite to CentraSite 10.0, then the default configuration is the legacy storage mode.

You can use the command line interface to manage the run-time event storage in CentraSite. You can use this tool to perform the following tasks:

- Check the status of the current active run-time event store.
- Switch the mode of the run-time event store.
- Remove the data content of the active run-time event store.

Managing Runtime Event Store Through the Command Line Interface

This section describes operations you can perform to manage the runtime event store through CentraSite Command Line Interface (CLI).

Fetching State of Run-Time Event Store

Pre-requisites:

To fetch the current state of an active run-time event store through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `state runtimeEvent` for this purpose.

> To fetch the current state of run-time event store

- Run the command `state runtimeEvent`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd state runtimeEvent -user <USER-ID> -password <PASSWORD> [-url <CENTRASITE-URL>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .

The possible values for this command:

- `ok` - Indicates the legacy mode of run-time event storage.
- `flat` - Indicates the enhanced mode of run-time event storage.

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd state runtimeEvent -user Administrator -password manage -url http://localhost:53307/CentraSite/CentraSite
```

The response to this command could be:

```
Executing the command : state runtimeEvent
=====
<?xml version="1.0" encoding="UTF-8" ?>
<ino:response xmlns:ino="http://namespaces.softwareag.com/tamino/response2"
xmlns:xql="http://metalab.unc.edu/xql/">
  <ino:message ino:returnValue="0">
    <ino:mesageline>starting admin command
      ino:RuntimeEvents(&#39;state&#39;);
    </ino:mesageline>
  </ino:message>
  <RuntimeEvents>
    <Type>flat</Type>
    <Count>0</Count>
  </RuntimeEvents>
  <ino:message ino:returnValue="0">
    <ino:mesageline>admin command
      ino:RuntimeEvents(&#39;state&#39;); completed
    </ino:mesageline>
  </ino:message>
</ino:response>
=====
...
Successfully executed the command : state runtimeEvent
```

Changing the Storage Mode of Run-Time Event

Pre-requisites:

To change the current storage mode of run-time events through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

In certain circumstances, you may want to change the legacy storage mode of run-time events to the enhanced storage mode (and vice versa).

CentraSite provides a command tool named `switch runtimeEvent` for this purpose.

➤ To change the current storage mode of run-time events

- Run the command `switch runtimeEvent`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd switch runtimeEvent -user <USER-ID> -password <PASSWORD> [-url <CENTRASITE-URL>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd switch runtimeEvent -user Administrator -password manage -url http://localhost:53307/CentraSite/CentraSite
```

The response to this command could be:

```
Executing the command : switch runtimeEvent
=====
<?xml version="1.0" encoding="UTF-8" ?>
<ino:response xmlns:ino="http://namespaces.softwareag.com/tamino/response2"
xmlns:xql="http://metalab.unc.edu/xql/">
  <ino:message ino:returnValue="0">
    <ino:messageline>starting admin command
      ino:RuntimeEvents(&#39;switch&#39;)&#39;
    </ino:messageline>
  </ino:message>
  <ino:message ino:returnValue="0">
    <ino:messageline>admin command
      ino:RuntimeEvents(&#39;switch&#39;) completed
    </ino:messageline>
  </ino:message>
```

```
</ino:response>
=====
...
Successfully executed the command : switch runtimeEvent
```

Note:

If you want to check whether the current event storage mode is modified, execute the command `state runtimeEvent`.

Purging the Run-Time Events

Pre-requisites:

To purge events through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

You may want to remove unwanted data of an active run-time event store. In such cases, it may be necessary to purge the events in the run-time event store.

CentraSite provides a command tool named `purge runtimeEvent` for this purpose.

> To purge run-time events

- Run the command `switch runtimeEvent`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd purge runtimeEvent -user <USER-ID> -password <PASSWORD> [-url <CENTRASITE-URL>]`

The input parameters are:

Parameter	Description
CENTRASITE-URL	(Optional). The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
USER-ID	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
PASSWORD	The password for the CentraSite user identified by the parameter <code>USER-ID</code> .

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>CentraSiteCommand.cmd purge runtimeEvent -user Administrator -password manage -url http://localhost:53307/CentraSite/CentraSite
```

The response to this command could be:

```
Executing the command : purge runtimeEvent
Successfully executed the command : purge runtimeEvent
```

Managing Runtime Event Store Through Indexes and Aggregates

Runtime events can accumulate in a large volume of data in a very short time. With the size of database increasing exponentially, being able to query the large amount of collected runtime data and return results quickly with minimal time is crucially important. Also, when retrieving, purging, and archiving old runtime data from the database, some queries might take a long period of time to execute causing inconvenience to the customers.

To accelerate the query time for runtime data, CentraSite offers the *Customer index* and *Aggregate data* capabilities.

Consumer Index

The Customer index makes queries run faster and more efficiently. The Customer index results in a improved query performance compared to using no index.

Customer index is defined for the collection `RuntimeEvents`, which collects and stores event data in a database collection within the Event Receiver. The Consumer index field gets automatically filled with the event data collection. The index, however, can only be established if the runtime storage mode is `flat`. Queries that want to make use of that faster search mechanism must use the new index explicitly. This is, query expressions as

To make your queries run much faster, you must use the Consumer index explicitly.

To use the Consumer index, you must manually update the query expression:

```
declare namespace re='http://namespaces.CentraSite.com/Schema/SOALink'
let $consumerKey = " ... "
let $eventsForConsumer := collection('RuntimeEvents')/re:events[re:eventsDetails
[re:attributeKey = 'uddi_a7476ff0-a108-11dd-9c38-d8fd010529cc' and re:attributeValue
= $consumerKey]]
...
```

After you make the required changes, the query expression would look like the following:

```
declare namespace re='http://namespaces.CentraSite.com/Schema/SOALink'
let $consumerKey = " ... "
let $eventsForConsumer := collection('RuntimeEvents')/re:events[?]
...
```

Note:

The Consumer index mechanism will not retrieve event data if the value of the attribute `$consumerKey` is empty, for example, `$consumerKey = .` However, the mechanism will relieve event data if the value of the attribute `$consumerKey` is set to `unknown`.

Aggregate Data

If the acceleration in query performance using the Consumer index does not satisfy and further purging is not possible, you can base queries on aggregates of event data. CentraSite aggregates data by default.

There are two types of aggregate event data:

- Event data, which reports the frequency count of events every hour.

- Event data, which reports the frequency count of events every day.

Note:

The reporting aggregate event data for every hour and every day depends on the local timezone.

Each aggregate data contains a collection of events within the specified time period concerning a specific triplet of service, target, and consumer.

To make use of the aggregates, event queries can be redefined to be based upon aggregates instead of the original runtime data. Redefining such a query means to read documents of the doctypes `rea:aggregateEvents1` (for aggregates on an hourly basis) and `rea:aggregateEvents2` (for aggregates on an hourly basis) instead of documents of the doctype `events`, all in the collection `RuntimeEvents`. Instead of collecting events within a certain time period, aggregates are summed and totally contained within the time period. Events that are not included by these aggregates are added separately.

To use the aggregate data on an hourly basis, you must manually update the query expression:

```
declare namespace re='http://namespaces.CentraSite.com/Schema/SOALink';
...
let $events := collection('RuntimeEvents')/re:events [re:timestamp ge $startTime
and re:timestamp lt $endTime]
```

The above query expression should be replaced to look like the following:

```
declare namespace re='http://namespaces.CentraSite.com/Schema/SOALink';
...
let $aggregatedEvents := collection('RuntimeEvents')/re:aggregateEvents1
[@re:creationTime ge $startTime and @re:creationTime lt $endTime]
  $firstHour := xs:dateTime(concat(substring-before($fromString,":"),":00:00")) +
  xs:dayTimeDuration('PT1H'),
  $eventsBefore := collection('RuntimeEvents')/re:events [re:timestamp ge
  $startTime and re:timestamp lt xs:dateTime($firstHour)]
  $lastHour := xs:dateTime(concat(substring-before(string($to),":"),":00:00")),
  $eventsAfter := collection('RuntimeEvents')/re:events [re:timestamp ge
  $lastHour and re:timestamp lt , $to]
  return ($eventsBefore, $aggregatedEvents, $eventsAfter)
```

To use the aggregate data on a daily basis, the query needs to be more complex, since hours up to the first complete day within the specified time period, complete days within the time period, and hours after the last complete day up to the end-time requires to be added up to a fixed sum. This would make the query look like the following:

```
declare namespace re='http://namespaces.CentraSite.com/Schema/SOALink';
...
declare function local:aggregateEvents($fromString as xs:string,
                                       $toString as xs:string) {
  let $from := xs:dateTime($fromString),
      $to := xs:dateTime($toString),
      $firstDay := xs:date(substring-before($fromString,"T"))
                    + xs:dayTimeDuration("P1D"),
      $lastDay := xs:date(substring-before(string($to),"T")),
      $firstHour := xs:dateTime(concat(
                            substring-before($fromString,":"),":00:00"))
                    + xs:dayTimeDuration('PT1H'),
      $lastHour := xs:dateTime(concat(
                            substring-before(string($to),":"),":00:00")),
```

```

$eventsBefore := collection('RuntimeEvents')/re:events [re:timestamp ge
                                                    $from and re:timestamp lt
xs:dateTime($firstHour)]
$hoursBefore := collection('RuntimeEvents')/re:
                aggregateEvents1[@re:creationTime ge $firstHour and
@re:creationTime lt xs:dateTime($firstDay)]
$eventsInDays := collection('RuntimeEvents')/re:
                aggregateEvents2[@re:creationTime ge $firstDay and
@re:creationTime lt $lastDay]
$hoursAfter := collection('RuntimeEvents')/re:
                aggregateEvents1[@re:creationTime ge xs:dateTime($lastDay) and
@re:creationTime lt $lastHour]
$eventsAfter := collection('RuntimeEvents')/re:events
                [re:timestamp ge $lastHour and re:timestamp lt ,$to]
return if ($to lt $from)
    then ()
    else ($eventsBefore, $hoursBefore, $eventsInDays,
        $hoursAfter, $eventsAfter)
}

```

You can change the data aggregation settings by adding or modifying the attribute `EventProcessingSchedulingSettings` that is listed below `GUIConfiguration` in the **centrasite.xml** configuration file. You can find the **centrasite.xml** on `<CentraSiteInstall_Directory>\cast\cswebapps\BusinessUI\custom\conf`.

Default setting for the attribute `EventProcessingSchedulingSettings` is:

```

<EventProcessingSchedulingSettings>
  <EventProcessingInterval1>60</EventProcessingInterval1>
  <EventProcessingInterval2>24</EventProcessingInterval2>
</EventProcessingSchedulingSettings>

```

This default setting denotes that aggregates are built for hours and days.

Important:

Be aware that changing the values of this default setting will lead to incorrect search results for queries that combine aggregates produced with different aggregation settings.

Monitoring Logs

CentraSite offers logging and monitoring capabilities across different services.

CentraSite collects and displays log information of runtime events and performance metrics, policy details, approval history, and auditable events.

Viewing Policy Log

To view the policy log, you must have the View Policy Log permission.

The Policy Log contains information about the design/change-time policies that CentraSite has executed. By default, CentraSite only logs information about policies that fail. However, you can optionally configure CentraSite to log information about policies that resulted in success, informational, warning, and failure alerts.

> To view the policy log

1. In CentraSite Control, go to **Administration > Logs > Policy Log**.
2. To define the kind of log entry you want to view, specify an appropriate information for each of the displayed fields:

In this field... Do the following...

Object Name *Optional.* A pattern string that describes the names of the objects (of **Object Type**) whose log entries you want to view.

You can provide the exact name or use a pattern string consisting of a character sequence and the % wildcard character (which represents any string of characters). For example, if you specify the pattern string 'A%', CentraSite displays entities whose names start with 'A'.

Leave **Entity Name** empty to view all names.

Policy Type The type of policy whose log entries you want to view. To view the log entries for design/change-time policies, select **Design/Change Time** from the drop-down list (if it is not already selected).

Object Type The object type whose log entries you want to view.

Event Type The event type whose log entries you want to view.

Policy Status The policy execution status that you want to view. A policy's execution status is the result set of each of its action's execution result. CentraSite writes the following policy execution status to the policy log depending on the log configuration:

Icon	Description
	<i>Success.</i> Displays policies that have resulted in success alert.
	<i>Info.</i> Displays policies that have resulted in informational alert.
	<i>Inprogress.</i> Displays policies that have resulted in inprogress alert.
	<i>Warning.</i> Displays policies that have resulted in warning alert.
	<i>Failure.</i> Displays policies that have resulted in failure alert.

Execution Date *Optional.* The time period that you want to examine. Leave the **From** and **To** fields empty to view log entries for all dates.

3. Click **Search** to fetch the details of the policy log entries.

- To view details for a particular entry in the returned list, click the name of the policy.

Note:

If a policy included a WS-I action, the log entry for the policy includes a link to the results of the WS-I action.

Viewing Approval History Log

To view the Approval History Log, you must have the `View Approval History` permission.

The Approval History Log contains a record of all approval requests that have been triggered by a policy with an approval workflow action. This log shows the status of each approval request that has been submitted to CentraSite.

➤ To view the Approval History Log

- In CentraSite Control, go to **Administration > Logs > Approval History**.
- To filter your search for the approval log information, specify an appropriate information for each of the displayed fields:

In this field...	Do the following...
Object Name	The object that was submitted for approval.
Approval Flow Name	The name assigned to the approval workflow.
Requestor	Click Select User and select the user who requested the approval.
Approver	Click Select User(s) and select the user(s) who approved the request.
Object Type	Select the object type whose log entries you want to view from the drop-down list.
Request Date From/To	Use the calendar to specify a date range for the requests.
Approval Date From/To	Use the calendar to specify a date range for the approvals.
Status	Select a status of the approval request (for example, In Progress, Approved, and so on).

- Click **Search** to fetch the details of the approval log entries.
- To view the details of an approval workflow, click the hyperlinked value that you see in the **Approval Flow Name** column.

The **Approval Flow Information** panel provides the following information about each approval workflow.

Field	Description
Approval Flow Name	Name of the approval workflow.
Mode	Mode of the approval workflow (for example, Anyone or Everyone).
Status	Status of the approval policy (for example, In Progress, Approved, New, No Action, Pending, or Rejected).
Creation Date	Date when the approval workflow was created.

The **Requestor Summary** panel displays the following information about the requestor of an approval workflow.

Field	Description
Requestor Name	User who triggered the approval workflow.
Approval Type	Name of the approval action (for example, Initiate Approval or Initiate Group-dependent Approval)
Entity	Name of the entity on which the approval was triggered.
Reason for Request	Additional comments or descriptive information stating the reason for the approval request.

The **Approver Summary** panel displays the following information about the designated approver(s) of an approval workflow.

Field	Description
Approver	User who approved or rejected the approval workflow.
Action	Action of the approver (for example, Approved or Rejected).
Comments	Additional comments or descriptive information stating the reason for approval or rejection.

Viewing Audit Log

To view the Audit Log of an asset, you must have View permission on the asset.

An audit log reports the creation or update activities of a particular asset (including changes in an asset's lifecycle state). You can view the audit logs of any asset that you can view.

➤ **To view the audit log of an asset**

1. In CentraSite Control, go to **Asset Catalog > Browse**.

A list of currently defined assets is displayed in the **Assets** pane.

2. In the **Assets** pane, right-click an asset whose audit log data you want to view, and then click **Details**, or select the check boxes for multiple assets, click the **Actions** menu, and then click **Details**.
3. Click the **Audit Log** tab.

The **Audit Log** profile displays the following log information for this asset:

Field	Description
Event Type	The type of event that was executed on the asset (for example, created or updated).
Date/Time	The date and time that the event was executed.
User	The user who executed the event.
Comment	Descriptive information about the event executed on the asset.

Viewing Runtime Events Log

The runtime event log contains information about run-time events that have occurred in a target (that is, a policy-enforcement point (PEP) or a run-time monitoring component).

The target publishes to CentraSite the runtime events that have occurred (assuming that the target type contains a MIB file in its target type definition file). CentraSite provides predefined event types for use with webMethods Mediator or any third-party PEP that is integrated with CentraSite.

You can view the runtime event log of a virtual service that is published to the webMethods Mediator in the **Events** tab (CentraSite Control) and **Runtime Events** profile (CentraSite Business UI).

Viewing Runtime Performance Log

Gateways capture run-time metrics for Virtual Service assets. If you are using the Mediator gateway, Mediator's data collector captures Key Performance Indicator (KPI) metrics for each virtual service and publishes them to CentraSite at regular intervals. If you are using a run-time monitoring component such as Insight, the monitoring component captures the KPI metrics of all rogue assets and publishes them to CentraSite at regular intervals.

You can view the runtime performance of a virtual service that is published to the webMethods Mediator in the **Performance** tab (CentraSite Control) and **Runtime Metrics** profile (CentraSite Business UI).

14 Exporting and Importing Registry Objects

■ Introduction to Export and Import of Registry Objects	1518
■ Exporting and Importing Registry Objects through CentraSite Business UI	1527
■ Exporting and Importing Registry Objects through CentraSite Control	1531
■ Exporting and Importing Registry Objects through Command Line	1536
■ Best Practices: Exporting and Importing Assets	1540

Introduction to Export and Import of Registry Objects

CentraSite enables you to export registry objects from a registry to an archive and then import these registry objects from the archived file into the same registry or to another instance of CentraSite.

Note:

Export and Import feature is designed specifically for exporting selected objects from the registry and not for backing up the entire instance of the CentraSite registry or repository.

You can select the following native CentraSite objects for export:

- Organizations

Note:

By exporting an organization, it is possible to export and import the associated users, groups, roles, and permissions.

- Asset Types (definitions)
- Taxonomies
- Lifecycle Models
- Design-Time Policies
- Assets (instances)
- Run-Time Policies
- Report Templates

In addition, you can select the following asset types belonging to other components of the webMethods Product Suite for export:

- BPM Process Project
- Integration Server Package

Important:

When you export an object, CentraSite generally exports a number of additional objects besides the one you select. The specific set of objects that CentraSite exports with an object varies by object type. After CentraSite completes the export process, it generates a report that identifies each of the objects that it attempted to export and indicates whether or not the object was exported successfully. The export report also identifies all referenced objects that were omitted from the archive file.

Exporting Registry Object

To export an object successfully, you must have view permission on the object that you select for export and all the additional objects that CentraSite exports with that object. This section explains how the export process handles:

- **Object Ownership:** Object's ownership and organizational attributes are *included* with the exported object. However, this data is imported with an object only if the **Keep current owner** and **Keep current organization** options are enabled at import time. However, how to assign the ownership to the imported object is determined by whether the:
 - Imported object replaces an existing object if the object in the archive is newer than the existing object in the registry or the **Allow replace of registry objects** option is used. The existing object is replaced but its **Owner** and **Organization** attributes remain unchanged (the updated object belongs to the same organization and user as it did before it was replaced by the copy from the archive file).
 - Imported object is added to the registry as a new object, its organizational and user ownership is determined by the following parameters:
 - If **Keep current owner** is set in the import dialog and the original owner exists with the same UUID on the target, the objects is long to that user after import. If the user with the same UUID does not exist on the target, the import of the particular object fails.

If **Keep current owner** is *not* set in the import dialog, the importing user is the owner of the object.
 - If **Keep current organization** is set in the import dialog, the original associations of the objects are maintained if the organization exists with the same UUID on the target. If the organization with the same UUID does not exist on the target, the import of the particular object fails.
- **Instance Level Permissions:** The export process *does not* export the instance-level permissions that are assigned to an object.
- **Supporting Documents:** When you export an asset, the asset's supporting documents are also exported with the asset.

Note:

The export feature does not provide a way to export supporting documents or other repository items directly. However, you can use the **Download** button on the Supporting Document Library page to download documents from one instance of CentraSite and then upload them to another.

- **Other Versions of an Object:** If you export an asset that is versioned, the exported asset includes a reference to the *previous version* of the asset. However, the referenced version itself does not be included in the archive.
- **Different Versions of CentraSite:** CentraSite does not restrict the transfer of objects between different product versions. However, if a user-defined type is exported and already exists on the target machine, the import fails if the existing type already has instances and the modification of the type due to the different product version is such that the existing instances do not validate with the type any longer. Typically, enhancements of the type with additional fields do not cause any problems, but modification of existing fields may lead to unexpected results.

Objects included by CentraSite

The additional objects that CentraSite includes in the archive file when it exports an object are determined by the specific export handler that is associated with the exported object's type. Generally, CentraSite includes the following objects with any object that you export:

- The supporting documents that are attached to the object.
- The type definition of the object, if the object type is a custom object type or a modified predefined object type.
- The taxonomy associated with the object type, if the taxonomy is a custom taxonomy or a modified predefined taxonomy.
- The association types associated with the object type, if the association types is a custom association type or a modified predefined association type.

Additionally, if the object is an instance of a composite type, CentraSite also exports the object's components referred by Relationship attributes.

Objects not included by CentraSite

The export set for an object generally contains all objects referenced by an object (referenced standard objects such as predefined asset types are not included, since these can be expected to exist on the target machine). However, you may examine the export report for warnings. If there are warnings and they identify objects that is not present in the target registry at import time (usually indicated by a INMIEW0038 warning in the report), you must either:

- Export the referenced objects separately and import them into the target registry before you import the archive file that you just created.

or

- Use the **Add to List** command to build a list that includes the object that you want to export *and* all the objects that are identified by the INMIEW0038 warnings and export the list. This generates an archive file that contains the exported objects and the objects that it references.

Exporting Objects that Use Custom Associations

CentraSite does not automatically export the referenced objects that are referred by custom associations unless the check box **Include assets referenced by selected assets** is selected. In this case, a warning message is displayed to indicate the missing references. Ensure that such referenced objects are already present in the target registry when importing archives.

Importing Registry Object

You import an object by importing the archive file to which it was previously exported. You can import an object into the same CentraSite registry from which it was originally exported or to a different CentraSite registry. The **Import** dialog displays the contents of the archive to be imported and you can select either the entire archive or just a subset of the objects to import.

Note:

If the archive contains a registry object with references which cannot be accepted during import, the import process continues but this object is not imported.

To import an object, you must have permissions to create that object in the target registry. If the object that is to be imported already exists in the target registry, you must have permission to edit the existing object. If you attempt to import an object but do not have the proper permissions, the particular objects is ignored during the import process.

To import object types, you must have the Asset Type Administrator role. To import organizations or users, you must have the CentraSite Administrator role. When an archive is imported, the importer reads the contents of the archive file and either adds its contents to the target registry (if the object does not already exist) or replaces existing objects with the objects from the archive. The importer verifies the object's Universally Unique Identifier (UUID) to determine whether it already exists. An object is ignored when the same object with an identical timestamp already exists. If objects are identical and the object to be imported has an older timestamp than the one in the registry, the import is rejected unless the **Allow replace of registry objects** option is used.

If you set the **Keep current owner** option in the **Import** dialog and the original owner exists on the target machine, the assets belongs to that user after import. As a further requirement, this user on the target machine must have the same UUID as the original owner. If this user does not exist on the target, the import fails. The **Keep current owner** option is disregarded if a conflicting user is ignored at import. In this case, the import succeeds and the objects belongs to the importing user.

If you set the **Allow replace of registry objects** option in the **Import** dialog, the imported object overwrites the existing object even if it is older than the object in the target registry.

If you set the **Import groups that the user belongs to** option in the **Import** dialog, the importer contains the user to be imported and the groups that the user belongs to. This applies only to user-defined groups. The system-defined groups are not imported.

Object Identity

Each registry object is identified by its unique UUID. CentraSite assigns UUID to an object when it adds the object to the registry. After an object is created, its UUID cannot be changed. You may see an object's UUID displayed in the user interface. When the UUID is displayed for an object, it appears in the object's **Key** attribute. Example of UUID: uddi:2d621948-89f3-11df-851c-a3fb7c9c098e.

During an export or import process the importer uses the UUID to determine whether an imported object already exists in the target registry. If an identical object (an object with an identical UUID) already exists in the target registry, the timestamps of both the existing object and that to be imported determine whether and in which way the importing process continues.

When a user imports an archive and determines that an object in the archive file already exists in the registry, then depending on the timestamps of the objects:

- The imported object has a newer timestamp than the existing one. The existing object is *replaced* and the import executes successfully.
- The timestamps are identical. The object is *ignored* and the import executes successfully.
- The imported object has an older timestamp than the existing one. The object is *rejected* but this does not cause the import process to fail.

If the **Allow replace of registry objects** option has been set in the **Import** dialog, the existing object is overwritten automatically even if the imported object is older than the object in the target registry. Due to this, all objects in the registry depending on it is affected as well.

If you intend to use this functionality to transfer objects between instances of CentraSite, ensure that the system clocks on all of the involved servers are synchronized.

If an object to be imported contains a reference to another object, the importer determines whether the referenced object is available in the archive file or in the registry. If the referenced object is available in the archive file, the importer imports it as necessary to resolve the reference. If the referenced object is not available in the archive file or the target registry, the object containing the reference is not imported.

If an object to be imported includes state information, the importer verifies whether a lifecycle model containing the specified state exists on the target registry. The option **Keep lifecycle state** in the **Import** dialog determines whether the lifecycle state of assets in the archive should be preserved during the import. If this checkbox is marked, the lifecycle state of each imported asset is set to the same value as the asset's lifecycle state in the archive being imported. This operation is available when the lifecycle model itself is in the `productive` or `retired` state.

This operation is only possible for any given asset if the lifecycle model governing the imported asset contains the same lifecycle state as the state in which the asset was originally exported to the archive. If the lifecycle model for the imported asset does not contain the same state as the state of the asset in the archive, the state of the imported asset is set to the initial state of the lifecycle model that governs the imported asset.

If the target registry has no lifecycle model for the type of object that you are importing, the imported object's lifecycle state information, if present in the archive file, is ignored.

If the target registry uses a different lifecycle model than the one used by the imported object, the object's lifecycle state information, if present in the archive file, is ignored and the object enters the initial state of the lifecycle model that is in effect for its type on the target registry.

Important:

If the object you are importing was exported from an instance of CentraSite that has assigned a stage to the object's lifecycle state, the object can only be imported to the registry whose address is specified in that stage.

Policies Triggered During Import

When the import process adds a new object to the registry, CentraSite applies relevant PreCreate and PostCreate policies. If the import process replaces an existing object in the registry, CentraSite triggers the applicable PreUpdate and PostUpdate policies.

Note:

These policies are *not* activated when a complete organization is imported.

Versioned Objects During Import

If an object has multiple versions, on export only the selected objects are exported and not the entire version path. On import of a versioned object CentraSite checks whether a next lower version

number is present in the target registry. If there is a lower version number, the version relationship is retained. Otherwise the object is imported but does not take part of a version path.

Promotion

CentraSite supports configurations where different CentraSite instances are used to represent lifecycle stages or usages. For example, CentraSite can be set up with two instances such as **development** and **production** to represent the different phases in a software development lifecycle. It could also be set up with instances differentiating between asset creation and asset consumption aspects.

Promotion refers to the capability to copy an asset, a lifecycle model, or a policy from one lifecycle stage to another. This is done by exporting the object from its current registry and importing it into the registry that hosts the next phase of its lifecycle.

A stage definition in CentraSite is managed by a lifecycle stage object which describes a CentraSite instance by name and configuration information. Those stages are assigned to lifecycle model states to define the allowed promotion paths for objects in a certain lifecycle state. To promote an object from one lifecycle stage to another, the object is exported from the source registry and then imported in the target registry.

When any object from the source registry is in an end state, one or more registries can be defined to which the object can be imported. An attempt to import it into any other target subsequently fails. While importing, the registry object retains its lifecycle state if its lifecycle model is available in the target registry. If the model is unavailable, the object is set to the initial state in the default lifecycle model.

Exporting and Importing Specific Object Types

Important:

If your site has installed custom export handlers for certain object types, consult your administrator for information about the export sets that they produce.

Organizations

When you export an Organization, the export set consists of:

- The definition of the organization
- The contents of the Supporting Documents library of the organization
- The definition of groups, and roles together with their associated permissions

Additionally, you can optionally export the following objects with the organization:

- Users belonging to the organization
- Lifecycle models
- Policies

- Child organizations with their related objects.
- Assets belonging to the organization.

Note:

- Groups, Roles, and Permissions objects cannot be directly exported. You can only export these kinds of objects by exporting the organization to which they belong.
- If lifecycle models and policies are selected to be exported with the organization, users are automatically selected for export . If lifecycle models and policies are not included in the export, you have the option to select the users for export.
- If the organization is exported with users and a user is a member of one of the organization's groups without being related to the organization itself, then this user is not exported with the organization. This is true even if the user is the Organization Administrator.
- If the organization was exported without users or if the Organization Administrator belongs to a foreign organization, the Organization Administrator is set to the importing user on import, else, the Organization Administrator does not change. An organization's Primary Contact is always the importing user.
- If the replace option was set and the imported organization is already present in the target registry, its associated groups and users are updated. Furthermore, if the export included users, the groups are entirely replaced with the corresponding information of the export set, that is, users may be added to or deleted from the organization and the groups. However, if the export did not include users, the organization and its groups retain their users.
- If an archive with one or more organizations is imported, then the **Keep current organization** option is implicitly switched on, thereby ensuring that all objects are imported to their original organization.
- If the archive with an organization that was created using CentraSite 10.0 or earlier is imported, the newly introduced roles, for example, API Gateway Administrator and API Gateway Publisher are automatically included to the imported organization in the current version of CentraSite.

Users

If an organization with users is imported where the user is already present in the target registry, the import of the user object is ignored and the object is not replaced.

This is also true when only the user name but not its `key` attribute are identical; in this case the conflicting user is ignored at import and a warning message is displayed. The user is removed from its group and any **Keep current owner** settings referring to this user are ignored.

If a user is not already available in the target registry, then the user is imported into the original organization.

Asset Types

When you export an asset type, the export set consists of:

- The type definition.
- The XSD file that contains the type definition.

- Optionally, all instances of the asset type or virtual type.
- The user defined or customized predefined type definitions of all types and association types that are referred to by the asset type or virtual type being exported. This applies recursively, if a referenced asset type also contains such references, then these are also included in the export set. The instances of such referenced types are not included in the export set, even if the option **Include instances** is set.
- Other user defined or customized predefined objects (taxonomies and so on) that are referred to by the type being exported.

Note that there is a compatibility level number stored with the asset type. Therefore, you may not always be able to import the asset type archive in different CentraSite versions.

Taxonomies

When you export a taxonomy, the export set consists of:

- The taxonomy object
- All categories belonging to the taxonomy (the entire dependency tree).
- Repository resources (icons)
- Optionally, all objects classified by categories of this taxonomy.

Lifecycle Models

When you export a lifecycle model, the export set consists of:

- The lifecycle model object
- Optionally, all objects governed by the lifecycle model.
- All references to lifecycle state permissions. During the import of the archive in the target registry, if the user or group referenced in the lifecycle state permission exists in the target registry, then the permissions is saved, otherwise the references is removed during the import.

Design/Change-Time Policies

When you export a design-time policy, the export set consists of:

- The design/change time policy
- Policy actions that the policy uses.

Asset Instances

When you export an instance of an asset, the export set consists of:

- The asset object
- All supporting documents associated with the asset.

- The asset's type definition and the used association type definitions, whether the type is a custom type or a modified predefined type.
- The taxonomy associated with the asset type, whether the taxonomy is a custom taxonomy or a modified predefined taxonomy.
- Optionally, all referenced objects to which the asset has an association.
- For each exported asset instance, the export set contains all referenced asset types and association types. This means that when the export set is imported on the target machine, imported assets have no unsatisfied references.
- Assets that are referenced by the assets that you wish to export (if you select the **Include assets referenced by selected assets** option). If referenced assets themselves reference other assets, those assets are also included. This selection process is repeated until all asset references are satisfied.

Note:

- This option can cause the size of the export set to be very large.
- When you export a Service asset that refers to XML Schemas, the referenced XML Schemas are also exported automatically, provided that you have permission to export them. If you do not have permission to export a referenced XML Schema, the referenced XML Schema is not exported and a warning message is logged.
- When you export a Virtual Service asset that refers to a runtime alias defined with its policy, the alias is also part of the export archive. On import, this alias will be overwritten if the object is not present on the target CentraSite. The alias remains unchanged when the **Allow replace of registry objects** option is set.

If the asset is an instance of a composite asset, the export set includes:

- All components (shared and non-shared) associated with the asset.
- All required objects associated with the asset.

Report Templates

When you export a report template, the export set consists of:

- The report template
- The associated RPT design file from the repository.

BPM Process Project

The asset type `BPM Process Project` is one of the webMethods Product Suite asset types for which CentraSite provides an export function. When you export an instance of a BPM Process Project, the export set consists of:

- The BPM Process Project asset
- The BPM Process Project's type definition, whether the type is a custom type or a modified predefined type.

- All components (shared and non-shared) associated with the asset.
- All required objects associated with the asset.

Integration Server Assets

The asset type `IS Package` is one of the webMethods Product Suite asset types for which CentraSite provides an export function.

When you export an instance of an `IS Package`, the export set consists of:

- The asset.
- The asset's type definition, whether the type is a custom type or a modified predefined type.
- All components (shared and non-shared) associated with the asset.
- IS server assets that are referenced by the assets you wish to export (if you select the option **Include assets referenced by selected Assets**). If the referenced assets reference other assets, those assets are also included. This selection process is repeated until all asset references are attained.

Exporting and Importing Registry Objects through CentraSite Business UI

This section describes how you can perform exporting and importing objects through CentraSite Control.

Exporting Objects through CentraSite Business UI

To export an object, you must have View permission for the object you want to export.

With CentraSite Business UI, you can export one or more objects in each export operation. The export operation creates an archive file on the file system. The archive file consists a copy of the objects that you have exported. The archive file can be then imported into the same CentraSite registry or into a new registry.

Currently, you can do the export operation only for assets in CentraSite Business UI.

➤ To export an asset or a set of assets

1. In CentraSite Business UI, click the **Browse** link that is located in the upper-left corner of the menu bar.
2. In the **Additional Search Criteria** list, select **Asset Types**, and click **Choose**.

This opens the **Choose Asset Types** dialog box.

3. Select **Assets** check box and click **OK**.

This displays a list of defined users in the Search Results page.

4. Select an asset or set of assets you want to export and click **Export** from the action bar.

The **Export** dialog box shows the selected object.

5. Expand the **Advanced Settings** to display a list of the additional export options.

The available options depend on the type of object you wish to export.

Object type	Available export options
-------------	--------------------------

Asset	
-------	--

<ul style="list-style-type: none">■ Include assets referenced by selected assets

If the selected assets contain references to other assets, then include the referenced assets also in the export set. This selection process is repeated recursively until all asset references are satisfied.

6. If the selected assets contain references to other assets, select the **Include assets** checkbox to include the referenced assets also in the export set.

This selection process is repeated recursively until all asset references are satisfied.

7. Click **Apply Settings**.

8. The **Export** dialog box displays the list of selected assets and its dependent assets.

The checkbox beside each asset indicates whether or not the asset should be included in the export set. By default, all displayed assets are included in the export set.

If you wish to remove an asset from the export set, clear its checkbox. This removes the asset and all of its dependent assets (if any) from the export set.

9. Click **Export** to start the export operation.

Important:

If none of the asset is selected for export, the **Export** button is disabled.

Note:

If at any time you wish to abandon all your changes and return to your previous screen, just click the **Cancel** button.

An **Export Progress** popup will display the export progress bar.

10. Click **Download** if you wish to download the export archive file.

This starts the creation of the archive file.

The **Download** button remains disabled until the export operation is completed.

The default location to which the archive file is downloaded is `My Documents\Downloads`.

Importing Objects through CentraSite Business UI

To export an object, you must have View permission for the object you want to export.

CentraSite Business UI, you can do the import operation only for assets in CentraSite Business UI. You can import assets by importing the archive (zip) file to which the assets were previously exported. You can import assets into the same CentraSite registry from which they were originally exported or to a different CentraSite registry.

Predefined asset types that are exported from older versions of CentraSite may not get imported. The asset instances are imported only if they conform to the asset type schema in the target CentraSite registry.

> To import an asset or a set of assets

1. In CentraSite Business UI, click the **Create Asset** activity.

This opens the **Create New Asset** wizard. The bottom panel of the **Create New Asset** wizard shows the option to import an archive file.

2. To import an archive file, click **Choose** to navigate to the folder where the exported archive file resides, and choose the file.

When you select a file to import, the fields in the **Basic Information** panel cannot be edited.

3. Click **Next**.

4. The **Create New Asset** wizard displays the list of objects to import.

The checkbox beside each object indicates whether the object should be imported. By default, all objects displayed are included in the import set.

To exclude any object from the import set, clear its checkbox.

If you are importing an archive file that was generated prior to CentraSite version 9.0, the wizard does not display the list of objects. However, the objects are imported.

5. Expand **Advanced Settings** to display a set of additional import options. These settings are optional.

Option	Description
Change Owner	The imported assets can be assigned to the same owner as in the source CentraSite registry, or you can assign a new owner.

Option	Description
Change Organization	<p>The Change Owner field is type-ahead field. As you enter characters in this field, the dialog will list the usernames that match the characters you enter.</p> <p>When you import assets, you can import them into the same organization in the target CentraSite registry as in the source CentraSite registry from which they were exported, or you can assign a new organization.</p> <p>The Change Organization field is type-ahead field. As you enter characters in this field, the dialog will list the organization names that match the characters you enter.</p>
Retain lifecycle state	<p>This option determines whether the lifecycle state of the imported assets is preserved. Enable the option to retain the lifecycle state of the assets being imported.</p>
Overwrite existing entities	<p>This option specifies that existing assets with the same UUID in the target CentraSite registry will be overwritten, even if the asset in the archive is older than the one in the target CentraSite registry. Enable the option to overwrite the existing assets.</p>
Import groups that the user belongs to	<p>This option determines that when you import a user, whether you want to import the groups that the user belongs to. This applies only to user-defined groups. System-defined groups are not imported. Enable the option to import the groups.</p>
Ignore API keys and OAuth2 tokens	<p>This option determines whether the API keys and OAuth2 tokens of the imported assets are to be imported into the target CentraSite registry. Enable the option to ignore the API keys and OAuth2 tokens during the import process.</p>

6. Click **Import** to import the assets.
7. If an asset has an attribute that is required in the target CentraSite registry but not in the source CentraSite registry, CentraSite displays intermediate screens to provide values for each required attribute, before importing the asset.

This happens when an asset type definition in the source CentraSite registry is different from an asset type definition in the target CentraSite registry. For example, the asset type in the target CentraSite registry represents an updated version of the asset type with different attribute definitions.

8. When the import operation completes, the import wizard informs you if the import was successful or if there were any errors/warnings. Click **Download Import Log** to view the

import logs. When you click this link, the import log lists the status of all the objects stating whether they were successfully imported or if there were errors/warnings.

9. Click **OK** to terminate the import wizard.

Exporting and Importing Registry Objects through CentraSite Control

This section describes how you can perform exporting and importing objects through CentraSite Control.

Exporting Objects through CentraSite Control

You must have *view* permission for the assets you want to include in the export set.

With CentraSite Control, you can export one or more objects in each export operation. The export operation creates an archive file on the file system. The archive file consists a copy of the objects that you have exported. The archive file can be then imported into the same CentraSite registry or into a new registry.

Some objects in the registry do not support all the export methods. Check the user interface to see which controls are available for the type of object you want to export.

➤ To export an object or a set of objects

1. In CentraSite Control, go to the page that contains the object or the set of objects that you want to export.

For example, if you want to export a taxonomy, go to the **Administration > Taxonomies** page or if you want to export the contents of a list, go to **Home > My CentraSite > My Favorites** to see your defined lists.

2. Select the object or the set of objects you want to export and **Export** from the context menu.

The export dialog displays the selected objects and all dependent objects. The colored icon beside each object indicates whether the object is one of the selected objects or a dependent object.

Note:

Alternatively,

- If the **Actions** menu is visible, select the object or set of objects you want to export and select **Actions > Export**.
- If the **Export** icon is visible, select the object or set of objects you want to export and click **Export**.

3. Clear the checkbox beside each object if you do not want to include the object in the export set.

By default, all displayed objects are included in the export set.

If you wish to remove an object from the export clear the check box. This removes the object and all of its dependent objects (if any) from the export set.

4. For some object types, additional export options are available. In this case, click **Export Options** that is activated.

A dialog opens that displays these additional export options. If you are exporting objects that are contained in a list and the list contains more than one object type, you see several tabs, with one tab per object type.

The available options depend on the type of object you wish to export:

Object type	Available export options
Asset	<ul style="list-style-type: none">■ Include assets referenced by selected assets. If the selected assets contain references to other assets, then include the referenced assets in the export set. This selection process is repeated recursively until all asset references are satisfied.
Lifecycle Model	<ul style="list-style-type: none">■ Include assets for selected lifecycle models. Include all assets that are governed by the lifecycle model.■ Include assets referenced by selected assets. If the assets that are selected by Include assets for selected lifecycle models option consists of references to other assets, then include the referenced assets in the export set. This selection process is repeated recursively until all asset references are satisfied.
Taxonomy	<ul style="list-style-type: none">■ Include assets for selected categories. Include all assets that are classified by categories of the taxonomy.■ Include assets referenced by selected assets. If the assets that are selected by Include assets for selected categories option consists of references to other assets, then include the referenced assets in the export set. This selection process is repeated recursively until all asset references are satisfied.
Asset Type	<ul style="list-style-type: none">■ Include assets for selected asset types. Include all assets that are instances of the asset types.■ Include assets referenced by selected assets. If the assets that are selected by Include assets for selected asset types option consists of references to other assets, then include the referenced assets also in the export set. This selection process is repeated recursively until all asset references are satisfied.
Organization	<ul style="list-style-type: none">■ Include users. Include all users who belong to the organization.■ Include child organizations. Include all organizations that are child organizations of the organization. If the <code>Include assets of</code>

Object type	Available export options
	<p>organization option is selected, include all assets that belong to the child organizations.</p> <ul style="list-style-type: none"> <li data-bbox="535 336 1490 409">■ Include lifecycle models and policies. Include all lifecycle models and policies that have been define for the organization. <li data-bbox="535 430 1490 504">■ Include assets of organization. Include all assets that belong to the organization. <li data-bbox="535 525 1490 709">■ Include assets referenced by selected assets. If the assets that are selected by Include assets of organization option consists of n references to other assets, then include the referenced assets in the export set. This selection process is repeated recursively until all asset references are satisfied.

5. Click **OK**.
6. Click **OK** to start the export.
7. Specify a name for the archive file when prompted to do so.
8. Verify the export logs and ensure that the objects were exported successfully.

Click on the **Export Log** link in the confirmation message that appears when the export is complete.

Important:

If the report indicates that certain associated objects have been omitted from the archive, you have to ensure that these objects are either present in the target registry when the archive is imported or create an archive that includes them.

Importing Objects through CentraSite Control

You can import objects from an archive file that was previously created using the CentraSite export feature. The archive you wish to import must reside in the file system of the computer where your browser is running.

» To import objects from an archive file

1. In CentraSite Control, go to any page that displays the **Import** button.

Examples of pages that include this button are:

Asset Catalog > Browse

Policies > Design/Change-Time

Administration > Taxonomies

2. Click **Import**.
 - a. Select **Archive** from the **Import as** drop-down list.
 - b. In the **File** field, type the name of the file that contains the archive.
 - c. Click **Finish**. This displays the **Import Preview** page.

Note:

If the archive you wish to import was created using CentraSite 8.2 or earlier, the **Import Preview** page is not available. In this case, when you click **Finish**, the import operation continues with the **Import Options** dialog in step 4.

The **Import Preview** page displays the names of the top-level objects available in the export archive. If the archive contains related objects that the top-level objects require for completeness, the related objects are not displayed but they are imported automatically along with the top-level objects. For example, an exported web service requires a related schema, so the archive file contains both the web service and the schema, and an import of the web service causes the related schema to be imported also. By default, all displayed top-level objects are selected for import. This is indicated by the marked checkbox beside the name of each object.

Note:

In some cases, there may be a dependency between top-level objects (for example, a web service that refers to a taxonomy), and the import operation ensures that such dependencies are retained. This means that if you clear the checkbox of a top-level object that is required by another top-level object, CentraSite ensures that the required object is nevertheless included in the import.

3. Click **Import Options** to access these options.
4. In the **Import Options** dialog box, set the following options:

Option	Meaning
Keep current organization / Assign new organization	<p>Select Keep current organization to import the objects into the same organization. The organization in the target registry must have the same name and UUID as in the source registry.</p> <p>Select Assign new organization to import the objects into a new organization. If you select this option, you can select the new organization using the Select Organization option .</p> <p>In some cases, the original organization is preserved during import even if you have selected a specific organization in this field. This happens if the object to be imported is:</p> <ul style="list-style-type: none"> ■ an organization ■ a system-wide lifecycle model

Option	Meaning
	<ul style="list-style-type: none"> ■ a system-wide policy <p>Asset types, Association types, and Taxonomies are not owned by any organization, so selecting an organization for such objects has no effect.</p>
Keep current owner / Assign new owner	<p>Select Keep current owner to assign the imported objects to the same owner as in the source registry. The owner in the target registry must have the same UUID as in the source registry.</p> <p>Select Assign new owner to assign the imported objects to a new owner. If you select this option, you can select the new owner using the Select Owner option.</p>
Keep lifecycle state	<p>This option determines whether the lifecycle state of the imported assets is preserved. Enable the option to retain the lifecycle state of the assets being imported.</p>
Allow replace of registry objects	<p>Select this option to specify that existing objects with the same UUID in the target registry is overwritten, even if the object in the archive is older than the one in the target registry.</p>
Import groups that the user belongs to	<p>When you import a user, you can specify whether you want to also import the groups that the user belongs to. This applies only to user-defined groups. The system-defined groups are not imported.</p>

5. Click **OK**.

If you are importing an archive that was created with CentraSite 8.2 or earlier, the import begins.

If you are importing an archive that was created after CentraSite 8.2, the **Import Preview** dialog is displayed.

6. Click **OK** to start the import.

When the import operation completes, the import wizard informs you if the import was successful or if there were any errors.

You can click **OK** here to terminate the import wizard without viewing the import log.

Alternatively, to see details of the objects that were imported, the wizard offers you a link to view the import log. When you click this link, the import log lists each object and indicates whether or not it was successfully imported. The import log also lists the import status of any related objects that were contained in the archive.

7. In the import log view page, click **OK** to terminate the import wizard.

Handling Missing Attributes of Assets When Importing

It is possible that an asset type definition in the source registry is different from an asset type definition with the same name in the target registry. For example, this may happen when the asset type in the target registry represents an updated version of the asset type with different attribute definitions. If an asset type has an attribute that is required in the target registry but not in the source registry, then a mismatch occurs.

CentraSite handles this situation on the target registry depending on the attribute type, that is:

- If the attribute type is one that has a default value, such as `slot` or `classification`, CentraSite assigns the default value to the attribute when the asset is imported.
- If the attribute type is one that does not have a default value, for example, an attribute type that represents a file name or relationship, an error occurs during the import of the archive because the target registry requires a value for the attribute. Hence, the affected asset is not imported and the import log contains appropriate error messages. In such cases, you need to ensure that a value is supplied for the attribute before the asset is exported from the source registry.

Exporting and Importing Registry Objects through Command Line

This section describes how you can perform exporting and importing objects through Command Line Interface.

Exporting Objects Through the Command Line

Pre-requisites:

To export objects through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

You can export registry objects into an archive in the CentraSite registry.

CentraSite provides a command tool named `ExportArchive` for this purpose.

- Run the command tool `ExportArchive`.

```
The syntax is of the format: C:\SoftwareAG\CentraSite\utilities>ExportArchive.cmd
<CentraSite-URL> <archive-filename> <username> <password> [<guid>...] [-help]
[-noinstances] [-deleteafterexport] [-orgnusers] [-orgwithpolicyandlifecycle]
[-orgnochildorganization] [-orgwithassets] [-withreferencedassets] [-ignore id]
[-ignoretypes] [-savedsearch searchname] [-savedsearchuser user] [-searchfile filename]
[-modifiedsince timespec] [-target name] [-displaymainobjects]
[-displaycollectorresult] [-full]
```

The input parameters and options are:

Parameter	Description
CentraSite-URL	(Optional). The URL of the CentraSite registry/repository. For example, <code>http://localhost:53307/CentraSite/CentraSite</code>
archive-filename	The name of the export archive (Zip) file. The archive file can contain an organization with its assets or can contain a set of objects that were exported from one or more organizations.
username	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
password	The password for the CentraSite user identified by the parameter <code>username</code> .
guid	The GUID of the object to be exported, prefixed by <code>uddi</code> . For example, <code>uddi:207ff1cc-25c5-544c-415c-5d98ea91060c</code> . The <code>guid</code> parameter can be specified more than once.
-deleteafterexport	(Optional). All exported objects, except the asset types, taxonomies, and organizations, are deleted after export.
-displaymainobjects	(Optional). Display the main objects to be exported but do NOT export.
-full	(Optional). Display all the objects to be exported with the <code>-display</code> option
-help	(Optional). Display the full description of <code>ExportArchive</code> command (with detailed parameters and options description).
-ignore id	(Optional). Ignore the specified object at export. The <code>-ignore id</code> option can be specified more than once.
-ignoretypes	(Optional). Ignore all implicit asset types of the assets to be exported.
-minimizeaudits	(Optional). Export only the minimal set of audit trail.
-modifiedsince timespec	(Optional). Export only the objects that were created and modified after the specified timestamp. Syntax: <code>xs:dateTime</code> or <code>xs:duration</code> .
-noinstances	(Optional). Do not export any instance of the exported asset type (represented by a <code>Concept</code> object) or the taxonomy (represented by a <code>Classification Scheme</code> object). For other objects, the <code>-noinstances</code> option is ignored.

Parameter	Description
-orgnochildorganization	(Optional). Do not export the child organizations of the exported organization.
-orgnusers	(Optional). Do not export the users of the exported organization.
-orgwithassets	(Optional). Export all the assets of the exported organization.
-orgwithpolicyandlifecycle	(Optional). Export the related policies and lifecycles of the exported organization.
-savedsearch searchname	(Optional). Add the result of the saved search to the objects to export. The -savedsearch searchname option can be specified more than once. A search can be assembled and saved with CentraSite.
-savedsearchuser user	(Optional). Use the specified user for the -savedsearch option instead of the session user.
-searchfile file	(Optional). Add the result of the search, specified in the xml file, to the objects to export. This option may be specified more than once.
-targetname	(Optional). Export a specified target (for example, Mediator).
-withreferencedassets	(Optional). Export the directly referenced (associated) assets of the exported asset.

Example (all in one line):

The command for exporting the asset given by its GUID from CentraSite running on *localhost*, (ignoring export of the associated asset type), with an administrator account *INTERNAL\Admin* and password *AdminPW* would be:

```
C:\SoftwareAG\CentraSite\utilities>ExportArchive.cmd -ignoretypes  
http://localhost:53307/CentraSite/CentraSite c:/tmp/export.zip INTERNAL\Admin AdminPW  
uddi:cf5945c5-1867-11e7-9501-8baf70aded74
```

Importing Objects from an Archive Through the Command Line

Pre-requisites:

To import objects through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

You can import registry objects from an archive into the CentraSite registry.

CentraSite provides a command tool named `ImportArchive` for this purpose.

- Run the command tool `ImportArchive`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>ImportArchive.cmd <CentraSite URL> <archive filename> <username> <password> [-help] [-setreplace] [-keepowner] [-setowner name] [-keeporganization] [-keeplcmstate] [-removemissingreferences] [-importgroup] [-executewsdlpolicy] [-importorg id] [-importorgname name] [-ignoreauthtokens] [-simulate] [-importkeys id[,id...]] [-sequential [-minimizeaudits] [-listonly] [-skip cnt]]`

The input parameters and options are:

Parameter	Description
CentraSite-URL	(Optional). The URL of the CentraSite registry/repository. For example, <code>http://localhost:53307/CentraSite/CentraSite</code>
archive-filename	The name of the exported archive (Zip) file. The archive file can contain an organization with its assets or can contain a set of objects that were exported from one or more organizations.
username	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
password	The password for the CentraSite user identified by the parameter <code>username</code> .
-help	(Optional). Display the full description of <code>ImportArchive</code> command (with detailed parameters and options description).
-executewsdlpolicy	(Optional). Execute the WSDL Regeneration policy for service import. By default, the command execution ignores the policy.
-ignoretypeversion	(Optional). Ignore the version check on asset types.
-importgroup	(Optional). Import the groups that include a single user.
-importkeys id[,id...]	(Optional). Import only those objects which have keys specified.
-importorg id	(Optional). Import the objects into the organization specified by the UUID.
-importorgname name	(Optional). Import the objects into the organization specified by the name.
-keeplcmstate	(Optional). Keep the LCM state of the object which is set at export.
-keepowner	(Optional). Keep the object owner instead of assigning the importing user.

Parameter	Description
-keeporganization	(Optional). Keep the organization of the imported objects instead of assigning the active one performing the message logging.
-listonly	(Optional). Print only the list of objects to be imported. The <code>-listonly</code> option can be used to adjust the <code>-skip</code> option. No updates will be performed with the <code>-sequential</code> option.
-minimizeaudits	(Optional). Import only first and last audit record for each object, along with the <code>-sequential</code> option.
-removemissingreferences	(Optional). Remove all missing associations that could cause dangling references.
-setowner name	(Optional). All the imported objects will be set to a specified user.
-setreplace	(Optional). Replace objects if already present in the target registry.
-simulate	(Optional). Simulate the import, however, do not update the objects in the target registry.
-sequential	(Optional). Imports the objects sequentially in a reasonable order.
-skip cnt	(Optional). Skip records before importing, along with the <code>-sequential</code> option.
-ignoreauthtokens	(Optional). Ignore API keys and OAuth2 tokens of the imported assets.

Example (all in one line):

The command for importing an archive into CentraSite running on host *myhost*, and force a replacement of all assets from archive in the CentraSite registry, with an administrator account *INTERNAL\Admin* and password *AdminPW* would be:

```
C:\SoftwareAG\CentraSite\utilities>ImportArchive.cmd -setreplace  
http://myhost:53307/CentraSite/CentraSite c:/tmp/export.zip INTERNAL\Admin AdminPW
```

Best Practices: Exporting and Importing Assets

When you export an instance of an asset from a lower version of CentraSite (say, source registry), CentraSite exports the asset and a number of referenced and associated objects to an archive file on the file system. The export set also includes the asset's type definition, if the type is a custom type or a modified predefined type.

When you import the same instance of asset to a higher version of CentraSite (say, target registry), CentraSite creates the imported asset, its referenced and associated objects, and the asset's type definition in the target registry.

If the imported asset's type definition is a custom type, the import operation between versions of CentraSite is successful and the appropriate entries are created in the target registry.

However, if the imported asset's type definition is a predefined asset type, the import operation between different versions of CentraSite results in various compatibility problems. Predefined asset types, for example, Service type definition which are updated during fix releases in a lower version of CentraSite might not be compatible with the same Service type definition available in a higher version of CentraSite. During import of such predefined asset types, CentraSite exhibits one of the following compatibility issues:

- **Scenario A:** If the asset's type definition includes minor changes on each of the two CentraSite versions, the import operation is partially successful. CentraSite filters the predefined asset type and writes the following message to the Import log:

```
INMIEE0072: Cannot update asset type because of version mismatch ("...")
```

- **Scenario B:** If the asset's type definition includes major changes on each of the two CentraSite versions, the import operation fails. For example, if the source registry contains an attribute that is not available in the target registry, CentraSite fails the import operation and writes the following error message to the Import log:

```
INMIEE0055: Asset to be imported does not fit to Asset type definition  
with attribute mismatch
```

To work around the scenario B, export the asset with the help of a temporary database. Perform an upgrade using the temporary database by following these steps:

1. Create a temporary database for the source registry. That is, do a backup of the lower version of CentraSite.
2. Upgrade the temporary database to the required target registry which is the higher version of CentraSite, by executing the CentraSite command tool `MoveCentraSiteRR.[cmd|sh]`.
3. Export asset from the upgraded temporary database to an archive file.
4. Import the archive file into the database of the target registry with the `replace` option.

Now, the predefined asset types in archive file and the database of the target registry are compatible and can be modified to suit the requirements.

15 Suite Usage Aspects

■ Introduction to Suite Usage Aspects	1544
■ Versioning Assets	1544
■ Modifying or Deleting Assets	1544
■ Publishing Assets	1544
■ CentraSite Communication with Designer UI	1545
■ User Accounts	1545
■ UDDI Clients	1545
■ Using CentraSite with ARIS	1546
■ Using CentraSite with webMethods API Portal	1558

Introduction to Suite Usage Aspects

Products of the webMethods suite use the CentraSite Registry Repository for storing and maintaining their objects and object types. These objects and object types should normally only be maintained by the suite products themselves, rather than by a CentraSite user who has access to such objects and object types. Thus, for example, a user of CentraSite who has access to the objects and object types, such as a user with the CentraSite Administrator role, must avoid making any change that would cause the objects or objects types to become unusable by the suite products.

Versioning Assets

CentraSite offers a versioning capability, which allows you to maintain several versions of an asset.

When you generate a new version of an asset, CentraSite adds a new asset of the same type to the catalog. The new asset has the same name and description as the one from which it was versioned and has an updated version number. The new version is related to the old version by a Super sedes association from the new version to the old version. In cases where the detail page of an asset has a Summary profile, the association is displayed under the Summary profile.

The versioning capability for the asset types such as IS Package and IS Service defined by the suite products is by default not activated. Unless the documentation for the suite components states otherwise, do not activate the versioning for these asset types.

Modifying or Deleting Assets

CentraSite users with the appropriate permissions can delete existing assets.

Generally, asset instances that are created and used by suite products are governed by policies that prevent their modification or deletion. The predefined design/change-time policy **Prevent Editing of webMethods Assets** is an example of such a policy. This means that a CentraSite user normally can not modify or delete the asset, due to the rules defined in the policy.

However, a user with the Modify Assets permission, or with a permission that implies the Modify Assets permission, can modify assets regardless of the policy settings. Also, if the suite product does not define a deletion policy for its assets, this gives CentraSite users with the appropriate permissions the theoretical possibility of modifying or deleting the assets. If you modify or delete such an asset in CentraSite, this may lead to inconsistencies or errors in the product that created the asset, if that product requires the asset definition to be still present and unchanged.

Deletion of an asset that belongs to a suite asset type should normally only be done using the Retract feature of the suite's Designer user interface.

Publishing Assets

When a suite product tries to update an asset that it previously created in CentraSite, the update fails if the asset is locked due to a pending activity at the CentraSite level. This can happen, for example, if a CentraSite policy related to the asset has been triggered that requires an approval action and the approval has not yet been granted.

Example: A user publishes a package from the Designer UI and this publish creates a service among other things in CentraSite. Then a CentraSite user with appropriate permissions tries to change the lifecycle state of this service and this triggers a policy you have defined for the service. If this policy has an approval action, then as long as the approval is pending no-one can edit this asset. This is according to design and is the same for all asset types (both suite and otherwise). If the user now tries to publish the package again, the publishing logic will try to edit the service and fails since CentraSite does not allow edits. The policy can be triggered by a CentraSite user as well. Basically if you have a policy for the specific asset type with approval action, you may run into this situation.

CentraSite Communication with Designer UI

Integration Server assets can be published at the package level to CentraSite from the Package Navigator view of the Designer. For publishing assets to CentraSite, the publisher uses CentraSite connection information provided in the Integration Server administration console.

The CentraSite connection information must specify a valid CentraSite URL and a user name which the publisher uses for authentication purposes to communicate with CentraSite.

If you change the CentraSite connection information in any way, for example, if you delete the user from CentraSite, it does not be possible to publish any assets from the Integration Server to CentraSite.

Remote Connection from Designer

Connecting remotely to CentraSite from Designer may result in time-outs depending on the individual environment and due to subsequent changes of the IP address (TCP/IP stack, VPN, and so on).

In such cases it may be necessary to restart/reconnect the following components:

- Software AG Runtime
- CentraSite Registry Repository
- Any client involved

User Accounts

Suite products generally communicate with CentraSite using existing CentraSite user accounts. Therefore, CentraSite administrators must be aware that any modification of the scope of these user accounts (for example, changing the password or permissions of the user) could affect the functionality of the suite product.

Before you modify the definition of a user account, check if any connected suite products use the user account, and ensure that the modification does not limit the functionality of the suite products.

UDDI Clients

If you are using a UDDI client to access CentraSite, note the following points:

- The UDDI term *Business* is the equivalent of the CentraSite term *Organization*. If you store a UDDI *Business* object in CentraSite, it is stored internally in CentraSite as an *Organization* object and is visible in CentraSite as an *Organization* object. When you retrieve the object through UDDI, it will of course be returned as a *Business* object.
- If you are using UDDI V2 to communicate with CentraSite, note that the predefined CentraSite UDDIv2 Inquiry Policy is by default not active. This policy is required for UDDI V2 processing.

To activate the policy, open CentraSite Control, display the list of design/change-time policies, select the UDDIv2 Inquiry Policy and set its lifecycle state to *Productive*.

Using CentraSite with ARIS

This section describes how ARIS uses the CentraSite Registry Repository for storing and maintaining their objects and object types.

CentraSite Profile for ARIS Properties

The asset type *Process* contains a profile named **ARIS Properties**, which includes attributes that are of use when CentraSite is integrated with the ARIS products.

The **ARIS Properties** profile includes the following attributes:

Attribute	Description
ARIS GUID	The Global Unique Identifier (GUID) that is assigned to the asset and uniquely identifies it once imported into ARIS.
ARIS Client	Download URL for ARIS Web client.
ARIS Publisher URL	URL for assets published through ARIS Publisher.
ARIS Connect URL	URL to establish connection to the ARIS Connect.

The asset type *Process* contains the ARIS Properties profile by default. An administrator can optionally select to disable this profile in the asset type definition. For example, if ARIS is not jointly used with CentraSite, an administrator might want to disable this profile for all assets on type *Process*.

Configuring CentraSite for Use with ARIS

ARIS uses CentraSite to retrieve and publish services and processes for process execution. When users define business processes with ARIS Architect, they can optionally publish the resulting processes and related services to CentraSite.

CentraSite includes predefined lifecycle models and predefined design/change-time policies to facilitate the governance of the business processes that ARIS publishes.

Note:

When users working on the same ARIS database are created as users in different CentraSite organizations, by default, they do not have visibility to each others' assets. Software AG recommends to either have all users of an ARIS database in the same CentraSite organization, or if this is not possible, to create a group in CentraSite containing all those users, and assign View or Modify permissions to the group.

Lifecycle Models That ARIS Uses

When ARIS publishes a business process to CentraSite, the business process is represented in the registry by a Process object. ARIS requires business processes (Process objects) to be governed by a lifecycle model. CentraSite includes a predefined lifecycle model called *BPM Process Lifecycle* for this purpose.

BPM Process Lifecycle Model

The *BPM Process Lifecycle* model defines the following states and transitions:

State	Has valid transitions to...
Requested	Implementing, Available, Retired, Rejected
Implementing	Implemented
Implemented	Testing
Testing	Tested
Tested	Available
Available	Retired
Retired	Requested
Rejected	<i>This is an end state. It has no outgoing transitions.</i>

Certain operations performed in ARIS trigger policies in CentraSite and these policies change the lifecycle state of a given Process object. When a user deletes a business process in ARIS, a policy in CentraSite automatically switches that process to the Retired state in the registry.

When you install CentraSite, this predefined lifecycle model is disabled (that is, inactive). To use CentraSite with ARIS, you must activate the lifecycle model. For more information, see *Activating the Lifecycle Models and Design/Change-Time Policies That ARIS Uses*, later in this topic.

Service Lifecycle Model

The *Service Lifecycle* model defines the following states and transitions:

State	Has valid transitions to...
Proposed	In Design, Rejected, Retired

State	Has valid transitions to...
In Design	In Development, Rejected
In Development	Implemented
Implemented	In Test
In Test	Tested, In Development
Tested	In Production
In Production	Under Maintenance, Deprecated
Under Maintenance	In Production
Deprecated	In Production, Retired
Retired	<i>This is an end state. It has no outgoing transitions.</i>
Rejected	<i>This is an end state. It has no outgoing transitions.</i>

Certain operations performed in ARIS trigger policies in CentraSite, and these policies change the lifecycle state of a given Service object. When a user deletes a service in ARIS, a policy in CentraSite automatically switches that process to the Retired state in the registry.

When you install CentraSite, this predefined lifecycle model is disabled (i.e., inactive). To use CentraSite with ARIS, you must activate the lifecycle model. For more information, see *Activating the Lifecycle Models and Design/Change-Time Policies That ARIS Uses*, later in this topic.

Customizing the Predefined Lifecycle Model for Process Objects

If the predefined lifecycle models that CentraSite provides does not suit your needs, you can customize them or create a new model for Process objects or Service objects. If you do this, however, you must also update the predefined policies that switch the lifecycle state of objects in the registry. For more information about which policies change the state of an object, see *Design/Change-Time Policies That ARIS Uses*, later in this topic.

Note:

The CentraSite Community Edition does not support the use of customized lifecycle models for Process objects or Service objects. If you are using the Community Edition, you must use the predefined lifecycle model that CentraSite provides.

Design/Change-Time Policies That ARIS Uses

CentraSite includes the following predefined policies.

Policy Name	Purpose
ARIS Delete Processes	Switches a specified Process object to the Retired lifecycle state. This policy executes when a process model is deleted by a Delete Process call from ARIS.

Policy Name	Purpose
-------------	---------

Note that if the Process object in CentraSite is not in a state that permits a transition to the Retired state (that is, if a transition to the Retired state is not permitted by the Process object's lifecycle model) the policy fails.

Important:

If you customize the predefined lifecycle model that CentraSite installs for Process objects, or if you apply a different lifecycle model to Process objects, you must modify this policy so that it switches the Process object to the appropriate state in your customized lifecycle model.

ARIS Delete Services	<p>Switches a specified Service (native or virtual) to the Retired lifecycle state. This policy executes when a service is deleted by a Delete Service call from ARIS.</p> <p>Note that if the Service in CentraSite is not in a state that permits a transition to the Retired state (that is, if a transition to the Retired state is not permitted by the Service's lifecycle model) the policy fails.</p> <p>Important: If you customize the predefined lifecycle model that CentraSite installs for Services, or if you apply a different lifecycle model to Services, you must modify this policy so that it switches the Service to the appropriate state in your customized lifecycle model.</p>
ARIS Release Processes	<p>Switches a specified Process object to the Available lifecycle state. This policy executes when a process model is released by a Release Process call from ARIS.</p> <p>Note that if the Process object in CentraSite is not in a state that permits a transition to the Available state (that is, the transition to the Available state is not permitted by the Process object's lifecycle model), the policy fails.</p> <p>Important: If you customize the predefined lifecycle model that CentraSite installs for Process objects, or if you apply a different lifecycle model to Process objects, you must modify this policy so that it switches the Process object to the appropriate state in your customized lifecycle model.</p>
ARIS webMethods Integration	<p>Sets instance-level permission and profile-level permissions to specified users or group for a Process object in CentraSite.</p> <ul style="list-style-type: none"> ■ This policy contains the action Set Instance and Profile Permissions, which assigns the instance-level permission and optionally profile-level permissions to a specific set of users or groups (called the ARIS webMethods Integration Group) for the Process object and related assets. The ARIS webMethods Integration Group includes the group of users who can be availed in ARIS or Designer for ARIS webMethods Integration. ■ You can apply this policy when the following events occur to a Process object: <ul style="list-style-type: none"> ■ PostCreate

Important:**Important:**

Policy Name Purpose

- PreUpdate
- OnTrigger

Enforce Unique Name Ensures that the names of Application Server objects, IS Server objects and IS Connection objects that are created in CentraSite are unique.

- This policy contains the action Enforce Unique Name, which you can configure to:
 - Enforce the unique name requirement for Application Server objects, IS Server objects, and IS Connection objects in all organizations defined in CentraSite.
 - Allow different versions of an object to exist in CentraSite with same the name.
- You can apply this policy when the following events occur to an object:
 - PreCreate
 - PreUpdate

Notify ARIS on Service Changes Notifies the ARIS APG service endpoint when a Service object (native or virtual) in CentraSite is updated.

- This policy contains the action Notify ARIS Service, which notifies the ARIS APG Service endpoint with the SOAP request message provided in this action. The APG Service endpoint is picked up from the associated ARIS Application Server.
- You can apply this policy when the following events occur to a Service object:
 - PostUpdate
 - PostStateChange
 - OnTrigger

Notify ARIS on Service Completion Notifies the ARIS APG service endpoint when a Service object (native or virtual) in CentraSite when a user changes the state of the service to a “completed” lifecycle state (for example, the Productive state).

- This policy contains the action Notify ARIS Service, which notifies the ARIS APG Service endpoint with the SOAP request message provided in this action. The APG Service endpoint is picked up from the associated ARIS Application Server.
- You can apply this policy when the following events occur to a Service object:
 - PostStateChange

Policy Name	Purpose
	<ul style="list-style-type: none"> ■ OnTrigger
Notify ARIS on Service Deletion	<p>Notifies the ARIS APG service endpoint when a Service object (native or virtual) in CentraSite is deleted.</p> <ul style="list-style-type: none"> ■ This policy contains the action Notify ARIS Service, which notifies the ARIS APG Service endpoint with the SOAP request message provided in this action. The APG Service endpoint is picked up from the associated ARIS Application Server. ■ You can apply this policy when the PostDelete event occurs to a Service object.
Reset Lifecycle State to Initial State	<p>Resets the lifecycle state of a Process object if it is republished from ARIS.</p> <ul style="list-style-type: none"> ■ This policy contains the action Set State, which you can set to any state of the BPM Process Lifecycle model. ■ You can apply this policy when the OnTrigger event occurs to a Process object.

Configuring and Activating Lifecycle Models and Design/Change-Time Policies That ARIS Uses

You use the following procedures to configure and activate the predefined lifecycle models (LCM) and design/change-time policies used by ARIS.

When you install CentraSite, the lifecycle models and design/change-time policies are in the inactive state.

Note:

The activation of the predefined lifecycle models (LCM) and design/change-time policies used by ARIS is not supported if you are using a CentraSite Community Edition license.

Configuring ARIS Change-Notification Policies

Use the following procedure to configure the Notify ARIS Service action in the ARIS change-notification policies (that is, the Notify ARIS on Service Changes, Notify ARIS on Service Completion and Notify ARIS on Service Deletion policies). The change-notification policies inform the ARIS server of changes that have been made to objects in the registry that belong to ARIS.

> To configure a change-notification policy

1. Go to **Policies > Design/Change Time** to display the policy list.
2. Enable the **Show Predefined Policies** option to display the predefined policies that CentraSite provides.

3. Click the policy that you want to configure.
4. Click the Notify ARIS Service action on the **Actions** tab to open the Edit Action Parameters page and then do the following:
 - a. Set the following parameters:

Parameter	Description
HTTP Basic Auth Enabled	<p><i>Boolean</i> Specifies whether the service is secured by Basic HTTP authentication.</p> <p>If you enable this option, you can optionally specify the user ID and password that CentraSite is to submit when it invokes the service in the following parameters. If you leave these parameters empty, CentraSite will submit the credentials belonging to the user who triggered this policy action.</p>
HTTP Basic Auth Username	The user ID that you want CentraSite to submit for HTTP basic authentication.
HTTP Basic Auth Password	The password associated with the user ID specified in HTTP Basic Auth Username.
SOAP Request Message	<p><i>String</i> The SOAP message that CentraSite is to submit to the ARIS service. The default message (shown below) includes substitution tokens that insert run-time data into the message. The substitution tokens are described in the documentation for the Send Email Notification action in the <i>CentraSite User's Guide</i>. You can use these substitution tokens to customize this message.</p>

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:web="http://www.idsscheer.com/age/webMethods/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:UpdateServiceRequest>
      <dbname>${context.ARIS_DB_CONTEXT}</dbname>
      <language>${user.locale}</language>
      <serviceDetail>
        <guid>${entity.key}</guid>
        <name>${entity.name}</name>
        <url>${entity.URL}</url>
        <lifeCycleState>${entity.state}</lifeCycleState>
        <owner>${entity.owner}</owner>
        <description>${entity.description}</description>
        <organization>${entity.organization}</organization>
        <version>${entity.version}</version>
        ${entity.attribute.Operations}
      </serviceDetail>
    </web:UpdateServiceRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Parameter	Description
	<pre></web:UpdateServiceRequest> </soapenv:Body> </soapenv:Envelope></pre>
SOAP Action	<i>String</i> The SOAP action that CentraSite sets in the message. If you do not set this parameter, CentraSite sets the SOAP action to the empty string.
Connection Timeout (in milliseconds)	<i>Number</i> The length of time in milliseconds that CentraSite will wait for a response from the remote machine. If the timeout limit is exceeded, the policy action fails. The default value is 60000 milliseconds.
Content Type	<i>String</i> The value that CentraSite is to assign to the Content-Type header in the SOAP request that it submits to the service. Default: text/xml.

- b. Click **Save** to save the parameter settings.
5. Click **Save** to save the updated policy.
6. Activate the policy.

Configuring Enforce Unique Name Policy

Use the following procedure to configure the Enforce Unique Name action in the Enforce Unique Name policy. This policy ensures that the names of the Application Server type objects that are created in CentraSite are unique.

➤ To configure the Enforce Unique Name policy

1. Go to **Policies > Design/Change Time** to display the policy list.
2. Enable the **Show Predefined Policies** option to display the predefined policies that CentraSite provides.
3. Click the Enforce Unique Name policy.
4. Click the Enforce Unique Name action on the **Actions** tab to open the Edit Action Parameters page and then do the following:
 - a. Set the following parameters:

Parameter	Description
Enforce Across Organizations	<i>Boolean</i> If this parameter is set to True, then the unique name requirement for Application Server type objects is enforced in all organizations defined in CentraSite.

Parameter	Description
Allow Different Versions	<i>Boolean</i> If this parameter is set to True, then different versions of an Application Server type object can exist in CentraSite with same the name.

The event scope of this action is:

- PreCreate
- PreUpdate

- b. Click **Save** to save the parameter settings.
5. Click **Save** to save the updated policy.
6. Activate the policy.

Configuring Reset Lifecycle State to Initial State Policy

Use the following procedure to configure the Set State action in the Reset Lifecycle State to Initial State policy. This policy resets the lifecycle state of a Process object if it is republished from ARIS.

➤ To configure the Reset Lifecycle State to Initial State policy

1. Go to **Policies > Design/Change Time** to display the policy list.
2. Enable the **Show Predefined Policies** option to display the predefined policies that CentraSite provides.
3. Click the Reset Lifecycle State to Initial State policy.
4. Click the Set State action on the **Actions** tab to open the Edit Action Parameters page, and then do the following.
 - a. Set the following parameter:

Parameter	Description
Change State To	<i>String</i> Set the parameter to any state of the “BPM Process Lifecycle” model.

The event scope of this action is:

- OnTrigger

- b. Click **Save** to save the parameter setting.
5. Click **Save** to save the updated policy.

6. Activate the policy.

Activating ARIS-Related LCMs and Policies through CentraSite Control

Use the following procedure to activate the lifecycle models and design/change-time policies for ARIS.

➤ To activate the ARIS-related LCM and policies

1. Open CentraSite Control.
2. Activate the ARIS lifecycle models using the following steps:
 - a. Go to **Administration > Lifecycles > Models**.
 - b. Locate the lifecycle model called BPM Process Lifecycle and activate it.
 - c. Locate the lifecycle model called Service Lifecycle and activate it.
3. Activate the state-change policies using the following steps:
 - a. Go to **Policies > Design/Change Time** to display the policy list.
 - b. Enable the **Show Predefined Policies** option to display the predefined policies that CentraSite provides.
 - c. Activate each of the following policies.
 - ARIS Release Processes
 - ARIS Delete Processes
 - ARIS Delete Services
 - ARIS webMethods Integration
 - Reset Lifecycle State to Initial State
4. Configure and activate each of the following policies.
 - Notify ARIS on Service Changes
 - Notify ARIS on Service Completion
 - Notify ARIS on Service Deletion
5. Configure and activate the Enforce Unique Name.
6. Configure and activate the Reset Lifecycle State to Initial State.

Activating Policies for ARIS through Command Line Interface

Pre-requisites:

To activate policies for ARIS through the CentraSite Command Line Interface, you must have the CentraSite Administrator role.

CentraSite provides a command tool named `EnableARISIntegration` for this purpose. The tool helps you to:

- Activate the lifecycle models and design/change-time policies that ARIS requires.
- Configure the Notify ARIS Service action in the ARIS change-notification policies (that is, the Notify ARIS on Service Changes, Notify ARIS on Service Completion, and Notify ARIS on Service Deletion policies). The change-notification policies inform the ARIS server of changes that have been made to objects in the registry that belong to ARIS.
- Configure the Enforce Unique Name action in the Enforce Unique Name policy. This policy ensures that the names of the Application Server type objects that are created in CentraSite are unique.
- Configure the Set State action in the Reset Lifecycle State to Initial State policy. This policy resets the lifecycle state of a Process object if it is republished from ARIS.

This is of a great usage for the Community edition of CentraSite as policies cannot be manipulated through its user interface.

> To activate policies

- Run the command `EnableARISIntegration`.

The syntax is of the format: `C:\SoftwareAG\CentraSite\utilities>EnableARISIntegration.cmd -user <USERNAME> -password <PASSWORD> [-dburl <CENTRASITE-URL> | [-h <HOSTNAME> -p <PORT>]]`

The input parameters are:

Parameter	Description
<i>CENTRASITE-URL</i>	The URL of the CentraSite registry. For example, <code>http://localhost:53307/CentraSite/CentraSite</code> .
	Note: If the registry or repository is running on a different machine and port number, you can use this parameter to specify its location instead of using the individual <code>-h</code> and <code>-p</code> parameters. (If you specify the <code>-dburl</code> parameter with the <code>-h</code> and <code>-p</code> parameters, the <code>-h</code> and <code>-p</code> parameters is ignored.)
<i>USERNAME</i>	The user ID of a registered CentraSite user who has the CentraSite Administrator role. For example, <code>Administrator</code> .
<i>PASSWORD</i>	The password for the registered CentraSite user identified by the parameter <code>-user</code> .

Parameter	Description
<i>HOSTNAME</i>	The host name or IP address of the computer where the CentraSite registry or repository component is running. If you omit this parameter, the importer assumes that the registry or repository is running on <code>localhost</code> .
<i>PORT</i>	The port number on which the CentraSite registry or repository is configured to listen for incoming requests. If you omit this parameter, the importer assumes that the registry or repository is listening on the default port, <code>53307</code> .

Example (all in one line):

```
C:\SoftwareAG\CentraSite\utilities>EnableARISIntegration.cmd -url
http://localhost:53307/CentraSite/CentraSite -user Administrator -password manage -h
localhost -p 53307
```

Provisioning CentraSite Services into ARIS Architect

CentraSite provides Web service operations for provisioning the Integration Server (IS) services and Web services into ARIS Architect.

The Web service operations filter Integration Server (IS) Services that are classified under the concept called `WMAssetType -> ReusableAsset`. An Integration Server service is classified under this concept, when the **Reuse** property value is set to `public` in Designer. This helps in filtering the default Integration Server services (example **wMPublic**), if they are published to CentraSite.

Following are the list of Web service operations that help in provisioning CentraSite services into ARIS:

- `findAllServices`
- `getAllAssociatedServices`
- `getAllServiceDetails`

These operations are part of `SearchService` (Web service).

Each of these operations have their object types differentiated using a property element that comes as part of the response payload as below.

Object Type Representation for Integration Server Service

```
<ns5:property><name>ServiceType</name><value>IS Service</value></ns5:property>
```

Object Type Representation for Web Service

```
<ns5:property><name>ServiceType</name><value>Service</value></ns5:property>
```

Using CentraSite with webMethods API Portal

This section describes how CentraSite uses API Portal to publish APIs to external developers and partners and provides design-time governance capabilities to the APIs. API Portal allows developers to self-register, learn about these APIs, and try out the APIs in their applications.

CentraSite Profiles for API Portal Properties

CentraSite includes the following predefined profiles for grouping the API Portal properties:

- API Portal Information
- API Portal Communication
- CentraSite Communication
- Published APIs

API Portal Information

The profile appears in the details page of a Service asset instance and its variants (REST Service, OData Service, Virtual Service, Virtual REST Service, and Virtual XML Service). This profile includes classification details of the asset, when CentraSite is integrated with webMethods API Portal.

The API Portal Information profile includes the following attributes:

Attribute	Description
API Proxy URLs	URL of the proxy server through which requests are routed to the actual endpoint of the native service or the Mediator endpoint of the virtual service.
API Grouping	<p>Groups the APIs by freely definable business terminology to indicate the API usage.</p> <p>The API Grouping taxonomy contains the following predefined categories in CentraSite:</p> <ul style="list-style-type: none">■ CRM■ Financing, Banking and Insurance■ Sales and Ordering■ Search■ Transportation and Warehousing <p>In addition, you can create your own custom categories.</p>
API Icon	An icon that would represent the API in API Portal.

Attribute	Description
API Maturity Status	<p>The maturity status of the API based on a customizable set of terms.</p> <p>The <code>API Maturity Status</code> taxonomy contains the following predefined categories in CentraSite:</p> <ul style="list-style-type: none"> ■ A <code>Beta</code> phase, the first stage of API maturity, when the features and functions of the API are currently undergoing beta testing. ■ An <code>Experimental</code> phase is the next stage of API maturity, when the usage of an API is limited and focused on gathering feedbacks. ■ A <code>Test</code> phase, when the features and functions of the API are undergoing testing in a controlled environment that mimics production scenarios. ■ A <code>Production</code> phase is the final stage of API maturity, when the features and functions of the API are available publically. <p>In addition, you can create your own custom categories.</p>
API Subscription Terms	<p>Specifies the category of the key assigned to the client to access the API based on subscription plans.</p> <p>The <code>API Subscription Terms</code> taxonomy contains the following predefined categories in CentraSite:</p> <ul style="list-style-type: none"> ■ <code>Donationware</code> - usage of the API is free of charge, but users are encouraged to make a donation if they like the API and want to continue using it. ■ <code>Flat Fee</code> - a fixed amount charged for unlimited use of the API for a limited period of time. ■ <code>Free</code> - usage of the API is free of charge. ■ <code>Freemium</code> - limited usage of the API for free, and then charge for the extra usage. ■ <code>Pay per use</code> - pay for usage of the API based on the transaction or volume. <p>In addition, you can create your own custom categories.</p>
List of Access Tokens	(Read-only). A list of the access tokens that were generated for the clients who requested for the API access through API Portal.
Supported Access Token Types	(Read-only). The type of client authentication mechanism for the API.

Attribute	Description
	<p data-bbox="479 252 1258 325">The possible client authentication mechanisms in CentraSite are:</p> <ul data-bbox="479 346 1289 892" style="list-style-type: none"><li data-bbox="479 346 1289 420">■ <code>API Key Authentication</code> - The API's authentication requires using an API key.<li data-bbox="479 441 1289 514">■ <code>Basic Authentication</code> - The API's authentication requires using Basic Access Authentication as described in RFC2617.<li data-bbox="479 535 1289 609">■ <code>Digest Authentication</code> - The API's authentication requires using Digest Access Authentication as described in RFC2617.<li data-bbox="479 630 1289 703">■ <code>OAuth 1.0 Authentication</code> - The API's authentication requires using OAuth 1.0 as described in RFC5849.<li data-bbox="479 724 1289 798">■ <code>OAuth 2.0 Authentication</code> - The API's authentication requires using OAuth 2.0 as described in RFC5849.<li data-bbox="479 819 1289 892">■ <code>x-{other}</code> - The API's authentication requires using another authentication method.
Deprecated	Marks the API as deprecated in API Portal.

Note:

By default, this profile shows disabled in the Service asset type definitions. However, an administrator can enable this profile in the type definitions as required. For example, if API Portal is jointly used with CentraSite, administrator can enable this profile for all of the Service type definitions.

API Portal Communication

The profile appears in the details page of an API Portal gateway asset instance. This profile includes the communication details for the CentraSite instance to send and receive data with the API Portal instance, when CentraSite is integrated with API Portal.

CentraSite Communication

The profile appears in the details page of an API Portal gateway asset instance. This profile includes the communication details for the API Portal instance to send and receive data with the CentraSite instance, when CentraSite is integrated with API Portal.

Published APIs

The profile appears in the details page of an API Portal gateway asset instance. This profile includes a list of the APIs that were published to the API Portal instance, when CentraSite is integrated with API Portal.

Configuring CentraSite for Use with API Portal

Before you start configuring CentraSite for API Portal, make sure the following products are installed in addition to CentraSite:

- webMethods API Portal
- webMethods Integration Server
- webMethods API Gateway

For more information about installing each of these products, see *Installing Software AG Products*.

Configuring and Activating Design/Change-Time Policies That API Portal Uses

You use the following procedures to configure and activate the predefined design/change-time policies used by API Portal.

When you install CentraSite, the design/change-time policies for API Portal communication are in the inactive state. You will not be able to configure a design/change-time policy while it is active. To configure an active policy, you must do one of the following:

- Switch the policy to the Suspended state (to deactivate it), update the policy and then switch it back to the Productive state (to reactivate it).
- Create a new version of the policy, make your changes to the new version of the policy and then switch the new version to the Productive state. Switching the new version of the policy to the Productive state will automatically Retire (and deactivate) the old version.

Note:

The activation of the predefined design/change-time policies used by API Portal is not supported if you are using a CentraSite Community Edition license.

Configuring Publish to API Portal Policy

Use the following procedure to configure the **Publish to API-Portal** action in the **Publish to API-Portal** policy. The policy (1) creates a new API and (2) updates an existing API metadata in the API Portal registry:

➤ To configure the publish to API Portal policy

1. In CentraSite Control, go to **Policies > Design/Change Time**.
2. Enable the **Show Predefined Policies** option.

A list of the predefined policies appears.

3. Click the Publish to API-Portal policy.
4. If the policy is active, deactivate it.
5. In the **Actions** tab, click the **Publish to API-Portal** action. In the Edit Action Parameters page, do the following.
 - a. Define the policy parameters:

API-Portal (Optional). (String). (Array). Name of the API Portal gateway to which the API would be published. This assumes that you have already registered the API Portal in CentraSite.

Note:

However, if this action is to be executed in a different event other than OnTrigger, for example, Pre-State Change or Post-State Change, that is not provided by default, you *must* specify a value for this field.

Endpoint Category (Optional). (String). (Array). A list of the specific taxonomy categories by which the base URLs (endpoints) of the API are classified.

REST Service Attributes (Optional). (String). (Array). The metadata bundle can be supplied with additional information of a REST API and published to an CentraSite gateway. You use this field to specify additional attributes of the REST API to be published to the CentraSite gateway.

SOAP Service Attributes (Optional). (String). (Array). The metadata bundle can be supplied with additional information of a SOAP-based API and published to an CentraSite gateway. You use this field to specify additional attributes of the SOAP API to be published to the CentraSite gateway.

The available event types are:

- **Pre-State Change**
- **Post-State Change**
- **OnTrigger**

- b. Click **Save** to save the parameter settings.
6. After you configure the parameters for all of the actions in the list, click **Save** to save the updated policy.
7. When you are ready to put the policy into effect, activate it.

Configuring Unpublish from API Portal Policy

Use the following procedure to configure the **Unpublish from API-Portal** action in the **Unpublish from API-Portal** policy. The policy removes specified API metadata bundle from the API Portal registry:

➤ To configure the unpublish from API Portal policy

1. In CentraSite Control, go to **Policies > Design/Change Time**.
2. Enable the **Show Predefined Policies** option.
A list of the predefined policies appears.
3. Click the Unpublish from API-Portal policy.
4. If the policy is active, deactivate it.
5. In the **Actions** tab, click the **Unpublish from API-Portal** action. In the Edit Action Parameters page, do the following.
 - a. Define the policy parameters:

API-Portal

(Optional). (String). (Array). Name of the API Portal gateway from which the API metadata is to be removed.

Note:

However, if you want to execute this policy in a **Pre-State Change** or **Post-State Change** event that is not provided by default, then you *must* specify a value for this field.

The available event types are:

- **Pre-Delete**
 - **Post-Delete**
 - **Pre-State Change**
 - **Post-State Change**
 - **OnTrigger**
- b. Click **Save** to save the parameter settings.
 6. After you configure the parameters for all of the actions in the list, click **Save** to save the updated policy.
 7. When you are ready to put the policy into effect, activate it.

16 **CentraSite and API Gateway Integration**

■ CentraSite and API Gateway Integration	1566
■ API Gateway Asset Mapping Details	1566
■ Virtual Service Mapping Details	1569
■ Alias Mapping Details	1572
■ Runtime Policy Mapping Details	1581
■ Consumer Application Mapping Details	1643
■ Modifications to Error Codes and Responses for Runtime Policies	1647

CentraSite and API Gateway Integration

Beginning with version 10.1, CentraSite supports publishing of Virtual Service assets from an instance of CentraSite to an instance of webMethods API Gateway. The CentraSite run-time policy configuration is disabled, by default and the configuration must be done from webMethods API Gateway. If you want to use CentraSite for run-time policy configuration, then you must enable CentraSite as a run-time application. For steps to enable the CentraSite run-time policies, see [“Enabling CentraSite Run-Time Aspects” on page 951](#).

Note:

It is recommended that CentraSite must be used as a design-time application with webMethods API Gateway as the run-time application. If you want to use Mediator as the run-time gateway, then enable CentraSite run-time policies following the steps listed in the [“Enabling CentraSite Run-Time Aspects” on page 951](#) section.

You can publish Virtual SOAP Services and Virtual REST Services from CentraSite to API Gateway in the following ways:

- Through CentraSite Business UI
- Through CentraSite Command Line tool

When publishing the Virtual Service assets, some of the associated runtime components are also published from CentraSite to API Gateway. The published runtime components are:

1. Runtime Policies
2. Runtime Aliases
3. Consumer Applications
4. API Gateway Asset
5. Virtual Services Assets

API Gateway Asset Mapping Details

This section describes how the API Gateway assets and their property values defined and published from CentraSite are mapped into API Gateway.

An API Gateway asset is published from CentraSite to API Gateway when a Virtual Service associated with the API Gateway asset is published from CentraSite to API Gateway.

Publishing of API Gateway assets from CentraSite to API Gateway is performed by invoking the API Gateway Deployer Service.

API Gateway Asset

The following table summarizes the mapping of an API Gateway asset:

CentraSite	API Gateway	Notes
Name	Communication: <ul style="list-style-type: none"> ■ Target name 	
Description	Not applicable	
Gateway : API Gateway	Not applicable	
Organization	Not applicable	
CentraSite Communication Information (API Gateway to CentraSite) <ul style="list-style-type: none"> ■ CentraSite Endpoint 	Communication: <ul style="list-style-type: none"> ■ Hostname ■ UDDI port 	
CentraSite Communication Information (API Gateway to CentraSite) <ul style="list-style-type: none"> ■ Username 	Communication: <ul style="list-style-type: none"> ■ Username 	
CentraSite Communication Information (API Gateway to CentraSite) <ul style="list-style-type: none"> ■ Password 	Not applicable	The parameter Password in CentraSite is not mapped with any of the parameters in API Gateway.
API Gateway Communication Information (CentraSite to API Gateway) <ul style="list-style-type: none"> ■ API Gateway Endpoint 	Not applicable	
API Gateway Communication Information (CentraSite to API Gateway) <ul style="list-style-type: none"> ■ API Gateway WebApp URL 	Web application load balancer URL	The parameter API Gateway WebApp URL in CentraSite maps to the parameter Web application load balancer URL in the Load balancer configuration page (go to <Username> > Administration > General > Load balancer) in API Gateway.
API Gateway Communication Information (CentraSite to API Gateway)	Communication: <ul style="list-style-type: none"> ■ Username 	The parameter Password in CentraSite is not mapped with any of the parameters in API Gateway.

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> Use CentraSite Credentials 		
API Gateway Communication Information (CentraSite to API Gateway) <ul style="list-style-type: none"> Username 	Username	The parameter Username in CentraSite maps to the default Username to login to API Gateway.
API Gateway Communication Information (CentraSite to API Gateway) <ul style="list-style-type: none"> Password 	Password	The parameter Password in CentraSite maps to the default Password to login to API Gateway.
API Gateway Communication Information (CentraSite to API Gateway) <ul style="list-style-type: none"> Sandbox 	Not applicable	
Not applicable	SNMP: <ul style="list-style-type: none"> Transport SNMP: <ul style="list-style-type: none"> Hostname SNMP: <ul style="list-style-type: none"> Port SNMP: <ul style="list-style-type: none"> Username SNMP: <ul style="list-style-type: none"> Authorization protocol SNMP: <ul style="list-style-type: none"> Privacy protocol Send SNMP traps to CentraSite	The SNMP configuration details in CentraSite maps to the configuration details of a SNMP server destination (in the Integration Server Administrator , go to Solutions > Mediator > Administration > SNMP).

CentraSite	API Gateway	Notes
	Force reset CentraSite communication and SNMP details	

Virtual Service Mapping Details

This section describes how the Virtual Services and their property values defined and published from CentraSite are mapped into API Gateway.

Publishing and unpublishing of Virtual Service assets from CentraSite to API Gateway are performed by invoking the API Gateway Deployer Service.

Virtual REST Services

Virtual Service Definition (VSD) of a Virtual REST Service asset contains the service name, service key, and the Swagger definition of the asset. API Gateway creates a REST API for the REST service that is defined by the VSD. A universally unique identifier (UUID) of the REST API is extracted from the VSD.

Note:

When a Virtual REST Service with an UUID, for example, `uddi:22beb489-2ba9-44c8-b189-5855e1d4d1ad`, is published from CentraSite to API Gateway, any existing REST API with the same UUID `uddi:22beb489-2ba9-44c8-b189-5855e1d4d1ad` in API Gateway is overwritten.

The following table summarizes the mapping of a Virtual REST Service in CentraSite to a REST API in API Gateway:

CentraSite	API Gateway
Virtual REST Service	REST API
Name	Name
Type: Virtual REST Service	Type: REST
Organization	Not applicable
Version	Version
Description	Description
Import From a Specification File	Type
<ul style="list-style-type: none"> ■ Swagger-2.0 ■ RAML-0.8 	<ul style="list-style-type: none"> ■ Swagger ■ RAML

CentraSite	API Gateway
Import a File: File	Select file
Import a File: URL	URL
Advanced Settings: Credentials	URL: Protected
<ul style="list-style-type: none"> ■ Username ■ Password 	<ul style="list-style-type: none"> ■ Username ■ Password
Not applicable	Maturity State
Not applicable	API grouping
Policy Actions	Policies
Key	API Identifier

Note:

By default, the REST APIs published to API Gateway are in Active state.

Virtual SOAP Services

Virtual Service Definition (VSD) of a Virtual Service asset contains the service name, service key, and the WSDL definition of the asset. API Gateway creates a SOAP API for the Web service that is defined by the VSD. A universally unique identifier (UUID) of the SOAP API is extracted from the VSD.

Note:

When a Virtual SOAP Service with an UUID, for example, `uddi:5f0ad20e-9bdd-11e4-9184-dc550a8f1855`, is published from CentraSite to API Gateway, any existing SOAP API with the same UUID `uddi:5f0ad20e-9bdd-11e4-9184-dc550a8f1855` in API Gateway is overwritten.

The following table summarizes the mapping of a Virtual SOAP Service in CentraSite to a SOAP API in API Gateway:

CentraSite	API Gateway
Virtual SOAP Service	SOAP API
Name	Name
Type: Virtual SOAP Service	Type: SOAP
Organization	Not applicable
Version	Version

CentraSite	API Gateway
Description	Description
Not applicable	Type ■ WSDL
Import a File: File	Select file
Import a File: URL	URL
Advanced Settings: Credentials ■ Username ■ Password	URL: Protected ■ Username ■ Password
Not applicable	Maturity State
Not applicable	API grouping
Policy Actions	Policies
Key	API Identifier

Note:

By default, the SOAP APIs published to API Gateway are in Active state.

Virtual OData Services

Virtual Service Definition (VSD) of a Virtual OData Service asset contains the service name, service key, and the Swagger definition of the asset. API Gateway creates an OData API for the OData service that is defined by the VSD. A universally unique identifier (UUID) of the OData API is extracted from the VSD.

Note:

When a Virtual OData Service with an UUID, for example, `uddi:9f0ad41e-9crd-11e4-9172-dc550a8f2345`, is published from CentraSite to API Gateway, any existing OData API with the same UUID `uddi:9f0ad41e-9crd-11e4-9172-dc550a8f2345` in API Gateway is overwritten.

The following table summarizes the mapping of a Virtual OData Service in CentraSite to an OData API in API Gateway:

CentraSite	API Gateway
Virtual OData Service	OData API
Name	Name
Type: Virtual OData Service	Type: OData

CentraSite	API Gateway
Organization	Not applicable
Version	Version
Description	Description
Import From a Specification File	Not applicable
<ul style="list-style-type: none"> ■ Swagger-2.0 ■ RAML-0.8 	
Import a File: File	Not applicable
Import a File: URL	URL
Advanced Settings: Credentials	URL: Protected
<ul style="list-style-type: none"> ■ Username ■ Password 	<ul style="list-style-type: none"> ■ Username ■ Password
Not applicable	Maturity State
Not applicable	API grouping
Policy Actions	Policies
Key	API Identifier

Note:

By default, the OData APIs published to API Gateway are in Active state.

Alias Mapping Details

This section describes how the aliases and their property values defined and published from CentraSite are mapped into API Gateway.

An alias is published from CentraSite to API Gateway when a Virtual Service associated with the alias is published from CentraSite to API Gateway.

Publishing of aliases from CentraSite to API Gateway is performed by invoking the API Gateway Deployer Service. The Deployer Service creates an alias in API Gateway for each `aliasDetails` element in the API's VSD.

List of Aliases

The following table summarizes the mapping of Aliases:

CentraSite	API Gateway	Notes
Simple Alias	Simple Alias	
Endpoint Alias	Endpoint Alias	
Secure Alias	Secure Alias	
Transformation Alias	Transformation Alias	Beginning with version 10.1, API Gateway supports Transformation Alias that holds a list of XSLT style sheets and can be used in the XSLT Transformation policies for request and response processing.
webMethods Integration Server Alias	webMethods Integration Server Alias	Beginning with version 10.1, API Gateway supports webMethods Integration Server Alias that holds the ESB service value and can be used to invoke the Invoke webMethods IS policy for request and response processing.

Simple Alias

The following table summarizes the mapping for a simple alias:

CentraSite	API Gateway	Notes
Alias Key	Not applicable	
Runtime alias type: Simple Alias	Type: ■ Simple Alias	
Name	Name	When a simple alias with a name, for example, TestAlias, is published from CentraSite, any existing alias with the same name TestAlias in API Gateway is overwritten.
Description	Description	
Default value	Default value	
Stage-specific values	Not applicable	

Endpoint Alias

The following table summarizes the mapping for an endpoint alias:

CentraSite	API Gateway	Notes
Alias Key	Not applicable	
Runtime alias type: Endpoint Alias	Type: <ul style="list-style-type: none"> ■ Endpoint Alias 	
Name	Name	When an endpoint alias with a name, for example, <code>TestAlias</code> , is published from CentraSite, any existing alias with the same name <code>TestAlias</code> in API Gateway is overwritten.
Description	Description	
Default value	Endpoint URI	
SOAP optimization method <ul style="list-style-type: none"> ■ None ■ MTOM ■ SWA 	Optimization technique <ul style="list-style-type: none"> ■ NONE ■ MTOM ■ SwA 	
Value	Endpoint URI	
Connection timeout	Connection timeout	
Read timeout	Read timeout	
Keystore alias	Keystore alias	
Client certificate alias	Key alias	
WS-Security header <ul style="list-style-type: none"> ■ Pass all security headers ■ Remove processed security headers 	Pass WS-Security Headers	
Stage-specific values	Not applicable	

Secure Alias

The following table summarizes the mapping for a secure alias:

CentraSite	API Gateway	Notes
Alias Key	Not applicable	
Runtime alias type: Secure Alias	Type	
	<ul style="list-style-type: none"> ■ HTTP Transport Security Alias ■ SOAP Message Security Alias 	
Authentication scheme: OAuth2 Authentication	Type: HTTP Transport security alias	
<ul style="list-style-type: none"> ■ OAuth2 token ■ Stage-specific values 	Authentication scheme: OAuth2 Authenticate using <ul style="list-style-type: none"> ■ Custom credentials <ul style="list-style-type: none"> ■ OAuth2 token ■ Incoming OAuth token 	
Authentication scheme: NTLM Authentication	Type: HTTP Transport security alias	
<ul style="list-style-type: none"> ■ Username ■ Password ■ Domain ■ Stage-specific values 	Authentication scheme: NTLM Authenticate using <ul style="list-style-type: none"> ■ Custom credentials <ul style="list-style-type: none"> ■ Username ■ Password ■ Domain ■ Incoming HTTP basic auth credentials ■ Transparent 	
Authentication scheme: HTTP Basic Authentication	Type: HTTP Transport security alias	
<ul style="list-style-type: none"> ■ Username ■ Password ■ Domain 	Authentication scheme: Basic Authenticate using	

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> ■ Stage-specific values 	<ul style="list-style-type: none"> ■ Custom credentials <ul style="list-style-type: none"> ■ Username ■ Password ■ Domain ■ Incoming HTTP basic auth credentials 	
<p>Authentication scheme: Kerberos Authentication</p>	<p>Type: HTTP Transport security alias</p>	
<ul style="list-style-type: none"> ■ Client Principal ■ Client Password ■ Service Principal ■ Service Principal Name Form <ul style="list-style-type: none"> ■ username ■ hostbased ■ Stage-specific values 	<p>Authentication scheme: Kerberos</p> <p>Authenticate using</p> <ul style="list-style-type: none"> ■ Custom credentials <ul style="list-style-type: none"> ■ Client Principal ■ Client Password ■ Service Principal ■ Service Principal Name Form <ul style="list-style-type: none"> ■ username ■ hostbased ■ Delegate incoming credentials <ul style="list-style-type: none"> ■ Client Principal ■ Client Password ■ Service Principal ■ Service Principal Name Form <ul style="list-style-type: none"> ■ username ■ hostbased ■ Incoming HTTP basic auth credentials 	

CentraSite	API Gateway	Notes
	<ul style="list-style-type: none"> ■ Service Principal ■ Service Principal Name Form ■ username ■ hostbased 	
Not applicable	<p>Type: HTTP Transport security alias</p> <p>Authentication scheme: JWT</p> <p>Authenticate using</p> <ul style="list-style-type: none"> ■ Incoming JWT 	
Not applicable	<p>Type: SOAP message security alias</p> <p>Authentication scheme: NONE</p> <p>Signing configurations</p> <ul style="list-style-type: none"> ■ Keystore alias ■ Key alias <p>Encryption configurations</p> <ul style="list-style-type: none"> ■ Truststore alias ■ Certificate alias 	
Not applicable	<p>Type: SOAP message security alias</p> <p>Authentication scheme: WSS Username</p> <p>Authenticate using</p> <ul style="list-style-type: none"> ■ Custom credentials <ul style="list-style-type: none"> ■ Username ■ Password <p>Signing configurations</p>	

CentraSite	API Gateway	Notes
	<ul style="list-style-type: none"> ■ Keystore alias ■ Key alias <p>Encryption configurations</p> <ul style="list-style-type: none"> ■ Truststore alias ■ Certificate alias 	
<p>Authentication scheme: Kerberos Authentication</p> <ul style="list-style-type: none"> ■ Client Principal ■ Client Password ■ Service Principal ■ Service Principal Name Form <ul style="list-style-type: none"> ■ username ■ hostbased 	<p>Type: SOAP message security alias</p> <p>Authentication scheme: Kerberos</p> <p>Authenticate using</p> <ul style="list-style-type: none"> ■ Custom credentials <ul style="list-style-type: none"> ■ Client Principal ■ Client Password ■ Service Principal ■ Service Principal Name Form <ul style="list-style-type: none"> ■ username ■ hostbased ■ Delegate incoming credentials <ul style="list-style-type: none"> ■ Client Principal ■ Client Password ■ Service Principal ■ Service Principal Name Form <ul style="list-style-type: none"> ■ username ■ hostbased ■ Incoming HTTP basic auth credentials <ul style="list-style-type: none"> ■ Service Principal 	

CentraSite	API Gateway	Notes
	<ul style="list-style-type: none"> ■ Service Principal Name Form ■ username ■ hostbased <p>Signing configurations</p> <ul style="list-style-type: none"> ■ Keystore alias ■ Key alias <p>Encryption configurations</p> <ul style="list-style-type: none"> ■ Truststore alias ■ Certificate alias 	
Not applicable	<p>Type: SOAP message security alias</p> <p>Authentication scheme: SAML</p> <p>SAML issuer</p> <p>Signing configurations</p> <ul style="list-style-type: none"> ■ Keystore alias ■ Key alias <p>Encryption configurations</p> <ul style="list-style-type: none"> ■ Truststore alias ■ Certificate alias 	
Name	Name	When a secure alias with a name, for example, TestAlias, is published from CentraSite, any existing alias with the same name TestAlias in API Gateway is overwritten.
Description	Description	
Stage-specific values	Not applicable	

webMethods Integration Server Alias

The following table summarizes the mapping for a webMethods Integration Server alias:

CentraSite	API Gateway	Notes
Alias Key	Not applicable	
Runtime alias type: webMethods Integration Server Alias	Type: <ul style="list-style-type: none"> webMethods IS Service Alias 	
Name	Name	When a webMethods Integration Server alias with a name, for example, TestAlias, is published from CentraSite, any existing alias with the same name TestAlias in API Gateway is overwritten.
Description	Description	
Default value	Not applicable	
Not applicable	Type: webMethods IS Service alias <ul style="list-style-type: none"> Service name 	

Transformation Alias

The following table summarizes the mapping for a transformation alias:

CentraSite	API Gateway	Notes
Alias Key	Not applicable	
Runtime alias type: Transformation Alias	Type: <ul style="list-style-type: none"> XSLT Transformation Alias 	
Name	Name	When an XSLT transformation alias with a name, for example, TestAlias, is published from CentraSite, any existing alias with the same name TestAlias in API Gateway is overwritten.
Description	Description	
Default value	Not applicable	
	Type: XSLT Transformation alias	

CentraSite	API Gateway	Notes
	<ul style="list-style-type: none"> Select Transformation file 	

Runtime Policy Mapping Details

This section describes how the runtime policies and their property values defined and published from CentraSite are mapped into API Gateway.

A runtime policy is published from CentraSite to API Gateway when a Virtual Service associated with the policy is published from CentraSite to API Gateway.

Publishing of runtime policies from CentraSite to API Gateway is performed by invoking the API Gateway Deployer Service.

Evaluate Kerberos

You can secure Web Service APIs using client's Kerberos token credentials. This policy action:

- Identifies the application against either the Registered Applications list or the Global Applications list in API Gateway.
- Validates the client's kerberos token against the list of users in the Integration Server on which API Gateway is running.

You can select the level at which the Kerberos identification and authentication should be enforced for APIs:

- Message level
- Transport layer

The **Evaluate Kerberos** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the following policies under the **Identify & Access** stage in API Gateway:

- If this policy action is set to secure SOAP API at the message level at the message level using the parameter `Enforcement Point: Message Level` in CentraSite, then the policy **Inbound Authentication - Message** with the parameter `Kerberos Token Authentication` is included to the API's policy list in API Gateway.
- If this policy action is set to secure SOAP or REST API at the transport layer using the parameter `Enforcement Point: Transport Level` in CentraSite, then the policy **Inbound Authentication - Transport** with the parameter `Kerberos Token Authentication` is included to the API's policy list in API Gateway.
- If this policy action is set to identify the application (against a list of global or registered consumers) using the parameter `Identify Consumer: Global Consumers/ Registered Consumers`, then the policy **Identify & Authorize Application** is included to the API's policy list in API Gateway.

- If this policy action is set to NOT identify the application (against a list of global or registered consumers) using the parameter `Identify Consumer: Do Not Identify`, then the policy **Identify & Authorize Application** is NOT included to the API's policy list in API Gateway.

Enforcement Point: Message Level Security

The following table summarizes the mapping of the policy enforcement at the message level:

CentraSite	API Gateway	Notes
Service Principal Name Form: <ul style="list-style-type: none"> ■ User 	Service Principal Name Form: <ul style="list-style-type: none"> ■ Username 	Beginning with version 10.1, the parameter Service Principal Name Form is removed from CentraSite and API Gateway. This is because the service principal name form set to <code>Host</code> or <code>Hostbased</code> is no longer supported in Integration Server. Therefore, whenever the Evaluate Kerberos policy is defined for an API in CentraSite or API Gateway, the service principal name form defaults to <code>User</code> or <code>Username</code> .
Service Principal Name Form: <ul style="list-style-type: none"> ■ Host 	Service Principal Name Form: <ul style="list-style-type: none"> ■ Hostbased 	
Service Principal Name	Service Principal Name	
Service Principal Password	Service Principal Password	
Identify Consumer: <ul style="list-style-type: none"> ■ Do Not Identify 	Not applicable	When the parameter <code>Identify Consumer</code> is set to <code>Do Not Identify</code> in CentraSite, the policy Identify and Authorize Application that is used to identify an application is NOT included to the API's list of policies in API Gateway.
Identify Consumer: <ul style="list-style-type: none"> ■ Global Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> ■ Global applications 	When the parameter <code>Identify Consumer</code> is set to <code>Global Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a global consumer application is included to the API's list of policies in API Gateway. Also, the parameter <code>Identification Type</code> is set to <code>Kerberos Token</code> in the policy Identify and Authorize Application .
Identify Consumer: <ul style="list-style-type: none"> ■ Registered Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> ■ Registered applications 	When the parameter <code>Identify Consumer</code> is set to <code>Registered Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a registered consumer

CentraSite	API Gateway	Notes
		application is included to the API's list of policies in API Gateway. Also, the parameter <code>Identification Type</code> is set to <code>Kerberos Token</code> in the policy Identify and Authorize Application .

Enforcement Point: Transport Layer Security

The following table summarizes the mapping of the policy enforcement at the transport layer:

CentraSite	API Gateway	Notes
Service Principal Name Form: <ul style="list-style-type: none"> ■ User 	Service Principal Name Form: <ul style="list-style-type: none"> ■ Username 	Beginning with version 10.1, the parameter Service Principal Name Form is removed from CentraSite and API Gateway. This is because the service principal name form set to <code>Host</code> or <code>Hostbased</code> is no longer supported in Integration Server. Therefore, whenever the Evaluate Kerberos policy is defined for an API in CentraSite or API Gateway, the service principal name form defaults to <code>User</code> or <code>Username</code> .
Service Principal Name Form: <ul style="list-style-type: none"> ■ Host 	Service Principal Name Form: <ul style="list-style-type: none"> ■ Hostbased 	
Service Principal Name	Service Principal Name	
Service Principal Password	Service Principal Password	
Identify Consumer: <ul style="list-style-type: none"> ■ Do Not Identify 	Not applicable	When the parameter <code>Identify Consumer</code> is set to <code>Do Not Identify</code> in CentraSite, the policy Identify and Authorize Application that is used to identify an application is NOT included to the API's list of policies in API Gateway.
Identify Consumer: <ul style="list-style-type: none"> ■ Global Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> ■ Global applications 	When the parameter <code>Identify Consumer</code> is set to <code>Global Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a global consumer application is included to the API's list of policies in API Gateway. Also, the parameter <code>Identification Type</code> is set to <code>Kerberos Token</code> in the policy Identify and Authorize Application .

CentraSite	API Gateway	Notes
Identify Consumer: <ul style="list-style-type: none"> ■ Registered Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> ■ Registered applications 	When the parameter <code>Identify Consumer</code> is set to <code>Registered Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a registered consumer application is included to the API's list of policies in API Gateway. Also, the parameter <code>Identification Type</code> is set to <code>Kerberos Token</code> in the policy Identify and Authorize Application .

Evaluate HTTP Basic Authentication

You can secure Web Service APIs using HTTP basic authentication. This policy action:

- Identifies the application against either the Registered Applications list or the Global Applications list in API Gateway.
- Validates the client's credentials contained in the request's Authorization header against the list of users in the Integration Server on which API Gateway is running.

The **Evaluate HTTP Basic Authentication** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the following policies under the **Identify & Access** stage in API Gateway:

- If this policy action is set to identify the application (against a list of global or registered Applications) using the parameter `Identify Consumer: Global Consumers/ Registered Consumers`, then the policy **Identify & Authorize Application** is included to the API's policy list in API Gateway.
- If this policy action is set to NOT identify the application (against a list of global or registered Applications) using the parameter `Identify Consumer: Do Not Identify`, then the policy **Identify & Authorize Application** is NOT included to the API's policy list in API Gateway.
- If this policy action is set to validate the client's credentials in the request's Authorization header against the list of users in Integration Server using the parameter `Authenticate User` set to `true` in CentraSite, then the policy **Inbound Authentication - Transport** is included to the API's policy list in API Gateway.
- If this policy action is set to NOT validate the client's credentials in the request's Authorization header against the list of users in Integration Server using the parameter `Authenticate User` set to `false` in CentraSite, then the policy **Inbound Authentication - Transport** is NOT included to the API's policy list in API Gateway.
- If this policy action is set to NOT validate the client's credentials in the request's Authorization header (using the parameter `Authenticate User` set to `false`) and NOT identify the application (using the parameter `Identify Consumer` set to `Do Not Identify`), then the policy **Validate HTTP Headers** is included to the API's policy list in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Identify Consumer: <ul style="list-style-type: none"> ■ Do Not Identify 	Not applicable	When the parameter <code>Identify Consumer</code> is set to <code>Do Not Identify</code> in CentraSite, the policy Identify and Authorize Application that is used to identify an application is NOT included to the API's list of policies in API Gateway.
Identify Consumer: <ul style="list-style-type: none"> ■ Global Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> ■ Global applications 	When the parameter <code>Identify Consumer</code> is set to <code>Global Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a global consumer application is included to the API's list of policies in API Gateway. Also, the parameter <code>Identification Type</code> is set to <code>Kerberos Token</code> in the policy Identify and Authorize Application .
Identify Consumer: <ul style="list-style-type: none"> ■ Registered Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> ■ Registered applications 	When the parameter <code>Identify Consumer</code> is set to <code>Registered Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a registered consumer application is included to the API's list of policies in API Gateway. Also, the parameter <code>Identification Type</code> is set to <code>Kerberos Token</code> in the policy Identify and Authorize Application .
Authenticate User (check box) (check box)	Not applicable	<ul style="list-style-type: none"> ■ When the parameter <code>Authenticate User</code> check box is selected (set to <code>true</code>) in CentraSite, the policy Inbound Authentication - Transport with the parameter <code>HTTP Basic Authentication</code> that is used to secure the API at the transport layer is included to the API's list of policies in API Gateway. ■ When the parameter <code>Authenticate User</code> check box is NOT selected (set to <code>false</code>) in CentraSite, the policy Inbound Authentication - Transport that is used to secure the API at the transport layer is NOT

CentraSite	API Gateway	Notes
		included to the API's list of policies in API Gateway. Also, if the parameter <code>Identify Consumer</code> is set to <code>Do Not Identify</code> in CentraSite, the policy Validate HTTP Headers that is used to validate the presence of HTTP headers, or header values, or both in incoming API requests is included to the API's list of policies in API Gateway.

Evaluate Client Certificate for SSL Connectivity

You can secure Web Service APIs using client's SSL certificate. This policy action identifies the application against either the Registered Applications list or the Global Applications list in API Gateway.

The **Evaluate Client Certificate for SSL Connectivity** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the policy **Identify and Authorize Application** under the **Identify & Access** stage in API Gateway

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Identify Consumer: <ul style="list-style-type: none"> ■ Global Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> ■ Global applications 	When the parameter <code>Identify Consumer</code> is set to <code>Global Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a global consumer application is included to the API's list of policies in API Gateway. Also, the parameter <code>Identification Type</code> is set to <code>SSL Certificate</code> in the policy Identify and Authorize Application .
Identify Consumer: <ul style="list-style-type: none"> ■ Registered Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> ■ Registered applications 	When the parameter <code>Identify Consumer</code> is set to <code>Registered Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a registered consumer application is included to the API's list of policies in API Gateway. Also, the parameter <code>Identification Type</code> is set to <code>SSL Certificate</code> in the policy Identify and Authorize Application .

Evaluate Hostname

You can secure Web Service APIs using client's host name. This policy action identifies the application against either the Registered Applications list or the Global Applications list in API Gateway.

The **Evaluate Hostname** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the policy **Identify and Authorize Application** under the **Identify & Access** stage in API Gateway

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Identify Consumer: <ul style="list-style-type: none"> ■ Global Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> ■ Global applications 	When the parameter <code>Identify Consumer</code> is set to <code>Global Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a global consumer application is included to the API's list of policies in API Gateway. Also, the parameter <code>Identification Type</code> is set to <code>Hostname Address</code> in the policy Identify and Authorize Application .
Identify Consumer: <ul style="list-style-type: none"> ■ Registered Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> ■ Registered applications 	When the parameter <code>Identify Consumer</code> is set to <code>Registered Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a registered consumer application is included to the API's list of policies in API Gateway. Also, the parameter <code>Identification Type</code> is set to <code>Hostname Address</code> in the policy Identify and Authorize Application .

Evaluate IP Address

You can secure Web Service APIs using client's IP address. This policy action identifies the application against either the Registered Applications list or the Global Applications list in API Gateway.

The **Evaluate Hostname** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the policy **Identify and Authorize Application** under the **Identify & Access** stage in API Gateway

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Identify Consumer: <ul style="list-style-type: none"> Global Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> Global applications 	When the parameter <code>Identify Consumer</code> is set to <code>Global Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a global consumer application is included to the API's list of policies in API Gateway. Also, the parameter <code>Identification Type</code> is set to <code>IP Address Range</code> in the policy Identify and Authorize Application .
Identify Consumer: <ul style="list-style-type: none"> Registered Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> Registered applications 	When the parameter <code>Identify Consumer</code> is set to <code>Registered Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a registered consumer application is included to the API's list of policies in API Gateway. Also, the parameter <code>Identification Type</code> is set to <code>IP Address Range</code> in the policy Identify and Authorize Application .

Evaluate XPath Expression

You can secure Web Service APIs using client's authentication credentials that are represented as an XPath expression. This policy action identifies the application against either the Registered Applications list or the Global Applications list in API Gateway.

The **Evaluate XPath Expression** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the policy **Identify & Authorize Application** under the **Identify & Access** stage in API Gateway as follows:

- If this policy action is set to identify the application (against a list of global or registered Applications) using the parameter `Identify Consumer: Global Consumers/ Registered Consumers`, then the policy **Identify & Authorize Application** is included to the API's policy list in API Gateway.
- But, if this policy action is set to NOT identify the application (against a list of global or registered Applications) using the parameter `Identify Consumer: Do Not Identify`, it is not possible to publish the API from CentraSite to API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Identify Consumer:	Not applicable	When the parameter <code>Identify Consumer</code> is set to <code>Do Not Identify</code> in CentraSite,

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> Do Not Identify 		<p>publishing of API from CentraSite to API Gateway fails with the following exception: <i>Could not deploy the asset <asset_name> on given target <API Gateway_name>.</i></p> <p>To work around this issue, you have to set the value of the parameter <code>Identify Consumer</code> to either <code>Global Consumers</code> or <code>Registered Consumers</code>, and then publish the API to API Gateway.</p>
<p>Identify Consumer:</p> <ul style="list-style-type: none"> Global Consumers 	<p>Application Lookup Condition:</p> <ul style="list-style-type: none"> Global applications 	<p>When the parameter <code>Identify Consumer</code> is set to <code>Global Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a global consumer application is included to the API's list of policies in API Gateway. Also, the parameter <code>Identification Type</code> is set to <code>XPath</code> expression in the policy Identify and Authorize Application.</p>
<p>Identify Consumer:</p> <ul style="list-style-type: none"> Registered Consumers 	<p>Application Lookup Condition:</p> <ul style="list-style-type: none"> Registered applications 	<p>When the parameter <code>Identify Consumer</code> is set to <code>Registered Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a registered consumer application is included to the API's list of policies in API Gateway. Also, the parameter <code>Identification Type</code> is set to <code>XPath</code> expression in the policy Identify and Authorize Application.</p>
<p>Namespace:</p> <ul style="list-style-type: none"> Prefix 	<p>Namespace:</p> <ul style="list-style-type: none"> Namespace Prefix 	
<p>Namespace:</p> <ul style="list-style-type: none"> URI 	<p>Namespace:</p> <ul style="list-style-type: none"> Namespace URI 	
<p>XPath Expression</p>	<p>Identification Query:</p> <ul style="list-style-type: none"> Query Expression 	

Evaluate OAuth2 Token

You can secure Web Service APIs using client's OAuth2 token credentials. This policy action identifies the application against either the Registered Applications list or the Global Applications list in API Gateway.

The **Evaluate OAuth2 Token** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the policy **Identify & Authorize Application** under the **Identify & Access** stage in API Gateway as follows:

- If this policy action is set to identify the application (against a list of global or registered Applications) using the parameter `Identify Consumer: Global Applications / Registered Applications`, then the policy **Identify & Authorize Application** is included to the API's policy list in API Gateway.
- If this policy action is set to NOT identify the application (against a list of global or registered Applications) using the parameter `Identify Consumer: Do Not Identify`, then the policy **Identify & Authorize Application** is NOT included to the API's policy list in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Identify Consumer: <ul style="list-style-type: none"> ■ Do Not Identify <hr/> Identify Consumer: <ul style="list-style-type: none"> ■ Global Consumers <hr/> Identify Consumer: <ul style="list-style-type: none"> ■ Registered Consumers 	Not applicable	When the parameter <code>Identify Consumer</code> is set to any of the three options in CentraSite, the policy Identify and Authorize Application that is used to identify an application is included to the API's list of policies in API Gateway. The parameter Identification Type is set to <code>OAuth2 Token</code> in the policy Identify and Authorize Application . Also, the policy is internally defined to identify the application against the Registered Applications list in API Gateway.
Authenticate Access Token: <ul style="list-style-type: none"> ■ True ■ False 	Not applicable	

Usage of Do Not Identify with remote Integration Server acting as an OAuth 2.0 Authorization Server:

- In CentraSite, when the parameter `Identify Consumer` is set to `Do Not Identify`, OAuth 2.0 validation occurs against a remote Integration Server, which acts as an OAuth 2.0 authorization server. But the application is NOT identified against the list of OAuth 2.0 consumer applications in CentraSite.

- In API Gateway, when a remote Integration Server acts as an OAuth 2.0 authorization server, each application in API Gateway creates an OAuth 2.0 client in the remote Integration Server. Any incoming request from the application is validated against the remote Integration Server, which acts as an OAuth 2.0 authorization server. Also, the application is identified against the list of registered applications in API Gateway.
- For the OAuth 2.0 clients in remote Integration Server that are not mapped to the OAuth 2.0 consumer applications in CentraSite, API Gateway offers a migration utility that uses the webMethods IS Service: `pub.apigateway.migration.remoteoauth.migrateRemoteOauthClients`.

This service creates an application for each OAuth 2.0 client that contains an associated scope-id that is same as that of the API ID in API Gateway, and registers the created application to that API. API Gateway checks for the presence of scope-id against the ID of all APIs.

Note:

Software AG recommends you to invoke the webMethods IS Service only after you migrate all OAuth 2.0 APIs from CentraSite to API Gateway.

Pre-requisite: To run the webMethods IS Service is that a remote Integration Server should be configured as OAuth 2.0 authorization server in the Integration Server where API Gateway is running (in the Integration Server Administrator, go to **Security -> Oauth -> Edit OAuth Global Settings -> Authorization server**).

Evaluate WSS Username Token

You can secure SOAP APIs using client's WSS username token. This policy action:

- Identifies the application against either the Registered Applications list or the Global Applications list in API Gateway.
- Validates the client's WSS username token against the list of users in the Integration Server on which API Gateway is running.

The **Evaluate WSS Username Token** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the following policies under the **Identify & Access** stage in API Gateway as follows:

- If this policy action is defined for an API in CentraSite, then the policy **Inbound Authentication - Message** is included to the API's policy list in API Gateway.
- If this policy action is set to identify the application (against a list of global or registered Applications) using the parameter `Identify Consumer: Global Applications / Registered Applications`, then the policy **Identify & Authorize Application** is included to the API's policy list in API Gateway.
- If this policy action is set to NOT identify the application (against a list of global or registered Applications) using the parameter `Identify Consumer: Do Not Identify`, then the policy **Identify & Authorize Application** is NOT included to the API's policy list in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Identify Consumer: <ul style="list-style-type: none"> Do Not Identify 	Token Assertions: <ul style="list-style-type: none"> Require WSS Username token 	<p>When the parameter <code>Identify Consumer</code> is set to <code>Do Not Identify</code> in CentraSite, the policy Identify and Authorize Application that is used to identify an application is NOT included to the API's list of policies in API Gateway.</p> <p>However, the policy Inbound Authentication - Message with the parameter <code>Require WSS Username token</code> that is used to enforce the SOAP message security is included to the API's list of policies in API Gateway.</p>
Identify Consumer: <ul style="list-style-type: none"> Global Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> Global applications Token Assertions: <ul style="list-style-type: none"> Require WSS Username token 	<p>When the parameter <code>Identify Consumer</code> is set to <code>Global Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a global consumer application is included to the API's list of policies in API Gateway. The parameter <code>Identification Type</code> is set to <code>WS Security Username token</code> in the policy Identify and Authorize Application.</p> <p>Also the policy Inbound Authentication - Message with the parameter <code>Require WSS Username token</code> that is used to enforce the SOAP message security is included to the API's list of policies in API Gateway.</p>
Identify Consumer: <ul style="list-style-type: none"> Registered Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> Registered applications Token Assertions: <ul style="list-style-type: none"> Require WSS Username token 	<p>When the parameter <code>Identify Consumer</code> is set to <code>Registered Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a registered consumer application is included to the API's list of policies in API Gateway. The parameter <code>Identification Type</code> is set to <code>Require WSS Username token</code> in the policy Identify and Authorize Application.</p> <p>Also the policy Inbound Authentication - Message with the parameter <code>Require WSS Username token</code></p>

CentraSite	API Gateway	Notes
		that is used to enforce the SOAP message security is included to the API's list of policies in API Gateway.

Evaluate WSS X.509 Certificate

You can secure SOAP APIs using client's x.509 certificate. This policy action:

- Identifies the application against either the Registered Applications list or the Global Applications list in API Gateway.
- Validates the client's x.509 certificate against the list of users the Integration Server on which API Gateway is running.

The **Evaluate WSS X.509 Certificate** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the following policies under the **Identify & Access** stage in API Gateway as follows:

- If this policy action is defined for an API in CentraSite, then the policy **Inbound Authentication - Message** is included to the API's policy list in API Gateway.
- If this policy action is set to identify the application (against a list of global or registered Applications) using the parameter `Identify Consumer: Global Applications / Registered Applications`, then the policy **Identify & Authorize Application** is included to the API's policy list in API Gateway.
- If this policy action is set to NOT identify the application (against a list of global or registered Applications) using the parameter `Identify Consumer: Do Not Identify`, then the policy **Identify & Authorize Application** is NOT included to the API's policy list in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Identify Consumer: <ul style="list-style-type: none"> ■ Do Not Identify 	Token Assertions: <ul style="list-style-type: none"> ■ Require X.509 Certificate 	<p>When the parameter <code>Identify Consumer</code> is set to <code>Do Not Identify</code> in CentraSite, the policy Identify and Authorize Application that is used to identify an application is NOT included to the API's list of policies in API Gateway.</p> <p>However, the policy Inbound Authentication - Message with the parameter <code>WS Security X.509 Certificate</code> that is used to enforce the SOAP message security is included to the API's list of policies in API Gateway.</p>

CentraSite	API Gateway	Notes
Identify Consumer: <ul style="list-style-type: none"> Global Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> Global applications Token Assertions: <ul style="list-style-type: none"> Require X.509 Certificate 	<p>When the parameter <code>Identify Consumer</code> is set to <code>Global Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a global consumer application is included to the API's list of policies in API Gateway. The parameter <code>Identification Type</code> is set to <code>WS Security X.509 Certificate</code> in the policy Identify and Authorize Application.</p> <p>Also the policy Inbound Authentication - Message with the parameter <code>Require X.509 Certificate</code> that is used to enforce the SOAP message security is included to the API's list of policies in API Gateway.</p>
Identify Consumer: <ul style="list-style-type: none"> Registered Consumers 	Application Lookup Condition: <ul style="list-style-type: none"> Registered applications Token Assertions: <ul style="list-style-type: none"> Require X.509 Certificate 	<p>When the parameter <code>Identify Consumer</code> is set to <code>Registered Consumers</code> in CentraSite, the policy Identify and Authorize Application that is used to identify a registered consumer application is included to the API's list of policies in API Gateway. The parameter <code>Identification Type</code> is set to <code>WS Security X.509 Certificate</code> in the policy Identify and Authorize Application.</p> <p>Also the policy Inbound Authentication - Message with the parameter <code>Require X.509 Certificate</code> that is used to enforce the SOAP message security is included to the API's list of policies in API Gateway.</p>

Require API Key Check

You can secure Web Service APIs using API Key authentication. This policy action identifies and validates the consumer against the list of registered applications in API Gateway.

The **Require API Key Check** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the policy **Identify & Authorize Application** in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Not applicable	Identification Type: <ul style="list-style-type: none"> API Key 	When the policy Require API Key Check is defined in CentraSite, the policy Identify and Authorize Application that is used to identify an application is included to the API's list of policies in API Gateway. The parameter <code>Identification Type</code> is set to <code>API Key</code> in the policy Identify and Authorize Application .

Require Encryption

You can secure SOAP APIs using encryption. This policy action mandates that an XML element of the SOAP request message must be encrypted.

The **Require Encryption** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the policy **Inbound Authentication - Message** under the **Identify & Access** stage in API Gateway. The policy definition is mapped into the section `Require Encryption`.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Encrypt By: Element	Encrypted Elements	
Namespace:	Namespace:	
<ul style="list-style-type: none"> Prefix 	<ul style="list-style-type: none"> Namespace Prefix 	
Encrypt By: Element	Encrypted Elements	
Namespace:	Namespace:	
<ul style="list-style-type: none"> URI 	<ul style="list-style-type: none"> Namespace URI 	
Encrypt By: Element	Encrypted Elements	
<ul style="list-style-type: none"> Element Required to be Encrypted 	<ul style="list-style-type: none"> XPath 	
Encrypt By: Part	Encrypted Parts	
Encrypt Part	<ul style="list-style-type: none"> Entire SOAP Body 	
<ul style="list-style-type: none"> Soap Body 	<ul style="list-style-type: none"> All SOAP Attachments 	
Encrypt By: Part	Encrypted Parts	
Encrypt SOAP headers	SOAP Header:	

CentraSite	API Gateway	Notes
■ Name	■ Header Name	
Encrypt By: Part	Encrypted Parts	
Encrypt SOAP headers	SOAP Header:	
■ Namespace	■ Header Namespace	

Require Signing

You can secure SOAP APIs using signature. This policy action mandates that an XML element of the SOAP request message must be signed.

The **Require Signing** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the policy **Inbound Authentication - Message** under the **Identify & Access** stage in API Gateway. The policy definition is mapped into the section **Require Signature**.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Sign By: Element	Signed Elements	
Namespace:	Namespace:	
■ Prefix	■ Namespace Prefix	
Sign By: Element	Signed Elements	
Namespace:	Namespace:	
■ URI	■ Namespace URI	
Sign By: Element	Signed Elements	
■ Element Required to be Signed	■ XPath	
Sign By: Part	Signed Parts	
Sign Part	■ Entire SOAP Body	
■ Soap Body	■ All SOAP Attachments	
Sign By: Part	Signed Parts	
Sign SOAP headers	SOAP Header:	
■ Name	■ Header Name	
Sign By: Part	Signed Parts	

CentraSite	API Gateway	Notes
Sign SOAP headers	SOAP Header:	
<ul style="list-style-type: none"> ■ Namespace 	<ul style="list-style-type: none"> ■ Header Namespace 	

Require SSL

You can secure SOAP APIs using SSL. This policy action mandates that client certificates must be sent only over a two-way SSL connection .

The **Require SSL** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the policy **Require HTTP / HTTPS** under the **Transport** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Client Certificate Required (check box)	Protocol: <ul style="list-style-type: none"> ■ HTTP ■ HTTPS 	<ul style="list-style-type: none"> ■ When the parameter <code>Client Certificate Required</code> check box is selected (set to true) in CentraSite, the policy Require HTTP / HTTPS with the parameter <code>HTTPS</code> is included (if not already included) to the API's list of policies in API Gateway. ■ When the parameter <code>Client Certificate Required</code> check box is not selected (set to false) in CentraSite, the default policy Require HTTP / HTTPS with the default parameter <code>HTTP</code> is included to the API's list of policies in API Gateway.

Require Timestamps

You can secure SOAP APIs using timestamps. This policy action mandates that timestamp must be included in the SOAP message header.

The **Require Timestamps** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the policy **Inbound Authentication - Message** under the **Identify & Access** stage in API Gateway. The policy definition is mapped into the section `Require Timestamp`.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Not applicable	Token Assertions: <ul style="list-style-type: none"> Require Timestamp 	When the policy action Require Timestamps is included to the list of policies in a SOAP API, the corresponding policy Inbound Authentication - Message with the parameter Token Assertions set to <code>Require Timestamp</code> is included to the API's list of policies in API Gateway.

Require WSS SAML Token

You can secure SOAP APIs using WS-Security SAML tokens. This policy action uses a SAML assertion token to validate the applications.

The **Require WSS SAML Token** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the policy **Inbound Authentication - Message** under the **Identify & Access** stage in API Gateway. The policy definition is mapped into the section `Require SAML Token`.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
	Token Assertions: <ul style="list-style-type: none"> Require SAML Token 	When the policy action Require WSS SAML Token is included to the list of policies in a SOAP API, the corresponding policy Inbound Authentication - Message with the parameter Token Assertions set to <code>Require SAML Token</code> is included to the API's list of policies in API Gateway.
SAML Version: <ul style="list-style-type: none"> SAML 1.1 SAML 2.0 	SAML Version: <ul style="list-style-type: none"> SAML 1.1 SAML 2.0 	
SAML Subject Confirmation: Bearer <ul style="list-style-type: none"> WS Trust Version 	SAML Subject Confirmation: Bearer of Token <ul style="list-style-type: none"> WS-Trust Version 	

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> ■ VERSION_05_02 ■ VERSION_05_12 	<ul style="list-style-type: none"> ■ WS-Trust 1.0 ■ WS-Trust 1.3 	
<ul style="list-style-type: none"> ■ Issuer Address 	<ul style="list-style-type: none"> ■ Encrypt Signature 	
<ul style="list-style-type: none"> ■ Metadata Reference Address 	<ul style="list-style-type: none"> ■ Issuer Address 	
<ul style="list-style-type: none"> ■ Key Size 	<ul style="list-style-type: none"> ■ Metadata Reference Address 	
<ul style="list-style-type: none"> ■ Request Security Token Template Parameters 	<ul style="list-style-type: none"> ■ Algorithm Suite 	
<ul style="list-style-type: none"> ■ Key 	<ul style="list-style-type: none"> ■ Basic128 	
<ul style="list-style-type: none"> ■ Value 	<ul style="list-style-type: none"> ■ Basic128Rsa15 ■ Basic128Sha256 ■ Basic128Sha256Rsa15 ■ Basic192 ■ Basic192Rsa15 ■ Basic192Sha256 ■ Basic192Sha256Rsa15 ■ Basic256 ■ Basic256Rsa15 ■ Basic256Sha256 ■ Basic256Sha256Rsa15 ■ TripleDe ■ TripleDesRsa15 ■ TripleDesSha256 	
	<ul style="list-style-type: none"> ■ Require Security Token Template Parameters 	
	<ul style="list-style-type: none"> ■ Key ■ Value 	
<p>SAML Subject Confirmation: Holder of Key (Asymmetric)</p>	<p>SAML Subject Confirmation: Holder of Key - Public</p>	
<ul style="list-style-type: none"> ■ WS Trust Version 	<ul style="list-style-type: none"> ■ WS-Trust Version 	

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> ■ VERSION_05_02 ■ VERSION_05_12 	<ul style="list-style-type: none"> ■ WS-Trust 1.0 ■ WS-Trust 1.3 	
<ul style="list-style-type: none"> ■ Algorithm Suite <ul style="list-style-type: none"> ■ Basic128 ■ Basic128Rsa15 ■ Basic128Sha256 ■ Basic128Sha256Rsa15 ■ Basic192 ■ Basic192Rsa15 ■ Basic192Sha256 ■ Basic192Sha256Rsa15 ■ Basic256 ■ Basic256Rsa15 ■ Basic256Sha256 ■ Basic256Sha256Rsa15 ■ TripleDe ■ TripleDesRsa15 ■ TripleDesSha256 	<ul style="list-style-type: none"> ■ Encrypt Signature ■ Issuer Address ■ Metadata Reference Address ■ Algorithm Suite <ul style="list-style-type: none"> ■ Basic128 ■ Basic128Rsa15 ■ Basic128Sha256 ■ Basic128Sha256Rsa15 ■ Basic192 ■ Basic192Rsa15 ■ Basic192Sha256 ■ Basic192Sha256Rsa15 ■ Basic256 ■ Basic256Rsa15 ■ Basic256Sha256 ■ Basic256Sha256Rsa15 	
<ul style="list-style-type: none"> ■ Encrypt Signature <ul style="list-style-type: none"> ■ Yes ■ No 	<ul style="list-style-type: none"> ■ TripleDe ■ TripleDesRsa15 ■ TripleDesSha256 	
<ul style="list-style-type: none"> ■ Layout <ul style="list-style-type: none"> ■ Lax ■ LaxTsFirst ■ LaxTsLast ■ Strict 	<ul style="list-style-type: none"> ■ Require Security Token Template Parameters <ul style="list-style-type: none"> ■ Key ■ Value 	
<ul style="list-style-type: none"> ■ Holder of Key Asymmetric Parameters 		

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> ■ Initiator Token Inclusion <ul style="list-style-type: none"> ■ Always ■ AlwaysToInitiator ■ AlwaysToRecipient ■ Never ■ Once ■ Recipient Token Inclusion <ul style="list-style-type: none"> ■ Always ■ AlwaysToInitiator ■ AlwaysToRecipient ■ Never ■ Once ■ Issuer Address ■ Metadata Reference Address ■ Key Size ■ Request Security Token Template Parameters <ul style="list-style-type: none"> ■ Key ■ Value 		

SAML Subject Confirmation:

Holder of Key (Symmetric)

- WS Trust Version
 - VERSION_05_02
 - VERSION_05_12
- Algorithm Suite
 - Basic128
 - Basic128Rsa15
 - Basic128Sha256

CentraSite	API Gateway	Notes
<ul style="list-style-type: none">■ Basic128Sha256Rsa15■ Basic192■ Basic192Rsa15■ Basic192Sha256■ Basic192Sha256Rsa15■ Basic256■ Basic256Rsa15■ Basic256Sha256■ Basic256Sha256Rsa15■ TripleDe■ TripleDesRsa15■ TripleDesSha256		
■ Encrypt Signature		
<ul style="list-style-type: none">■ Yes■ No		
■ Layout		
<ul style="list-style-type: none">■ Lax■ LaxTsFirst■ LaxTsLast■ Strict		
■ Holder of Key Symmetric Parameters		
<ul style="list-style-type: none">■ Protection Token Inclusion		
<ul style="list-style-type: none"><ul style="list-style-type: none">■ Always■ AlwaysToInitiator■ AlwaysToRecipient■ Never■ Once		

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> ■ Issuer Address ■ Metadata Reference Address ■ Key Size ■ Request Security Token Template Parameters <ul style="list-style-type: none"> ■ Key ■ Value 	<p data-bbox="691 646 1084 716">SAML Subject Confirmation: Holder of Key - Symmetric</p> <ul style="list-style-type: none"> ■ WS-Trust Version <ul style="list-style-type: none"> ■ WS-Trust 1.0 ■ WS-Trust 1.3 ■ Encrypt Signature ■ Issuer Address ■ Metadata Reference Address ■ Algorithm Suite <ul style="list-style-type: none"> ■ Basic128 ■ Basic128Rsa15 ■ Basic128Sha256 ■ Basic128Sha256Rsa15 ■ Basic192 ■ Basic192Rsa15 ■ Basic192Sha256 ■ Basic192Sha256Rsa15 ■ Basic256 ■ Basic256Rsa15 ■ Basic256Sha256 ■ Basic256Sha256Rsa15 	

CentraSite	API Gateway	Notes
	<ul style="list-style-type: none"> ■ TripleDe ■ TripleDesRsa15 ■ TripleDesSha256 ■ Require Security Token Template Parameters ■ Key ■ Value 	

Validate SAML Audience URIs

You can secure SOAP APIs using Audience URIs. This policy action verifies whether any of the audience URI within a valid condition element in SAML assertion matches with any of the configured URIs .

The **Validate SAML Audience URIs** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the policy **Inbound Authentication - Message** under the **Identify & Access** stage in API Gateway. The policy definition is mapped into the section `validate SAML Audience URI`.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Allowed URIs:	Allowed URIs:	
<ul style="list-style-type: none"> ■ URI 	<ul style="list-style-type: none"> ■ URI 	
Match Criteria:	Match Criteria:	
<ul style="list-style-type: none"> ■ Allow Sublevels 	<ul style="list-style-type: none"> ■ Allow Sublevels 	
Match Criteria:	Match Criteria:	
<ul style="list-style-type: none"> ■ Exact Match 	<ul style="list-style-type: none"> ■ Exact Match 	

Validate Schema

You can secure SOAP APIs using schema validation. This policy action validates XML request messages, response messages, or both, against the XML schema referenced in the API's WSDL.

The **Validate Schema** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the policy **Validate Schema** under the **Request Processing** or **Request Processing** stage in API Gateway as follows:

- If this policy action is defined to validate SOAP request messages with the parameter `Validate SOAP Message` set to `Request` in CentraSite, then the policy **Validate Schema** under the **Request Processing** stage is included to the API's policy list in API Gateway.
- If this policy action is defined to validate SOAP response messages with the parameter `Validate SOAP Message` set to `Response` in CentraSite, then the policy **Validate Schema** under the **Response Processing** stage is included to the API's policy list in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Validate SOAP Message: <ul style="list-style-type: none"> ■ Request 	Feature Name: <ul style="list-style-type: none"> ■ Feature Value: <ul style="list-style-type: none"> ■ True ■ False 	When the parameter <code>Validate SOAP Message</code> is set to <code>Request</code> in CentraSite, the policy Validate Schema under Request Processing stage is included to the API's list of policies in API Gateway.
Validate SOAP Message: <ul style="list-style-type: none"> ■ Response 	Feature Name: <ul style="list-style-type: none"> ■ Feature Value: <ul style="list-style-type: none"> ■ True ■ False 	When the parameter <code>Validate SOAP Message</code> is set to <code>Response</code> in CentraSite, the policy Validate Schema under Response Processing stage is included to the API's list of policies in API Gateway.

HTTP Basic Authentication

This policy action uses basic HTTP authentication credentials to authenticate applications.

The **HTTP Basic Authentication** policy in CentraSite is mapped into the policy **Outbound Authentication - Transport** under the **Routing** stage in API Gateway. The parameter `Authentication Scheme` is set to `Basic`.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Authenticate Using: Custom Credentials <ul style="list-style-type: none"> ■ Username ■ Password ■ Domain 	Authenticate using: Custom Credentials <ul style="list-style-type: none"> ■ Username ■ Password ■ Domain 	
Authenticate Using: Secure Alias	Authenticate scheme: Alias	

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> Alias Name 		
Authenticate Using:	Authenticate using:	
Use Existing Credentials	Incoming HTTP basic auth credentials	

NTLM Authentication

This policy action uses NTLM credentials to authenticate applications.

The **NTLM Authentication** policy in CentraSite is mapped into the policy **Outbound Authentication - Transport** under the **Routing** stage in API Gateway. The parameter `Authentication Scheme` is set to NTLM.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Authenticate Using:	Authenticate using:	
Custom Credentials	Custom Credentials	
<ul style="list-style-type: none"> Username Password Domain 	<ul style="list-style-type: none"> Username Password Domain 	
Authenticate Using:	Authenticate scheme:	
Secure Alias	Alias	
<ul style="list-style-type: none"> Alias Name 		
Authenticate Using:	Authenticate using:	
Transparent	Transparent	
Authenticate Using:	Authenticate using:	
Use Existing Credentials	Incoming HTTP basic auth credentials	

OAuth2 Authentication

This policy action uses basic OAuth 2.0 token credentials to authenticate applications.

The **OAuth2 Authentication** policy in CentraSite is mapped into the policy **Outbound Authentication - Transport** under the **Routing** stage in API Gateway. The parameter `Authentication Scheme` is set to `OAuth2`.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Authenticate Using:	Authenticate using:	
Custom Token	Custom Credentials	
■ Custom Credential	■ Custom credentials	
■ OAuth2 Token	■ OAuth2 token	
Authenticate Using:	Authenticate scheme:	
Secure Alias	Alias	
■ Alias Name		
Authenticate Using:	Authenticate using:	
Use Existing Token	Incoming OAuth token	

Kerberos Authentication

This policy action uses Kerberos token credentials to authenticate applications.

You can select the level at which the Kerberos authentication should be enforced for APIs:

- Message level
- Transport layer

The **Kerberos Authentication** policy action in CentraSite is mapped into the following policies in API Gateway. In both cases, the parameter `Authentication Scheme` is set to `Kerberos`.

- If this policy action is set to authenticate SOAP APIs at the message level using the parameter `Enforcement Point: Message Level` in CentraSite, then the policy **Outbound Authentication - Message** under the **Routing** stage is included to the API's policy list in API Gateway.
- If this policy action is set to authenticate SOAP and REST APIs at the transport layer using the parameter `Enforcement Point: Transport Level` in CentraSite, then the policy **Outbound Authentication - Transport** under the **Routing** stage is included to the API's policy list in API Gateway.

Enforcement Point: Message Level Security

The following table summarizes the mapping of the policy enforcement at the message level:

CentraSite	API Gateway	Notes
Authenticate Using: Custom Credentials <ul style="list-style-type: none"> ■ Client Principal ■ Client Password ■ Service Principal ■ Service Principal Form <ul style="list-style-type: none"> ■ hostbased ■ username 	Authenticate using: Custom Credentials <ul style="list-style-type: none"> ■ Client principal ■ Client password ■ Service principal ■ Service principal nameform <ul style="list-style-type: none"> ■ Username ■ Hostbased 	
Authenticate Using: Delegating Incoming Credentials <ul style="list-style-type: none"> ■ Client Principal ■ Client Password ■ Service Principal ■ Service Principal Form <ul style="list-style-type: none"> ■ hostbased ■ username 	Authenticate using: Delegating incoming credentials <ul style="list-style-type: none"> ■ Client principal ■ Client password ■ Service principal ■ Service principal nameform <ul style="list-style-type: none"> ■ Username ■ Hostbased 	
Authenticate Using: Secure Alias <ul style="list-style-type: none"> ■ Alias Name 	Authenticate scheme: Alias	
Authenticate Using: Use Existing Credentials <ul style="list-style-type: none"> ■ Service Principal ■ Service Principal Form <ul style="list-style-type: none"> ■ hostbased ■ username 	Authenticate using: Incoming HTTP basic auth credentials <ul style="list-style-type: none"> ■ Service principal ■ Service principal nameform <ul style="list-style-type: none"> ■ Username ■ Hostbased 	

Enforcement Point: Transport Layer Security

The following table summarizes the mapping of the policy enforcement at the transport layer:

CentraSite	API Gateway	Notes
Authenticate Using: Custom Credentials <ul style="list-style-type: none"> ■ Client Principal ■ Client Password ■ Service Principal ■ Service Principal Form <ul style="list-style-type: none"> ■ hostbased ■ username 	Authenticate using: Custom Credentials <ul style="list-style-type: none"> ■ Client principal ■ Client password ■ Service principal ■ Service principal nameform <ul style="list-style-type: none"> ■ Username ■ Hostbased 	
Authenticate Using: Delegating Incoming Credentials <ul style="list-style-type: none"> ■ Client Principal ■ Client Password ■ Service Principal ■ Service Principal Form <ul style="list-style-type: none"> ■ hostbased ■ username 	Authenticate using: Delegating incoming credentials <ul style="list-style-type: none"> ■ Client principal ■ Client password ■ Service principal ■ Service principal nameform <ul style="list-style-type: none"> ■ Username ■ Hostbased 	
Authenticate Using: Secure Alias <ul style="list-style-type: none"> ■ Alias Name 	Authenticate scheme: Alias	
Authenticate Using: Use Existing Credentials <ul style="list-style-type: none"> ■ Service Principal ■ Service Principal Form <ul style="list-style-type: none"> ■ hostbased 	Authenticate using: Incoming HTTP basic auth credentials <ul style="list-style-type: none"> ■ Service principal ■ Service principal nameform 	

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> ■ username 	<ul style="list-style-type: none"> ■ Username ■ Hostbased 	

SAML Authentication

This policy action uses SAML token credentials to authenticate applications.

The **SAML Authentication** policy in CentraSite is mapped into the policy **Outbound Authentication - Message** under the **Routing** stage in API Gateway. The parameter **Authentication Scheme** is set to SAML.

The SAML Issuer Communication details in CentraSite are mapped into the SAML issuer configuration page (go to **<Username> > Administration > Security > SAML issuer**) in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Not applicable	Authenticate scheme: Alias	
Signing Alias	Signing configurations <ul style="list-style-type: none"> ■ Keystore alias ■ Key alias 	<p>This parameter refers to the authentication configurations that should be used for signing outbound messages.</p> <p>The parameter Signing Alias of CentraSite is mapped in to the parameter Key alias in API Gateway.</p> <p>The parameter Keystore alias is mapped in to the parameter Keystore alias as defined in the Keystore/Truststore configuration page (go to Username > Administration > General > Security) in API Gateway.</p>
Encryption Alias	Encryption configurations <ul style="list-style-type: none"> ■ Truststore alias ■ Certificate alias 	<p>This parameter refers to the authentication configurations that should be used for encrypting outbound messages.</p>

CentraSite	API Gateway	Notes
		<p>The parameter Encryption Alias of CentraSite is mapped in to the parameter Certificate alias in API Gateway.</p> <p>The parameter Truststore alias is mapped in to the parameter Truststore alias as defined in the Keystore/Truststore configuration page (go to Username > Administration > General > Security) in API Gateway.</p>
<p>Issuer Communication</p> <ul style="list-style-type: none"> ■ Action <ul style="list-style-type: none"> ■ Act as Delegation ■ Normal Client 	<p>SAML Issuer Configuration</p> <ul style="list-style-type: none"> ■ Name ■ Act as Delegation ■ Normal Client 	
<p>Issuer Communication</p> <ul style="list-style-type: none"> ■ Communicate Using: WSS Username (Message) <ul style="list-style-type: none"> ■ WSS Username Configuration <ul style="list-style-type: none"> ■ Username ■ Password 	<p>SAML Issuer Configuration</p> <ul style="list-style-type: none"> ■ Communicate using: WSS Username ■ Authenticate using: Custom credentials <ul style="list-style-type: none"> ■ Username ■ Password 	
<p>Issuer Communication</p> <ul style="list-style-type: none"> ■ Communicate Using: Kerberos Over Transport (Message) <ul style="list-style-type: none"> ■ Authentication Mode: Custom Credentials <ul style="list-style-type: none"> ■ Client Principal ■ Client Password ■ Service Principal ■ Service Principal Form 	<p>SAML Issuer Configuration</p> <ul style="list-style-type: none"> ■ Communicate using: Kerberos ■ Authenticate using: Custom Credentials <ul style="list-style-type: none"> ■ Client Principal ■ Client Password ■ Service Principal ■ Service Principal Form 	

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> ■ Host Based ■ Username ■ Authentication Mode: Delegate Incoming Token ■ Client Principal ■ Client Password ■ Service Principal ■ Service Principal Form <ul style="list-style-type: none"> ■ Host Based ■ Username 	<ul style="list-style-type: none"> ■ Username ■ Hostbased ■ Authenticate using: Delegate incoming credentials ■ Client Principal ■ Client Password ■ Service Principal ■ Service Principal Form ■ Username ■ Hostbased ■ Authenticate using: Incoming HTTP basic auth credentials ■ Service Principal ■ Service Principal Form ■ Username ■ Hostbased 	
Endpoint	Endpoint URI	
Applies to	Applies to	
SAML Version	SAML Version	
<ul style="list-style-type: none"> ■ SAML 1.1 ■ SAML 2.0 	<ul style="list-style-type: none"> ■ SAML 1.1 ■ SAML 2.0 	
WS-Trust Version:	WS-Trust Version:	
<ul style="list-style-type: none"> ■ VERSION_05_02 ■ VERSION_05_12 	<ul style="list-style-type: none"> ■ WS-Trust 1.0 ■ WS-Trust 1.3 	
Signing Alias	Signing configurations:	This parameter refers to the authentication configurations
	<ul style="list-style-type: none"> ■ Keystore alias 	

CentraSite	API Gateway	Notes
	<ul style="list-style-type: none"> ■ Key alias 	<p>that should be used for signing outbound messages.</p> <p>The parameter Signing Alias of CentraSite is mapped in to the parameter Key alias in API Gateway.</p> <p>The parameter Keystore alias is mapped in to the parameter Keystore alias as defined in the Keystore/Truststore configuration page (go to Username > Administration > General > Security) in API Gateway.</p>
<p>Encryption Alias</p>	<p>Encryption configurations:</p> <ul style="list-style-type: none"> ■ Truststore alias ■ Certificate alias 	<p>This parameter refers to the authentication configurations that should be used for encrypting outbound messages.</p> <p>The parameter Encryption Alias of CentraSite is mapped in to the parameter Certificate alias in API Gateway.</p> <p>The parameter Truststore alias is mapped in to the parameter Truststore alias as defined in the Keystore/Truststore configuration page (go to Username > Administration > General > Security) in API Gateway.</p>
<p>Extended Parameters</p> <ul style="list-style-type: none"> ■ Key Size ■ Key Type ■ Signature Algorithm ■ Encryption Algorithm ■ CanonicalizationAlgorithm ■ ComputedKeyAlgorithm 	<p>Request security token template parameters:</p> <ul style="list-style-type: none"> ■ Key ■ Value 	

CentraSite	API Gateway	Notes
■ Encryption		
■ ProofEncryption		
■ KeyWrapAlgorithm		
■ SignWith		
■ EncryptWith		

Set Media Type (Request Handling)

This policy action specifies the content type for a REST request.

The **Set Media Type** policy action under the **Request Handling** accordion in CentraSite mapped into the policy **Set Media Type** under the **Transport** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Media Type:	Default Content-Type:	Though this policy is listed in the Policy catalog of a SOAP API, it is applicable only for a REST enabled SOAP API in API Gateway.

Set Media Type (Response Processing)

This policy action specifies the content type for a REST response.

The **Set Media Type** policy action under the **Response Processing** accordion in CentraSite mapped into the policy **Set Media Type** under the **Transport** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Media Type:	Default Accept Header:	Though this policy is listed in the Policy catalog of a SOAP API, it is applicable only for a REST enabled SOAP API in API Gateway.

Request Transformation

This policy action specifies the XSLT transformation file to transform request messages from applications into a format required by the native API in the SOAP request before it is submitted to the native API.

The **Request Transformation** policy action under the **Request Handling** accordion is mapped into the policy **XSLT Transformation** under the **Request Processing** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Select Type: Transformation Alias <ul style="list-style-type: none"> ■ Alias Name 	XSLT Transformation Alias	
Select Type: Transformation File <ul style="list-style-type: none"> ■ Transformation File 	XSLT Document <ul style="list-style-type: none"> ■ XSLT File ■ XSLT Features <ul style="list-style-type: none"> ■ Feature Name ■ Feature Value 	<p>Regardless of a XSLT file name, for example, PreProcessingXSLT.xsl defined in CentraSite, the XSLT file name is transformed into a default name, RequestProcessingXSLT_<X>.xsl in API Gateway. Here, X is an integer, which denotes the number of XSLT files that are mapped from CentraSite to API Gateway.</p> <p>Let us consider an API contains two Request Transformation polices, each defined with an unique XSLT file, PreProcessingXSLT1.xsl and PreProcessingXSLT2.xsl in CentraSite. When this API is published from CentraSite to API Gateway, the two Request Transformation polices are mapped into a single Request Transformation policy in API Gateway. The XSLT file names are transformed as RequestProcessingXSLT_1.xsl and RequestProcessingXSLT_2.xsl in API Gateway.</p>

Response Transformation

This policy action specifies the XSLT transformation file to transform response messages from native APIs into a format required by the application.

The **Response Transformation** policy action under the **Response Processing** accordion in CentraSite is mapped into the policy **XSLT Transformation** under the **Response Processing** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Select Type: Transformation Alias <ul style="list-style-type: none"> Alias Name 	XSLT Transformation Alias	
Select Type: Transformation File <ul style="list-style-type: none"> Transformation File 	XSLT Document <ul style="list-style-type: none"> XSLT File XSLT Features <ul style="list-style-type: none"> Feature Name Feature Value 	<p>Regardless of a XSLT file name, for example, PostProcessingXSLT.xsl defined in CentraSite, the XSLT file name is transformed into a default name, ResponseProcessingXSLT_<X>.xsl in API Gateway. Here, X is an integer, which denotes the number of XSLT files that are mapped from CentraSite to API Gateway.</p> <p>Let us consider an API contains two Response Transformation polices, each defined with an unique XSLT file, PostProcessingXSLT1.xsl and PostProcessingXSLT2.xsl in CentraSite. When this API is published from CentraSite to API Gateway, the two Response Transformation polices are mapped into a single Response Transformation policy in API Gateway. The XSLT file names are transformed as ResponseProcessingXSLT_1.xsl and ResponseProcessingXSLT_2.xsl in API Gateway.</p>

Invoke webMethods Integration Server (Request Handling)

This policy action pre-processes the request messages and transforms the message into a format required by the native API, before API Gateway sends the requests to native APIs .

The **Invoke webMethods Integration Server** policy action under the **Response Processing** accordion in CentraSite is mapped into the policy **Invoke webMethods IS** under the **Request Processing** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Select Type: webMethods IS Service	webMethods IS Service	

CentraSite	API Gateway	Notes
■ webMethods IS Service		
Select Type: webMethods IS Service Alias	webMethods IS Service	
■ webMethods IS Service Alias		

Invoke webMethods Integration Server (Response Processing)

This policy action pre-processes the native API's response messages into a format required by the application, before API Gateway returns the responses to the application .

The **Invoke webMethods Integration Server** policy action under the **Response Processing** accordion in CentraSite is mapped into the policy **Invoke webMethods IS** under the **Response Processing** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Select Type: webMethods IS Service	webMethods IS Service	
■ webMethods IS Service		
Select Type: webMethods IS Service Alias	webMethods IS Service	
■ webMethods IS Service Alias		

Conditional Error Processing

This policy action returns the custom error message (and the native provider's service fault content) to the application when the native provider returns the service fault.

The **Conditional Error Processing** policy action under the **Error Handling** accordion in CentraSite is mapped into the policy **Conditional Error Processing** under the **Error Handling** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Error Conditions	Error Conditions	
Type: HTTP Header	Type: Header Error Criteria	

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> Header Name Header Value 	<ul style="list-style-type: none"> Header Name Header Value 	
<p>Error Conditions</p> <p>Type: Status Code</p> <ul style="list-style-type: none"> Code 	<p>Error Conditions</p> <p>Type: Status Code Error Criteria</p> <ul style="list-style-type: none"> Code 	
<p>Error Conditions</p> <p>Type: XPath Expression</p> <ul style="list-style-type: none"> XPath Expression Namespaces <ul style="list-style-type: none"> Prefix URI Value 	<p>Error Conditions</p> <p>Type: XPath Expression</p> <ul style="list-style-type: none"> XPath Expression Namespace <ul style="list-style-type: none"> Namespace Prefix Namespace URI Value 	
<p>Pre-Processing</p> <p>Type: ESB</p> <ul style="list-style-type: none"> webMethods IS Service 	<p>Pre-Processing</p> <p>Type: Invoke webMethods Integration Server Service</p> <ul style="list-style-type: none"> webMethods IS Service 	
<p>Pre-Processing</p> <p>Type: XSLT</p> <ul style="list-style-type: none"> Transformation File 	<p>Pre-Processing</p> <p>Type: XSLT Transformation</p> <p>XSLT Document</p> <ul style="list-style-type: none"> XSLT File XSLT Features <ul style="list-style-type: none"> Feature Name Feature Value 	
<p>Failure Message</p> <p>Content Type</p> <ul style="list-style-type: none"> json text 	<p>Failure Message</p> <p>Failure Messages</p> <ul style="list-style-type: none"> json text 	

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> ■ xml Error Template Use as default (check box)	<ul style="list-style-type: none"> ■ xml Error Template Use as default (check box)	
Custom Error Variable	Custom Error Variable	
Payload Type	Payload Type	
<ul style="list-style-type: none"> ■ Request ■ Response 	<ul style="list-style-type: none"> ■ Request ■ Response 	
Name	Name	
XPath Expression	XPath Expression	
Namespaces	Namespaces	
<ul style="list-style-type: none"> ■ Prefix ■ URI 	<ul style="list-style-type: none"> ■ Namespace Prefix ■ Namespace URI 	
Post-processing	Pre-Processing	
Type: XSLT	Type: XSLT Transformation	
<ul style="list-style-type: none"> ■ Transformation File 	XSLT Document <ul style="list-style-type: none"> ■ XSLT File ■ XSLT Features <ul style="list-style-type: none"> ■ Feature Name ■ Feature Value 	
Send Native Provider Fault Message (check box)	Send Native Provider Fault Message (check box)	

Require HTTP/HTTPS

This policy action specifies the HTTP and HTTPS protocol and the SOAP format for an API to accept and process the requests.

The **Require HTTP / HTTPS** policy action under the **Request Handling > Protocol** accordion in CentraSite is mapped into the policy **HTTP / HTTPS** under the **Transport** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Protocol <ul style="list-style-type: none"> ■ HTTP ■ HTTPS 	Protocol <ul style="list-style-type: none"> ■ HTTP ■ HTTPS 	When the policy Require SSL is included to the API's policy list in CentraSite, the parameter HTTPS is selected in API Gateway.
SOAP Version <ul style="list-style-type: none"> ■ SOAP 1.1 ■ SOAP 1.2 	SOAP version <ul style="list-style-type: none"> ■ SOAP 1.1 ■ SOAP 1.2 	

Require JMS

This policy action specifies the JMS protocol and the SOAP format for an API to accept and process the requests.

The **Require JMS** policy action under the **Request Handling > Protocol** accordion in CentraSite is mapped into the policy **Require JMS** under the **Transport** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
JMS Trigger	JMS Provider Endpoint Alias	When the policy Require SSL is included to the API's policy list in CentraSite, the parameter HTTPS is selected in API Gateway.
SOAP Version <ul style="list-style-type: none"> ■ SOAP 1.1 ■ SOAP 1.2 	SOAP version <ul style="list-style-type: none"> ■ SOAP 1.1 ■ SOAP 1.2 	

Enable REST Support

When a SOAP API with this policy action is published from CentraSite to API Gateway, API Gateway Deployer Service looks for the `expose-as-rest` policy element in the `inSequence` of the Virtual Service Definition (VSD).

The following example shows the `inSequence` of a VSD holding the policy element `expose-as-rest`:

```
<inSequence>
<expose-as-rest soap-version="soap11"/>
  <http-config>
    <authentication mode="anonymous"/>
  </http-config>
  <send>
    <endpoint>
```

```

    <address isAlias="false" optimize="none" passSecurityHeaders="false"
      uri="https://test:8081/bayern/services/
        BayernService.BayernServiceHttpsEndpoint/">
      <connect-timeout>
        <duration>30</duration>
      </connect-timeout>
    </address>
  </endpoint>
</send>
</inSequence>

```

If the `inSequence` of the VSD holds the policy element `expose-as-rest`, then the API Gateway Deployer Service sets the `isRESTInvokeEnabled` flag to `true` for all operations of the SOAP API.

JMS Routing Rule

This policy action specifies the JMS provider queue to which the API Gateway submits the request, and the destination to which the API Gateway waits for the native SOAP API to return the response.

The **JMS Routing Rule** policy action under the **Policy Enforcement > JMS Routing** accordion in CentraSite is mapped into the policy **JMS Routing** under the **Routing** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Connection URL	Connection URL	
Destination Name	Reply to Destination	
Priority	Priority	
Timeout (ms)	Timeout (ms)	
Delivery Mode	Delivery Mode	
<ul style="list-style-type: none"> ■ Non Persistent ■ Persistent 	<ul style="list-style-type: none"> ■ Non Persistent ■ Persistent 	

Set Message Properties

This policy action specifies the JMS message properties that are to be sent to the native SOAP APIs.

The **Set Message Properties** policy action under the **Policy Enforcement > JMS Routing** accordion in CentraSite is mapped into the policy **JMS Properties** under the **Routing** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Property	JMS Property	

CentraSite	API Gateway	Notes
Name	JMS Property Key	
Property	JMS Property	
Value	JMS Property Value	

Set JMS Headers

This policy action specifies the JMS headers that are to be sent to the native SOAP APIs.

The **Set JMS Headers** policy action under the **Policy Enforcement > JMS Routing** accordion in CentraSite is mapped into the policy **JMS Properties** under the **Routing** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Header	JMS Property	
Name	JMS Property Key	
Header	JMS Property	
Value	JMS Property Value	

Log Invocation

This policy action enables logging of the incoming requests along with the request and response payloads to a specified destination.

The **Log Invocation** policy action under the **Policy Enforcement > Logging and Monitoring** accordion in CentraSite is mapped into the policy **Log Invocation** under the **Traffic Monitoring** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Payloads	<ul style="list-style-type: none"> ■ Store Request Payload 	
<ul style="list-style-type: none"> ■ Request 	<ul style="list-style-type: none"> ■ Store Response Payload 	
<ul style="list-style-type: none"> ■ Response 	<ul style="list-style-type: none"> ■ Compress Payload Data 	
Log Generation Frequency	Log Generation Frequency	
<ul style="list-style-type: none"> ■ Always 	<ul style="list-style-type: none"> ■ Always 	
	<ul style="list-style-type: none"> ■ On Failure 	

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> ■ On Failure ■ On Success 	<ul style="list-style-type: none"> ■ On Success 	
<p>Send Log Data</p> <ul style="list-style-type: none"> ■ API Portal ■ Audit Log ■ CentraSite ■ EDA / Database ■ ElasticSearch ■ E-mail ■ Local Log ■ SNMP 	<p>Send Log Data</p> <ul style="list-style-type: none"> ■ API Gateway ■ API Portal ■ Audit Log ■ CentraSite ■ Digital Events ■ Elasticsearch ■ E-mail ■ JDBC ■ Local Log: Log Level <ul style="list-style-type: none"> ■ Error ■ Info ■ Warn ■ SNMP 	<ul style="list-style-type: none"> ■ Beginning with version 10.1, API Gateway supports the following destinations: <ul style="list-style-type: none"> ■ Audit Log ■ CentraSite ■ Digital Events ■ Elasticsearch ■ Email ■ Local Log ■ SNMP ■ The destination name for database is changed from EDA / Database in CentraSite to JDBC in API Gateway. Therefore, when the parameter Send Log Data is set to EDA / Database in CentraSite, the pool alias definition and the corresponding configurations are mapped into JDBC in API Gateway. ■ When the EDA / Database destination is selected in CentraSite, it also automatically selects the Digital Events destination in API Gateway.

Monitor Service Performance

This policy action monitors the run-time performance conditions for an API, and sends alerts to a specified destination when the performance conditions are violated. However, this policy action monitors run-time performance for all applications.

The **Monitor Service Performance** policy action under the **Policy Enforcement > Logging and Monitoring** accordion in CentraSite is mapped into the policy **Monitor Service Performance** under the **Traffic Monitoring** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Action Configuration: Name <ul style="list-style-type: none"> ■ Availability ■ Average Response Time ■ Fault Count ■ Maximum Response Time ■ Minimum Response Time ■ Successful Request Count ■ Total Request Count 	Action Configuration: Name <ul style="list-style-type: none"> ■ Availability ■ Average Response Time ■ Fault Count ■ Maximum Response Time ■ Minimum Response Time ■ Successful Count ■ Total Request Count 	
Action Configuration: Operator <ul style="list-style-type: none"> ■ Equal To ■ Greater Than ■ Less Than 	Action Configuration: Operator <ul style="list-style-type: none"> ■ Equal To ■ Greater Than ■ Less Than 	
Action Configuration: Value	Action Configuration: Value	
Alert Parameters: Alert Interval (minutes)	Alert Interval: Unit: <ul style="list-style-type: none"> ■ Minutes ■ Hours ■ Days ■ Weeks 	
Alert for Consumer Applications	Consumer Applications	Names of the Consumer Applications in CentraSite are transformed into Application IDs in API Gateway.
Alert Parameters: Alert Destination <ul style="list-style-type: none"> ■ API Portal 	Alert Destination <ul style="list-style-type: none"> ■ API Gateway ■ API Portal 	<ul style="list-style-type: none"> ■ Beginning with version 10.1, API Gateway supports the following destinations: <ul style="list-style-type: none"> ■ Audit Log

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> ■ CentraSite ■ EDA/Database ■ Elasticsearch ■ E-mail ■ Local Log: Log Level <ul style="list-style-type: none"> ■ Error ■ Info ■ Warn ■ SNMP 	<ul style="list-style-type: none"> ■ CentraSite ■ Digital Events ■ Elasticsearch ■ Email ■ JDBC ■ Local Log: Log Level <ul style="list-style-type: none"> ■ Error ■ Info ■ Warn ■ SNMP 	<ul style="list-style-type: none"> ■ CentraSite ■ Digital Events ■ Elasticsearch ■ Email ■ Local Log ■ SNMP ■ The destination name for database is changed from EDA / Database in CentraSite to JDBC in API Gateway. Therefore, when the parameter Send Log Data is set to EDA / Database in CentraSite, the pool alias definition and the corresponding configurations are mapped into JDBC in API Gateway. ■ When the EDA / Database destination is selected in CentraSite, it also automatically selects the Digital Events destination in API Gateway.

Alert Parameters: Alert **Alert Message**
Message

Monitor Service Level Agreement

This policy action monitors the user-specified set of run-time performance conditions for an API, and sends alerts to a specified destination when the performance conditions are violated. This policy action monitors run-time performance for one or more specified applications.

The **Monitor Service Level Agreement** policy action under the **Policy Enforcement > Logging and Monitoring** accordion in CentraSite is mapped into the policy **Monitor Service Level Agreement** under the **Traffic Monitoring** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Actions: Name <ul style="list-style-type: none"> ■ Availability 	Action Configuration: Name	The Soft Limit configuration of Throttling Traffic Optimization policy action under the Policy Enforcement

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> ■ Average Response Time ■ Fault Count ■ Maximum Response Time ■ Minimum Response Time ■ Successful Request Count ■ Total Request Count 	<ul style="list-style-type: none"> ■ Availability ■ Average Response Time ■ Fault Count ■ Maximum Response Time ■ Minimum Response Time ■ Successful Count ■ Total Request Count 	<p>> Traffic Management accordion in CentraSite is mapped into this policy in API Gateway.</p>
<p>Actions: Operator</p> <ul style="list-style-type: none"> ■ Equal To ■ Greater Than ■ Less Than 	<p>Action Configuration: Operator</p> <ul style="list-style-type: none"> ■ Equal To ■ Greater Than ■ Less Than 	
<p>Actions: Value</p>	<p>Action Configuration: Value</p>	
<p>Alert Parameters: Alert Interval (minutes)</p>	<p>Alert Interval Unit</p> <ul style="list-style-type: none"> ■ Minutes ■ Hours ■ Days ■ Weeks 	
<p>Alert for Consumer Applications</p>	<p>Consumer Applications</p>	<p>Names of the consumer applications in CentraSite are transformed into Application IDs in API Gateway.</p> <p>As a pre-requisite the appropriate consumer applications should be migrated from CentraSite to API Gateway.</p>
<p>Alert Parameters: Alert Frequency</p> <ul style="list-style-type: none"> ■ Every Time 	<p>Alert Frequency</p> <ul style="list-style-type: none"> ■ Every Time ■ Only Once 	

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> Only Once 		
Alert Parameters: Alert Destination <ul style="list-style-type: none"> API Portal CentraSite EDA/Database Elasticsearch E-mail Local Log: Log Level <ul style="list-style-type: none"> Error Info Warn SNMP 	Alert Destination <ul style="list-style-type: none"> API Gateway API Portal CentraSite Digital Events Elasticsearch Email JDBC Local Log: Log Level <ul style="list-style-type: none"> Error Info Warn SNMP 	<ul style="list-style-type: none"> Beginning with version 10.1, API Gateway supports the following destinations: <ul style="list-style-type: none"> Audit Log CentraSite Digital Events Elasticsearch Email Local Log SNMP The destination name for database is changed from EDA / Database in CentraSite to JDBC in API Gateway. Therefore, when the parameter Send Log Data is set to EDA / Database in CentraSite, the pool alias definition and the corresponding configurations are mapped into JDBC in API Gateway. When the EDA / Database destination is selected in CentraSite, it also automatically selects the Digital Events destination in API Gateway.

Alert Parameters: Alert Message	Alert Message

Straight Through Routing

This policy action routes requests directly to a native endpoint that you specify.

The **Straight Through Routing** policy action under the **Policy Enforcement > Routing** accordion in CentraSite is mapped into the policy **Straight Through Routing** under the **Routing** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Route To:	Endpoint URI:	
Endpoint Properties: SOAP Optimization Method <ul style="list-style-type: none"> ■ MTOM ■ None ■ SWA 	SOAP Optimization Method: <ul style="list-style-type: none"> ■ MTOM ■ None ■ SWA 	
Endpoint Properties: HTTP Connection Timeout (seconds)	HTTP Connection Timeout (seconds)	
Endpoint Properties: Read Timeout (seconds)	Read Timeout (seconds)	
Endpoint Properties: SSL Configuration <ul style="list-style-type: none"> ■ Client Certificate Alias ■ Keystore Alias 	SSL Configuration <ul style="list-style-type: none"> ■ Keystore Alias ■ Key Alias 	
Endpoint Properties: WS-Security Header <ul style="list-style-type: none"> ■ Pass all security headers ■ Remove processed security headers 	Pass WS-Security Headers (check box)	<ul style="list-style-type: none"> ■ When the parameter <code>WS-Security Header</code> is set to <code>Pass</code> all security headers in CentraSite, the parameter <code>Pass WS-Security Headers</code> is selected in API Gateway. ■ When the parameter <code>WS-Security Header</code> is set to <code>Remove processed security headers</code> in CentraSite, the parameter <code>Pass WS-Security Headers</code> is NOT selected in API Gateway.

Content Based Routing

This policy action routes specific types of messages to specific endpoints based on specific values that appear in the request message, if the native API is hosted at two or more endpoints. The requests are routed according to the content-based routing rules you define.

The **Content Based Routing** policy action under the **Policy Enforcement > Routing** accordion in CentraSite is mapped into the policy **Content-based Routing** under the **Routing** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Route To:	Default Route To: Endpoint URI	
Endpoint Properties: SOAP Optimization Method	Default Route To : SOAP Optimization Method	
<ul style="list-style-type: none"> ■ MTOM ■ None ■ SWA 	<ul style="list-style-type: none"> ■ MTOM ■ None ■ SWA 	
Endpoint Properties: HTTP Connection Timeout (seconds)	Default Route To : HTTP Connection Timeout (seconds)	
Endpoint Properties: Read Timeout (seconds)	Default Route To : Read Timeout (seconds)	
Endpoint Properties: SSL Configuration	Default Route To: SSL Configuration	
<ul style="list-style-type: none"> ■ Client Certificate Alias ■ Keystore Alias 	<ul style="list-style-type: none"> ■ Keystore Alias ■ Key Alias 	
Endpoint Properties: WS-Security Header	Default Route To : Pass WS-Security Headers (check box)	
<ul style="list-style-type: none"> ■ Pass all security headers ■ Remove processed security headers 		<ul style="list-style-type: none"> ■ When the parameter WS-Security Header is set to Pass all security headers in CentraSite, the parameter Pass WS-Security Headers is selected in API Gateway. ■ When the parameter WS-Security Header is set to Remove processed security headers in CentraSite, the parameter Pass WS-Security Headers is NOT selected in API Gateway.
Not applicable	Rules : Name	<p>Regardless of a rule name, for example, Custom Rule, defined in CentraSite, the rule name is transformed into a default name, Rule <X> in API Gateway. Here, X is an integer, which denotes the number of rules that are mapped from CentraSite to API Gateway.</p> <p>Let us consider an API contains the Context Based Routing policy action</p>

CentraSite	API Gateway	Notes
		with two routing rules, each defined with an unique name, Custom Rule A and Custom Rule B in CentraSite. When this API is published from CentraSite to API Gateway, the two routing rules are mapped into the policy Context-based Routing in API Gateway. The rule names are transformed as Rule 1 and Rule 2 in API Gateway.
Routing Rule: XPath Expression	Rules: XPath Expression	
Routing Rule: Namespace	Rules: Namespace	
<ul style="list-style-type: none"> ■ Prefix ■ URI 	<ul style="list-style-type: none"> ■ Namespace Prefix ■ Namespace URI 	
Routing Rule: Route To	Rules: Route To	

Load Balancing and Failover Routing

This policy action routes the requests across multiple endpoints, if the native API is hosted at two or more endpoints.

The **Load Balancing and Failover Routing** policy action under the **Policy Enforcement > Routing** accordion in CentraSite is mapped into the policy **Load Balancer Routing** under the **Routing** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Route To:	Endpoint URI	
Endpoint Properties: SOAP Optimization Method	SOAP Optimization Method	
<ul style="list-style-type: none"> ■ MTOM ■ None ■ SWA 	<ul style="list-style-type: none"> ■ MTOM ■ None ■ SWA 	

CentraSite	API Gateway	Notes
Endpoint Properties: HTTP Connection Timeout (seconds)	HTTP Connection Timeout (seconds)	
Endpoint Properties: Read Timeout (seconds)	Read Timeout (seconds)	
Endpoint Properties: SSL Configuration	SSL Configuration	
<ul style="list-style-type: none"> ■ Client Certificate Alias ■ Keystore Alias 	<ul style="list-style-type: none"> ■ Keystore Alias ■ Key Alias 	
Endpoint Properties: WS-Security Header	Pass WS-Security Headers (check box)	<ul style="list-style-type: none"> ■ When the parameter WS-Security Header is set to Pass all security headers in CentraSite, the parameter Pass WS-Security Headers is selected in API Gateway. ■ When the parameter WS-Security Header is set to Remove processed security headers in CentraSite, the parameter Pass WS-Security Headers is NOT selected in API Gateway.
<ul style="list-style-type: none"> ■ Pass all security headers ■ Remove processed security headers 		
Timeout (seconds)	Suspend duration (seconds)	

Set Custom Headers

This policy action specifies the HTTP headers to authenticate application's requests before submitting to the native SOAP APIs.

The **Set Custom Headers** policy action under the **Policy Enforcement > Routing** accordion in CentraSite is mapped into the policy **Custom HTTP Header** under the **Routing** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Header	HTTP Header	
Name	HTTP Header Key	
Header	HTTP Header	

CentraSite	API Gateway	Notes
Value	HTTP Header Value	

Context Based Routing

This policy action routes specific types of messages to specific endpoints based on specific values that appear in the request message, if the native API is hosted at two or more endpoints. The requests are routed according to the context-based routing rules you define.

The **Context Based Routing** policy action under the **Policy Enforcement > Routing** accordion in CentraSite is mapped into the policy **Context-based Routing** under the **Routing** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Route To:	Default Route To: Endpoint URI	
Endpoint Properties: SOAP Optimization Method	Default Route To : SOAP Optimization Method	
<ul style="list-style-type: none"> ■ MTOM ■ None ■ SWA 	<ul style="list-style-type: none"> ■ MTOM ■ None ■ SWA 	
Endpoint Properties: HTTP Connection Timeout (seconds)	Default Route To : HTTP Connection Timeout (seconds)	
Endpoint Properties: Read Timeout (seconds)	Default Route To : Read Timeout (seconds)	
Endpoint Properties: SSL Configuration	Default Route To: SSL Configuration	
<ul style="list-style-type: none"> ■ Client Certificate Alias ■ Keystore Alias 	<ul style="list-style-type: none"> ■ Keystore Alias ■ Key Alias 	
Endpoint Properties: WS-Security Header	Default Route To : Pass WS-Security Headers (check box)	<ul style="list-style-type: none"> ■ When the parameter WS-Security Header is set to Pass all security headers in CentraSite, the parameter Pass WS-Security Headers is selected in API Gateway.
<ul style="list-style-type: none"> ■ Pass all security headers 		

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> ■ Remove processed security headers 		<ul style="list-style-type: none"> ■ When the parameter <code>WS-Security Header</code> is set to <code>Remove processed security headers</code> in CentraSite, the parameter <code>Pass WS-Security Headers</code> is NOT selected in API Gateway.
<p>Routing Rule : Name</p>	<p>Rules : Name</p>	<p>Regardless of a rule name, for example, Custom Rule, defined in CentraSite, the rule name is transformed into a default name, Rule <X> in API Gateway. Here, X is an integer, which denotes the number of rules that are mapped from CentraSite to API Gateway.</p> <p>Let us consider an API contains the Context Based Routing policy action with two routing rules, each defined with a unique name, Custom Rule A and Custom Rule B in CentraSite. When this API is published from CentraSite to API Gateway, the two routing rules are mapped into the policy Context-based Routing in API Gateway. The rule names are transformed as Rule 1 and Rule 2 in API Gateway.</p>
<p>Not applicable</p>	<p>Rules: Condition Operator</p> <ul style="list-style-type: none"> ■ Or ■ And 	
<p>Routing Rule: Condition</p> <ul style="list-style-type: none"> ■ Variable <ul style="list-style-type: none"> ■ Consumer ■ Custom Context Variable ■ Date ■ IP Address ■ IPv6 Address 	<p>Rules: Condition</p> <ul style="list-style-type: none"> ■ Variable <ul style="list-style-type: none"> ■ Consumer ■ Date ■ IPV4 ■ IPV6 ■ Predefined Context Variable 	<ul style="list-style-type: none"> ■ Beginning with version 10.1, API Gateway supports custom context variables in a routing rule that you create. The custom context variable name in CentraSite is automatically prefixed with <code>m_x:</code> in API Gateway. ■ For a list of the predefined context variables, see below.

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> ■ Predefined Context Variable ■ Time ■ Consumer 	<ul style="list-style-type: none"> ■ Custom Context Variable ■ Time ■ Variable Value 	
Routing Rule: Route To	Rules: Route To	

The Predefined Context Variables

Display Name in CentraSite	Display Name in API Gateway
Average Response Time	Not applicable
Fault Count	Not applicable
Minimum Response Time	Not applicable
Maximum Response Time	Not applicable
Success Count	Not applicable
Total Count	Not applicable
Client IP Address	Inbound IP
Consumer	Consumer
Inbound Content Type	Inbound Content Type
Inbound HTTP Method	Inbound HTTP Method
Inbound Protocol	Inbound Protocol
Mediator Hostname	Gateway Hostname
Mediator IP Address	Gateway IP
Mediator Target Name	Not applicable
Native Service Provider Error	Not applicable
Native Service Response Message	Not applicable
Outbound HTTP Method	Routing Method
User	User
Virtual Service Name	Not applicable
Virtual Service Operation	Operation Name

Display Name in CentraSite	Display Name in API Gateway
Virtual Service URI	Inbound Request URI
Not applicable	Inbound Accept

Dynamic Routing

This policy action routes the requests to dynamic endpoints based on specific criteria that you specify.

The **Dynamic Routing** policy action under the **Policy Enforcement > Routing** accordion in CentraSite is mapped into the policy **Dynamic Routing** under the **Routing** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Route using:	Rule: Route Using	
<ul style="list-style-type: none"> ■ Context Variable ■ Header <ul style="list-style-type: none"> ■ Header Name 	<ul style="list-style-type: none"> ■ Context ■ Header <ul style="list-style-type: none"> ■ Header Name 	
Route Through	Default Route To: Endpoint URI	
Default Route-To	Route To	
Endpoint Properties: SOAP Optimization Method	Default Route To: SOAP Optimization Method	
<ul style="list-style-type: none"> ■ MTOM ■ None ■ SWA 	<ul style="list-style-type: none"> ■ MTOM ■ None ■ SWA 	
Endpoint Properties: HTTP Connection	Default Route To: HTTP Connection	
Timeout (seconds)	Timeout (seconds)	
Endpoint Properties: Read Timeout (seconds)	Default Route To: Read Timeout (seconds)	
Endpoint Properties: SSL Configuration	Default Route To: SSL Configuration	
<ul style="list-style-type: none"> ■ Client Certificate Alias 	<ul style="list-style-type: none"> ■ Keystore Alias 	

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> Keystore Alias 	<ul style="list-style-type: none"> Key Alias 	
Endpoint Properties: WS-Security Header <ul style="list-style-type: none"> Pass all security headers Remove processed security headers 	Default Route To: Pass WS-Security Headers (check box)	<ul style="list-style-type: none"> When the parameter WS-Security Header is set to Pass all security headers in CentraSite, the parameter Pass WS-Security Headers is selected in API Gateway. When the parameter WS-Security Header is set to Remove processed security headers in CentraSite, the parameter Pass WS-Security Headers is NOT selected in API Gateway.

Authorize User

This policy action authorizes applications against a list of users and a list of groups registered in CentraSite or API Gateway.

The **Authorize User** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the policy **Authorize User** under the **Identify & Access** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Perform authorization against: List of users <ul style="list-style-type: none"> Users 	List of Users	
Perform authorization against: List of groups <ul style="list-style-type: none"> Groups 	List of Groups	

Allow Anonymous Usage

This policy action specifies whether to allow all users to access the API without restriction.

The **Allow Anonymous Usage** policy action under the **Policy Enforcement > Security** accordion in CentraSite is mapped into the policy **Identify & Authorize Application** under the **Identify & Access** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Allow Anonymous Usage <ul style="list-style-type: none"> ■ True ■ False 	Allow anonymous	<ul style="list-style-type: none"> ■ When the parameter Allow Anonymous Usage is set to <code>True</code> in CentraSite, the parameter Allow anonymous is selected in API Gateway. ■ When the parameter Allow Anonymous Usage is set to <code>False</code> or if the policy action is <code>NOT</code> defined in CentraSite, the parameter Allow anonymous is <code>NOT</code> selected in API Gateway.

Throttling Traffic Optimization

This policy action limits the number of API invocations during a specified time interval, and sends alerts to a specified destination when the performance conditions are violated. This action avoids overloading the back-end APIs and their infrastructure, to limit specific applications in terms of resource usage, and so on.

The **Throttling Traffic Optimization** policy action under the **Policy Enforcement > Traffic Management** accordion in CentraSite is mapped into the policy **Throttling Traffic Optimization** under the **Traffic Monitoring** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Hard Limit: Rule Name <ul style="list-style-type: none"> ■ Total Request Count 	Limit Configuration: Rule Name <ul style="list-style-type: none"> ■ Total Request Count 	
Hard Limit: Operator <ul style="list-style-type: none"> ■ Greater Than 	Limit Configuration: Operator <ul style="list-style-type: none"> ■ Greater Than 	
Hard Limit: Value	Limit Configuration: Value	
Hard Limit: Alert Message for Hard Limit	Alert Message	
Soft Limit : Rule Name <ul style="list-style-type: none"> ■ Total Request Count 	Action Configuration: Name <ul style="list-style-type: none"> ■ Total Request Count 	Depending on the value specified for the parameter Consumer Applications , the Soft Limit configuration of this policy action in CentraSite is mapped

CentraSite	API Gateway	Notes
Soft Limit : Operator <ul style="list-style-type: none"> Greater Than 	Action Configuration: Operator <ul style="list-style-type: none"> Greater Than 	into one of the following policies under the Traffic Monitoring stage in API Gateway:
Soft Limit : Value	Action Configuration: Value	<ul style="list-style-type: none"> If the value is set to a specific consumer, then this configuration is mapped into the policy Monitor Service Level Agreement. If the value is set to Any consumer application, then this configuration is mapped into the policy Monitor Service Performance.
Soft Limit : Alert Message for Hard Limit	Alert Message	
Consumer Applications	Consumer Applications	Names of the consumer applications in CentraSite are transformed into Application IDs in API Gateway.
Interval Value <ul style="list-style-type: none"> Minutes Hours Days Weeks 	Alert Interval Unit: <ul style="list-style-type: none"> Minutes Hours Days Weeks 	
Frequency <ul style="list-style-type: none"> Every Time Only Once 	Alert Frequency <ul style="list-style-type: none"> Every Time Only Once 	
Alert Destination <ul style="list-style-type: none"> API Portal CentraSite EDA/Database Elasticsearch E-mail Local Log: Log Level 	Alert Destination <ul style="list-style-type: none"> API Gateway API Portal CentraSite Digital Events Elasticsearch E-mail 	<ul style="list-style-type: none"> Beginning with version 10.1, API Gateway supports the following destinations: <ul style="list-style-type: none"> Audit Log CentraSite Digital Events Elasticsearch Email

CentraSite	API Gateway	Notes
<ul style="list-style-type: none"> ■ Error ■ Info ■ Warn ■ SNMP 	<ul style="list-style-type: none"> ■ JDBC ■ Local Log: Log Level <ul style="list-style-type: none"> ■ Error ■ Info ■ Warn ■ SNMP 	<ul style="list-style-type: none"> ■ Local Log ■ SNMP ■ The destination name for database is changed from EDA / Database in CentraSite to JDBC in API Gateway. Therefore, when the parameter Send Log Data is set to EDA / Database in CentraSite, the pool alias definition and the corresponding configurations are mapped into JDBC in API Gateway. ■ When the EDA / Database destination is selected in CentraSite, it also automatically selects the Digital Events destination in API Gateway.

Service Result Cache

This policy action enables caching of the results of SOAP and REST API invocations based on specific criteria that you specify.

The **Service Result Cache** policy action under the **Policy Enforcement > Traffic Management** accordion in CentraSite is mapped into the policy **Service Result Cache** under the **Traffic Monitoring** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Configure Caching based on: HTTP Header <ul style="list-style-type: none"> ■ Header Name ■ Add to Whitelist <ul style="list-style-type: none"> ■ Whitelist (Cache only for these values) 	Cache Criteria: HTTP Header <ul style="list-style-type: none"> ■ Cache responses only for these values 	
Configure Caching based on: Path	No cache criteria	If none of the cache criteria is specified, then API Gateway internally uses the Path criteria for caching the API response.

CentraSite	API Gateway	Notes
Configure Caching based on: XPath Expression	Cache Criteria: XPath Expression	
<ul style="list-style-type: none"> ■ Namespace <ul style="list-style-type: none"> ■ Prefix ■ URI ■ XPath Expression ■ Add to Whitelist <ul style="list-style-type: none"> ■ Whitelist (Cache only for these values) 	<ul style="list-style-type: none"> ■ Namespace <ul style="list-style-type: none"> ■ Prefix ■ URI ■ XPath Expression ■ Cache responses only for these values 	
Not applicable	Cache Criteria: Query Parameters <ul style="list-style-type: none"> ■ Parameter Name ■ Cache responses only for these values 	
Time to Live (e.g., 5d 4h 1m)	Time to Live (e.g., 5d 4h 1m)	
Maximum Response Payload Size (in KB, -1 = unlimited)	Maximum Response Payload Size (in KB)	

Identify Consumer (CentraSite Control)

A legacy policy action from version 8.2.2. This policy action:

- Identifies an application based on the kind of consumer identifier (IP address, HTTP authorization token, and so on.) you specify.
- Allows anonymous users to access the API.

The **Identify Consumer** action defined in a runtime policy in CentraSite Control and enforced on an API in CentraSite Business UI is mapped into the policy **Identify and Authorize Application** under the **Identify & Access** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Anonymous Usage Allowed <ul style="list-style-type: none"> ■ Yes ■ No 	Allow anonymous	<ul style="list-style-type: none"> ■ When the parameter Anonymous Usage Allowed is set to Yes in CentraSite, the parameter Allow anonymous is selected (set to True) in the policy Identify and Authorize Application. ■ When the parameter Anonymous Usage Allowed is set to No in CentraSite, the parameter Allow anonymous is NOT selected (set to False) in the policy Identify and Authorize Application.
Identify User Using <ul style="list-style-type: none"> ■ IP Address 	Identification Type <ul style="list-style-type: none"> ■ IP Address Range 	
Identify User Using <ul style="list-style-type: none"> ■ Token Derived from Host Name 	Identification Type <ul style="list-style-type: none"> ■ Hostname Address 	
Identify User Using <ul style="list-style-type: none"> ■ Token Derived from HTTP Header 	Identification Type <ul style="list-style-type: none"> ■ HTTP Basic Authentication 	
Identify User Using <ul style="list-style-type: none"> ■ Token Derived from WSS Header 	Identification Type <ul style="list-style-type: none"> ■ WS Security Username token 	
Identify User Using <ul style="list-style-type: none"> ■ Token Derived from Custom XPATH 	Identification Type <ul style="list-style-type: none"> ■ XPath expression 	
Identify User Using <ul style="list-style-type: none"> ■ Consumer Certificate 	Identification Type <ul style="list-style-type: none"> ■ WS Security X.509 Certificate 	
Identify User Using <ul style="list-style-type: none"> ■ Client Certificate for SSL Connectivity 	Identification Type <ul style="list-style-type: none"> ■ SSL Certificate 	

Note:

If this policy action is configured with **Authorize Against Registered Consumer** in CentraSite Control, the parameter **Application Lookup Condition** is set to `Registered` applications in the policy **Identify and Authorize Application** in API Gateway. If this policy action is NOT configured with **Authorize Against Registered Consumer** in CentraSite Control, then the parameter **Application Lookup Condition** is set to `Global` applications in the policy **Identify and Authorize Application** in API Gateway.

Authorize Against Registered Consumer (CentraSite Control)

A legacy policy action from version 8.2.2. This policy action authorizes requests against the Registered Applications list in API Gateway.

The **Authorize Against Registered Consumer** action defined in a runtime policy in CentraSite Control and enforced on an API in CentraSite Business UI is mapped into the policy **Identify and Authorize Application** under the **Identify & Access** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Not applicable	Application Lookup Condition: <ul style="list-style-type: none"> ■ Registered applications 	The parameter Application Lookup Condition is set to <code>Registered</code> applications in the policy Identify and Authorize Application .

Note:

This policy action is dependent on the **Identify Consumer** action in CentraSite Control.

Require HTTP Basic Authentication (CentraSite Control)

You can secure Web Service APIs using HTTP basic authentication. This policy action validates the client's credentials contained in the request's Authorization header against the list of users in the Integration Server on which API Gateway is running.

The **Require HTTP Basic Authentication** action defined in a runtime policy in CentraSite Control and enforced on an API in CentraSite Business UI is mapped into the policy **Inbound Authentication - Transport** under the **Identify & Access** stage in API Gateway.

The following table summarizes the mapping of this policy action:

CentraSite	API Gateway	Notes
Authentication Required	HTTP Basic Authentication	When the parameter Authentication Required is selected in CentraSite, the parameter HTTP Basic Authentication is selected (set to <code>True</code>) in the policy Inbound Authentication - Transport .

Consumer Application Mapping Details

This section describes how the consumer applications and their property values defined and published from CentraSite are mapped into API Gateway.

A consumer application is published from CentraSite to API Gateway in the following scenarios:

- When publishing Virtual Services, CentraSite sends a list of registered consumer applications along with the VSD of the service to API Gateway.

API Gateway, in turn, performs the following operations when a Virtual Service associated with the consuming applications is published:

- Creates a SOAP or REST API for the Virtual SOAP Service or Virtual REST Service.
- Create an applications for the consuming application.
- Registers an application to the API.
- When republishing Virtual Services, CentraSite sends the updated list of registered consumer applications along with the VSD of the service to API Gateway.

API Gateway, in turn, performs the following actions when a Virtual Service associated with the consuming applications is published:

- Updates the details of an existing application with the recent changes.
- Creates an application for the newly defined consuming application.
- Unregisters an application that has been recently removed from the API.
- When unpublishing Virtual Services, CentraSite also unpublishes the list of consumer applications that were registered to the service.

API Gateway, in turn, performs the following operations when a Virtual Service is unpublished:

- When synchronizing consumer applications, API Gateway performs the following actions:
 - Create a new consumer application and registers to the required API.
 - Updates the details of an existing consumer application and the application's registration status for an API.
 - Any consumer application provided by CentraSite but not associated to the Virtual Service is removed.
- Requesting API keys and OAuth2 tokens

Publishing of consumer applications from CentraSite to API Gateway is performed by invoking the API Gateway Deployer Service. The Deployer Service creates an application in API Gateway for each consumer application, API key, and OAuth2 token specified in the API publish request.

Application

The following table summarizes the mapping of an Application in CentraSite to an Application in API Gateway:

CentraSite	API Gateway	Notes
Application	Application	
Key	Application ID	When an application with an UUID, say, xxx, is published from CentraSite, any existing application with the same UUID xxx in API Gateway is overwritten.
Name	Name	
Description	Description	
Type	Type	
Version	Version	
Not applicable	API access key	
Not applicable	Client ID	
Not applicable	Client secret	
Not applicable	Client name	
Not applicable	Scopes <ul style="list-style-type: none"> ■ Name ■ Description 	
Not applicable	Token lifetime	
Not applicable	Token refresh limit	
Not applicable	Redirect URIs	

Application Identifiers

The following table summarizes the mapping of an Application's Identification in CentraSite to an Application's Identifiers in API Gateway:

CentraSite	API Gateway	Notes
Identification token	Other identifiers <ul style="list-style-type: none"> ■ Token 	Beginning with version 10.1, API Gateway offers the option Token by which incoming messages from a particular application can be recognized at run-time.
From IP-V4 Address	IP address range	
To IP-V4 Address	IP address range	
From IP-V6 Address	IP address range	
To IP-V6 Address	IP address range	
Partner Id	Partner identifier	Beginning with version 10.1, API Gateway offers the parameter Partner identifier to specify the third-party partner's identity.
Consumer Certificate	Client certificates	
Not applicable	Claims	
Not applicable	Other identifiers <ul style="list-style-type: none"> ■ Hostname ■ Token ■ Username ■ WS-Security username ■ XPath 	

Note:

When an application with an UUID, say, xxx, is published from CentraSite, any existing application with the same UUID xxx in API Gateway is overwritten.

API Key

The following table summarizes the mapping of an Application's API key:

CentraSite	API Gateway	Notes
API Key	Application	
Key	Application ID	

CentraSite	API Gateway	Notes
Name	Name	When an application with an UUID, say, xxx, is published from CentraSite, any existing application with the same UUID xxx in API Gateway is overwritten.
Not applicable	Description	
Not applicable	Created	
API Key String	API access key	
Not applicable	Client ID	
Not applicable	Client secret	

Note:

The identifiers of an API key in CentraSite are not mapped to API Gateway.

Note:

When an application with an UUID, say, xxx, is published from CentraSite, any existing application with the same UUID xxx in API Gateway is overwritten.

OAuth2 Token

The following table summarizes the mapping of an Application's OAuth2 token:

CentraSite	API Gateway	Notes
OAuth2 Token Client	Application	
Key	Application ID	
OAuth2 Client Name	Name	When an application with an UUID, say, xxx, is published from CentraSite, any existing application with the same UUID xxx in API Gateway is overwritten.
Not applicable	Created	
Not applicable	API access key	
OAuth2 Client Id	Client ID	
OAuth2 Client Secret	Client secret	

Note:

The identifiers of an API key in CentraSite are not mapped to API Gateway.

Note:

When an application with an UUID, say, xxx, is published from CentraSite, any existing application with the same UUID xxx in API Gateway is overwritten.

Modifications to Error Codes and Responses for Runtime Policies

API Gateway and Mediator attempt to return appropriate HTTP status codes for every request.

This section is to note the changes in the error codes and responses after migration to API Gateway from CentraSite and Mediator.

Scenario	Error Code and Response in CentraSite/Mediator	Error Code and Response in API Gateway
Policy Validate Schema failed to validate Schema	Error Code: 500 Error Response: N/A	Error Code: 400 Error Response: N/A
Policy fails to return default fault response	Error Code: N/A Error Response: Mediator encountered an error	Error Code: N/A Error Response: API Gateway encountered an error
Outbound HTTP Basic Authentication with Incoming Credentials fails to authenticate the client	Error Code: 500 Error Response: N/A	Error Code: 401 Error Response: N/A
Outbound OAuth2 Authentication with Incoming Token fails to authenticate the client	Error Code: 500 Error Response: N/A	Error Code: 401 Error Response: N/A
Require SSL	This policy is available under the Policy Enforcement > Security accordion in CentraSite.	This policy is not available in API Gateway.
All of the Evaluate policy actions (Evaluate HTTP Basic Authentication, Evaluate Hostname, Evaluate IP Address, and so on) fails to identify applications	Error Code: 500 Error Response: Consumer could not be identified	Error Code: 403 Error Response: Unable to identify the application for the request.

Scenario	Error Code and Response in CentraSite/Mediator	Error Code and Response in API Gateway
Evaluate HTTP Basic Authentication fails to authenticate applications	Error Code: 500 Error Response: Consumer could not be identified	Error Code: 401 Error Response: The request cannot be authenticated.
Evaluate HTTP Basic Authentication with Authenticate User set to Yes, and Identify Consumer set to Do not Identify fails to authenticate applications, when the API request is sent without an Authorization header	Error Code: N/A Error Response: Incoming request does not contain the Authorization header.	Error Code: N/A Error Response: API Gateway is unable to process incoming request.
Outbound NTLM Authentication with Existing Credentials fails to authenticate applications, when the API request is sent without an Authorization header	Error Code: N/A Error Response: Incoming request does not contain the Authorization header.	Error Code: 401 Error Response: API Gateway outbound client encountered Native service provider error.
Inbound REST requests against an unknown resource path	Mediator allows such requests.	Error Code: 404 Error Response: API Gateway rejects the REST request against an unknown resource path.
CentraSite and API Gateway support a different API Key header.	x-CentraSite-APIKey	x-Gateway-APIKey