

CentraSite

Introducing CentraSite

Version 9.5 SP1

November 2013

This document applies to CentraSite Version 9.5 SP1.

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

Copyright © 2005-2013 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 <http://documentation.softwareag.com/legal/>.

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 <http://documentation.softwareag.com/legal/> 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 and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

Document ID: IINM-IG-INTRO-95SP1-20140410

Table of Contents

Introducing CentraSite	v
1 CentraSite's Role in Today's SOA	1
Today's SOA Challenges	2
Where CentraSite Fits in Today's SOA	3
Use of CentraSite by other Software AG Products	6
2 CentraSite Editions	9
CentraSite Editions	10
The CentraSite Feature Matrix	10
3 CentraSite Features	13
The Registry	14
The Catalog	14
The Repository	15
Design/Change-Time Policies	16
Run-Time Policies	16
Virtual Services	17
SOALink	17
Lifecycle Management	18
Reporting	18
Impact Analysis	19
Security and Auditing	19
Role-Based Access	19
Federation with Other Registries	20
GUI Access to CentraSite	20
API Access to CentraSite	21
4 Topology And Architecture	23
Basic Topology	24
CentraSite Architecture	25
5 The Information Model	27
Overview	28
Objects in the CentraSite Information Model	28
Relationships Between Objects in the Registry	30

Introducing CentraSite

This document provides an overview of CentraSite concepts and architecture.

For information about new and updated features in the latest product release, see the *Release Notes* document of the webMethods Product Suite, available under the Software AG documentation web site at <http://documentation.softwareag.com/>.

The description of CentraSite's concepts and architecture is organized under the following sections:

CentraSite's Role in Today's SOA	Describes how CentraSite solves many of today's SOA challenges and explains where CentraSite fits in today's SOA environment.
CentraSite Editions	Describes the editions of CentraSite that are available from Software AG and lists the specific features in each.
CentraSite Features	Briefly describes CentraSite's major features.
Topology And Architecture	Outlines the major components and sub-components within CentraSite.
The Information Model	Introduces the major objects in the information model and describes how they are related.

1 **CentraSite's Role in Today's SOA**

■ Today's SOA Challenges	2
■ Where CentraSite Fits in Today's SOA	3
■ Use of CentraSite by other Software AG Products	6

This chapter covers the following topics:

Today's SOA Challenges

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 via 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 (e.g., 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.

Where CentraSite Fits in Today's SOA

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

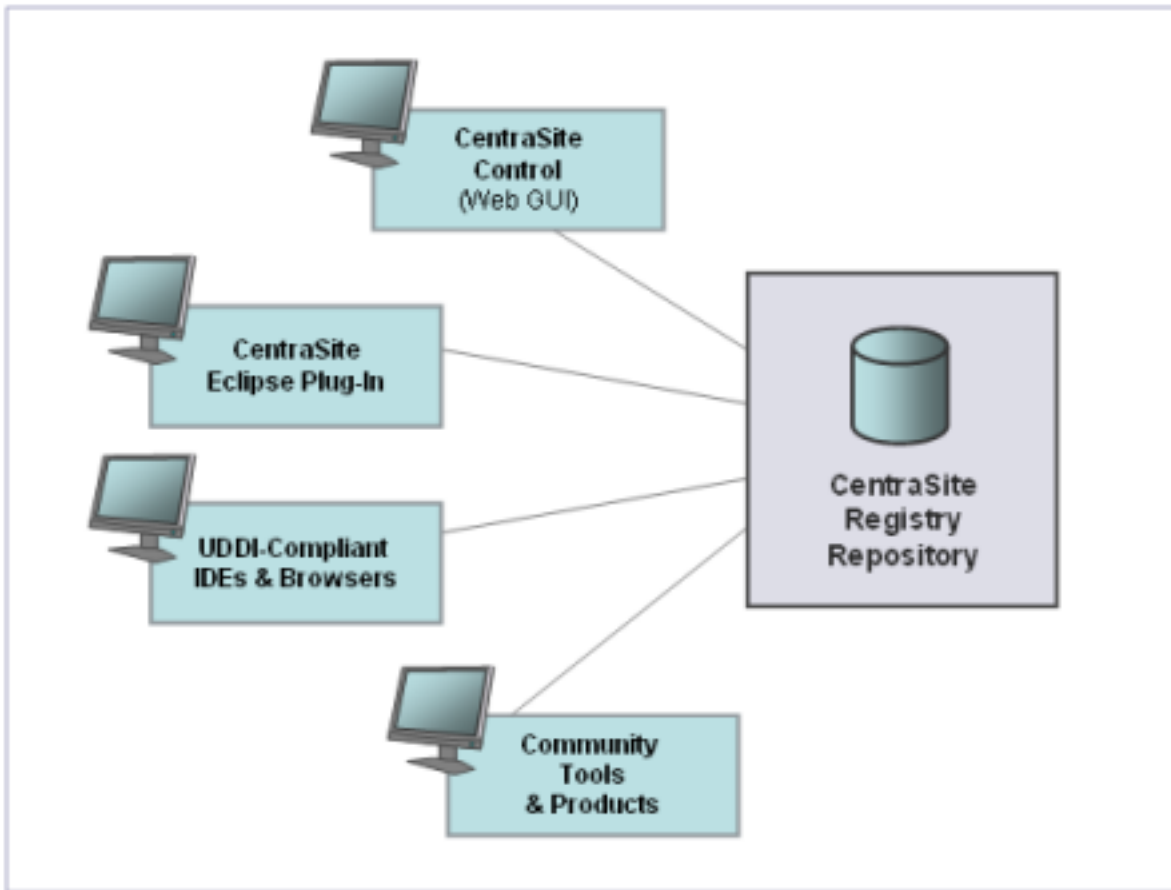
CentraSite supports the development of SOA-based applications by enabling developers, architects and business analysts to:

- 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 dependants.

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.
- *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



Asset Management and Governance

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/or 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.

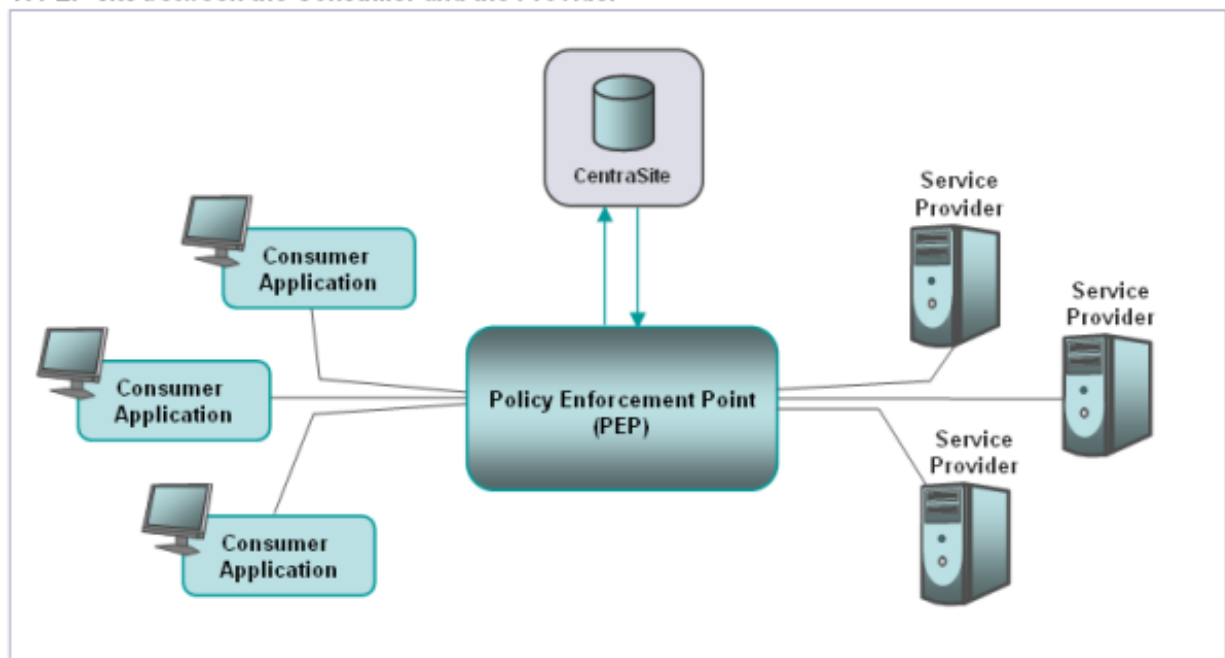
Administrators and architects use CentraSite Control, the browser-based user interface provided with CentraSite, to perform these types of governance-related tasks. Systems and tools from the CentraSite Community can also provide this kind of functionality. Some basic administrative tasks can also be 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/decryption), auditing/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 or XML 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/or 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 policy-enforcement points and run-time monitoring products:

- *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.
- *webMethods Insight*, which is a monitoring tool that 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 also includes a framework called *SOALink*, which enables it to interoperate with other, third-party PEPs.

Use of CentraSite by other Software AG Products

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

- 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 impact-analysis 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.

2 **CentraSite Editions**

■ CentraSite Editions	10
■ The CentraSite Feature Matrix	10

This chapter covers the following topics:

CentraSite Editions

CentraSite is available in the following editions. The edition that is enabled on your machine depends on the type of license that you have acquired from Software AG.

■ CentraSite Community Edition (CE)

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

■ CentraSite ActiveSOA

CentraSite ActiveSOA is a licensed version of CentraSite that has all of the features of the Community Edition plus a powerful set of SOA governance features. CentraSite ActiveSOA enables organizations to enforce standard processes and conventions through the application of policies. It includes lifecycle-management, which enables users to define lifecycle models and manage the state of an asset using the CentraSite Control user interface. It also supports the deployment of policies and virtual services on policy enforcement points in the run-time environment.

The CentraSite Feature Matrix

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

Feature		CE	ActiveSOA
Registry		✓	✓
Repository		✓	✓
Catalog Features		✓	✓
	- Browse Assets	✓	✓
	- Publish Assets	✓	✓
	- Keyword Search	✓	✓
	- Advanced Search	✓	✓
	- Version Asset	✓	✓

Feature		CE	ActiveSOA
	- 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		✓	✓
User, Groups, Roles Management		✓	✓
Taxonomy Management (Custom Taxonomies)		✓	✓
Asset Type Management (Custom Asset Types)			✓
Reporting		✓	✓
Eclipse-based GUI		✓	✓
Browser-based GUI (CentraSite Control)		✓	✓
XQuery and JAXR-based APIs		✓	✓
UDDI API		✓	✓

3

CentraSite Features

▪ The Registry	14
▪ The Catalog	14
▪ The Repository	15
▪ Design/Change-Time Policies	16
▪ Run-Time Policies	16
▪ Virtual Services	17
▪ SOALink	17
▪ Lifecycle Management	18
▪ Reporting	18
▪ Impact Analysis	19
▪ Security and Auditing	19
▪ Role-Based Access	19
▪ Federation with Other Registries	20
▪ GUI Access to CentraSite	20
▪ API Access to CentraSite	21

This chapter introduces the following 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 (e.g., 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 (e.g., 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
- 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 (e.g., assets such as ESB Services, 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. For information about configuring CentraSite to store metadata for custom asset types, see the document *Object Type Management*.



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 as described above. 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 will include 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. Additionally, the repository houses an organization's *supporting document library*. This library contains documents that an organization can arbitrarily attach to objects in the registry. For example, you might upload documents such as technical specifications, sample code and programming guides to the supporting document library, and then link these documents to various assets in the catalog. For more information about uploading documents to the supporting document library, see the section *Working with Supporting Documents* in the document *Using the Asset Catalog*.

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 available only in CentraSite ActiveSOA.

For more information about creating design/change-time policies, see the document *Working with Design/Change-Time Policies*.

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 Control to define run-time policies, associate them with services, and deploy them on specified PEPs 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 available only in CentraSite ActiveSOA.

For more information about creating run-time policies, see the document *Working with Run-Time Policies*.

Virtual Services

A virtual service functions as public-facing proxy for a Web service or XML 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 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 Control. 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 available only in CentraSite ActiveSOA.

For more information about defining virtual services, see the document *Working with Virtualized Services*.

SOALink

SOALink is a framework that enables interoperability between CentraSite and various SOA infrastructure components that participate in the SOA lifecycle. SOALink consists of a set of standards, APIs and extension points supported by CentraSite ActiveSOA.

SOALink leverages standards such as UDDI, SNMP, WS-Policy and WS-PolicyAttachment as the building blocks for the interface. In cases where a particular standard does not exist or is inadequate, SOALink defines a set of extensions to existing standards.

You can use the capabilities provided by SOALink to integrate CentraSite with third-party design-time tools and policy enforcement points.

SOALink is available only in CentraSite ActiveSOA.

Lifecycle Management

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 available only in CentraSite ActiveSOA.

For more information about defining lifecycle models, see the document *Customizing Lifecycle Management*.

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. Or you might 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.

For more information about the reporting capabilities in CentraSite, see the document *Working with Reports and Report Templates*.

Impact Analysis

Impact analysis refers to the ability to evaluate the effect of a proposed change on an existing system. In an SOA environment, where reuse is encouraged and assets often have numerous dependants, the ability to understand the consequences of a change is crucial.

CentraSite enables you to represent the relationships among the objects in your registry using customizable relationship attributes and ad hoc associations. For example, in an asset type that represents a consumer application, you might include a relationship attribute that allows developers to relate consumer applications to the Web services that they consume.

If an asset's relationships have been described using relationship attributes or ad hoc associations, they can be visualized using CentraSite's impact analysis feature. This feature summarizes an asset's relationships in a concise, browsable diagram. Using this display, you can easily locate and examine the objects to which an asset is related and evaluate how change to the asset might affect them.

For more information about defining relationships among assets see the document *Object Type Management*. For more information about viewing the Impact Analysis profile, see the document *Using the Asset Catalog*.

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/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.

For information about permissions and user management, see the document *User, Groups, Roles and Permissions*.

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 (e.g., Organizations, Policies, Life-cycle Models) users in that role can create. They also specify whether a user is allowed to perform certain restricted actions (e.g., 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. For more information about roles, see the document *User, Groups, Roles and Permissions*.

Federation with Other Registries

CentraSite's federation capabilities enable you to automatically synchronize the contents of other CentraSite or UDDI registries with your CentraSite registry. Federation is accomplished by configuring CentraSite to automatically replicate certain types of entries in its registry to other registries within your network or to mirror entries from other registries into your CentraSite registry.

GUI Access to CentraSite

CentraSite includes these graphical user interfaces (GUIs):

- *CentraSite Control* is a browser-based user interface that supplies access to all of CentraSite's features. Administrators use CentraSite Control to perform administrative activities such as managing organizations, users, groups and roles. Developers and analysts use CentraSite Control to browse the asset catalog, publish assets into the SOA, define policies, deploy run-time policies and virtual services, generate reports and monitor the performance of services in the run-time environment.
- The *Business User Interface* 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 into the SOA and generate reports.
- 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.)

API Access to CentraSite

Programmatic access to CentraSite is provided through a number of APIs and protocols, including JAXR and UDDI. For a list of the supported APIs and protocols, see the section *Ways to Interact with CentraSite* in the document *Logging On and Using the CentraSite UIs and APIs* .

4 Topology And Architecture

■ Basic Topology	24
■ CentraSite Architecture	25

This chapter covers the following topics:

Basic Topology

CentraSite consists of two basic components:

- **CentraSite Application Server Tier**

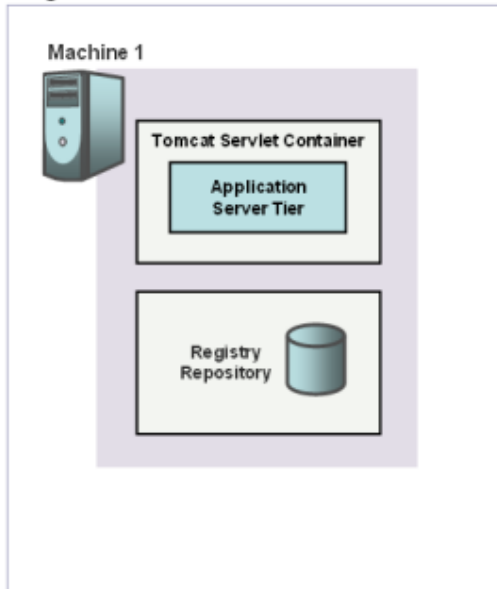
The *CentraSite Application Server Tier* is a servlet that runs in a Tomcat servlet container. This component hosts the CentraSite graphical user interfaces and also supplies the UDDI V3.0 interface to the registry.

- **CentraSite Registry Repository**

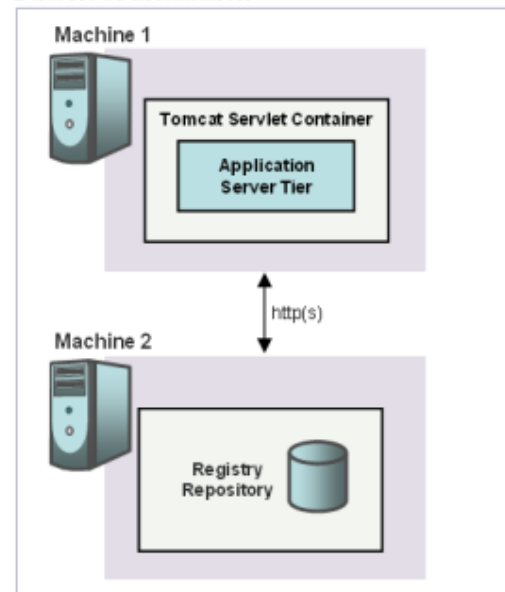
The *CentraSite Registry Repository* is the portion of CentraSite that hosts the registry and the repository.

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

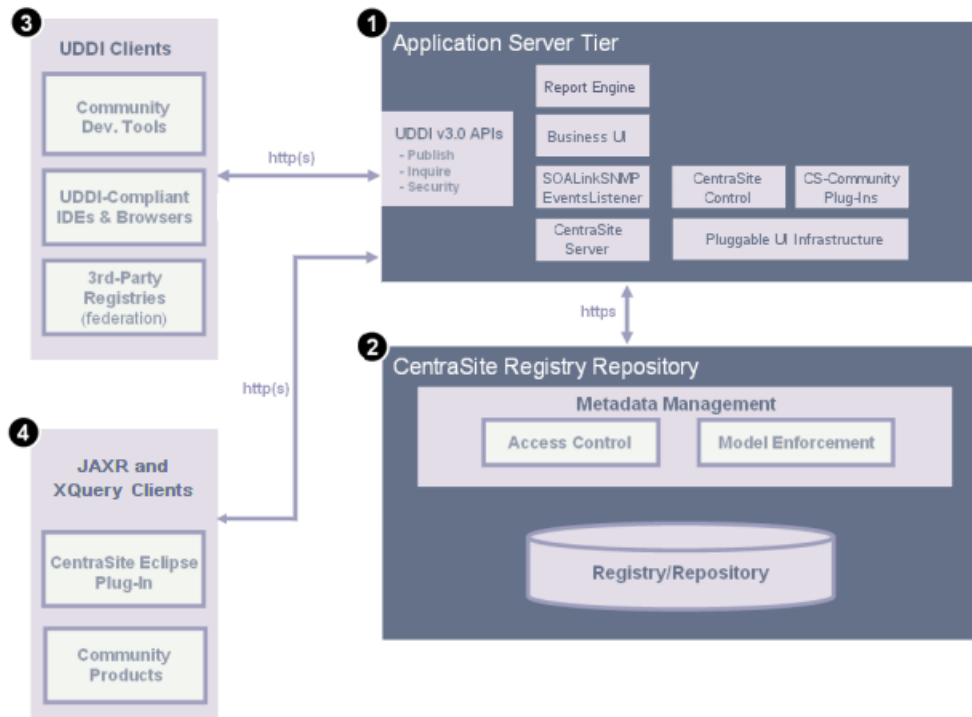


Dual Node Installation



CentraSite 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	<p>The CentraSite Application Server Tier (CAST) hosts CentraSite components that are provided as web applications in an application server. The components of the CentraSite Application Server Tier function as clients of the CentraSite Registry Repository component.</p> <p>CentraSite Control is CentraSite's browser-based user interface, providing general purpose features for administrators, designers and end users. Because this interface is built on Software AG's pluggable UI infrastructure, it can be easily extended and customized (for information about doing this, see the document <i>CentraSite Control Pluggable Architecture</i>). The pluggable infrastructure also enables you to plug into CentraSite additional user interfaces from the CentraSite Community.</p> <p>The Business UI is a lightweight alternative to CentraSite Control, offering high-level features for business users.</p> <p>The CentraSite Server performs authorization of communication from client applications to the CentraSite Registry Repository.</p> <p>The CentraSite Application Server Tier hosts the UDDI services that client programs use to interact with CentraSite using the UDDI V3 API.</p>

#	Description
	The CentraSite Control and UDDI services interact with the Registry and Repository component using the CentraSite API for JAXR (registry) and HTTP (repository).
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 via the CentraSite Server component of the Application Server Tier.</p>
3	UDDI clients interact with CentraSite using the UDDI V3 services that reside on the CentraSite Application Server Tier. UDDI clients include developer tools supplied by CentraSite Community partners, third-party UDDI browsers, UDDI-compliant IDEs and UDDI registries that are federated with CentraSite.
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>

5

The Information Model

■ Overview	28
■ Objects in the CentraSite Information Model	28
■ Relationships Between Objects in the Registry	30

This chapter covers the following topics:

Overview

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.

The object types that make up the CentraSite registry are all extensions of the basic JAXR Registry-Object class.

Objects in the CentraSite Information Model

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

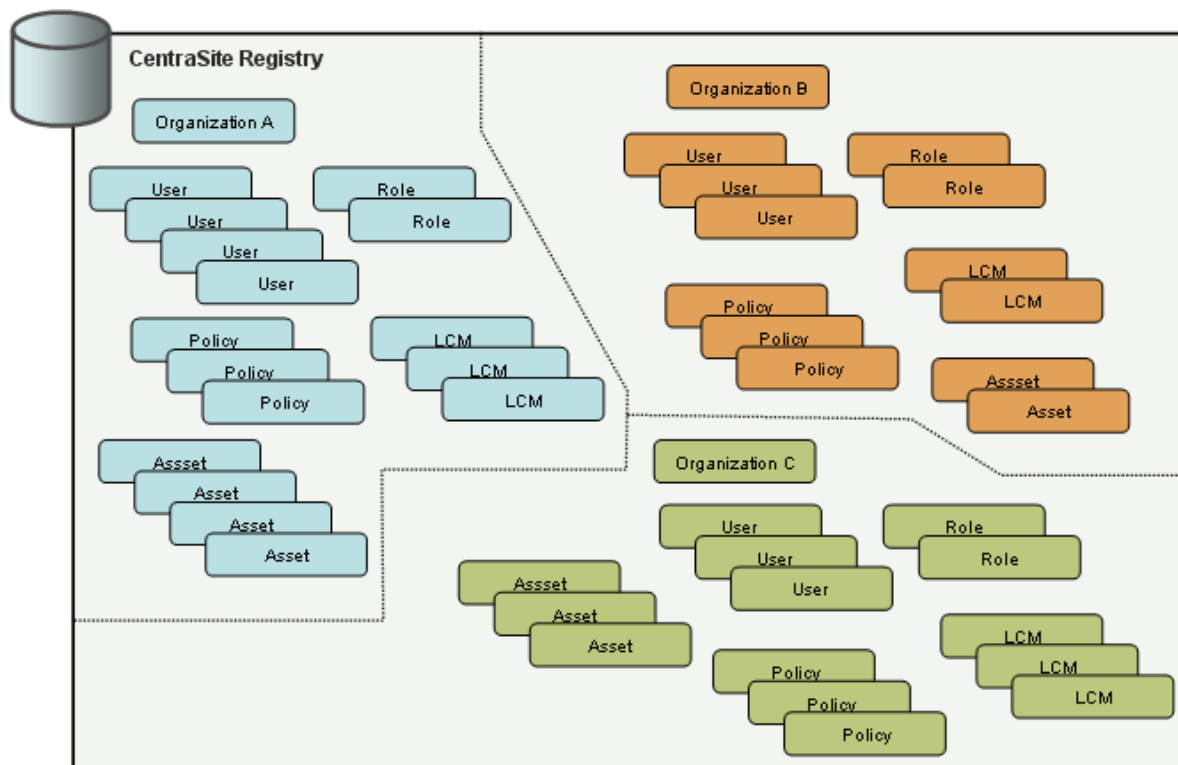
- System-related objects include objects such as organizations, users, groups, roles, taxonomies, policies and lifecycle models. These type of objects do not appear in the catalog, however they play a key role in managing its content.

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.

For a complete description of each of the system-related object types in the CentraSite information model, see the document *Object Type Management*.

- *Assets* refer to registry objects that represent the artifacts in your SOA. As installed, the CentraSite registry supports four asset types (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. For a complete description of the installed asset types and information about defining additional asset types, see the document *Object Type Management*.

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 Web service asset might include a relationship attribute called "Uses", which relates the service to its constituent artifacts (i.e., 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.

For more information about defining relationship attributes, see the document *Object Type Management*.