

CentraSite

Suite Usage Aspects

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-AG-SUITE-95SP1-20140410

Table of Contents

Preface	v
1 Versioning Assets	1
2 Modifying or Deleting Assets	3
3 Publishing an Asset	5
4 Communication with Designer UI	7
Remote Connection from Designer	8
5 User Accounts	9
6 UDDI Clients	11
7 Using CentraSite with ARIS	13
Configuring CentraSite for Use with ARIS	14
The Lifecycle Models that ARIS Uses	14
The Design/Change-Time Policies that ARIS Uses	16
Activating the Lifecycle Models and Design/Change-Time Policies that ARIS Uses	18
Provisioning CentraSite Services into ARIS Architect	29

Preface

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 Control 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.

This document summarizes the main points to watch out for when working with objects and object types used by the suite products.

Versioning Assets

Modifying or Deleting Assets

Publishing an Asset

Communication with Designer UI

User Accounts

UDDI Clients

Using CentraSite with ARIS

1 Versioning Assets

CentraSite offers a versioning capability, which allows you to maintain several versions of an asset. This feature is described in the section *Versioning an Asset* in the document *Using the Asset Catalog*.

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 *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 versioning capability for the asset types 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.

2 Modifying or Deleting Assets

CentraSite users with the appropriate permissions can delete existing assets. This standard functionality is described in the section *Deleting an Asset* in the document *Using the Asset Catalog*.

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 will normally not be able to 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, rather than as a CentraSite user.

3 Publishing an Asset

When a suite product tries to update an asset that it previously created in CentraSite, the update will fail 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 as per 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 will fail since CentraSite will 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.

4

Communication with Designer UI

■ Remote Connection from Designer	8
---	---

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 will 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, etc.).

In such cases it may be necessary to restart/reconnect the following components:

- Tomcat
- CentraSite Registry Repository
- Any client involved

5 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 (e.g. 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 will not limit the functionality of the suite products.

6 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 will be visible in CentraSite Control as an *Organization* object. When you retrieve the object via 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".

7

Using CentraSite with ARIS

■ Configuring CentraSite for Use with ARIS	14
■ The Lifecycle Models that ARIS Uses	14
■ The Design/Change-Time Policies that ARIS Uses	16
■ Activating the Lifecycle Models and Design/Change-Time Policies that ARIS Uses	18
■ Provisioning CentraSite Services into ARIS Architect	29

This section contains information and procedures relating to the use of CentraSite with ARIS.

Configuring CentraSite for Use with ARIS

ARIS uses CentraSite to retrieve/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. To use CentraSite with ARIS, you must first configure and activate these features as described in the following sections.



Note: When users working on the same ARIS database are created as users in different CentraSite organizations, by default they will not have visibility to each others' assets. It is recommended 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 “Read” or “Modify” permissions to the group.

The 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.

- [The BPM Process Lifecycle Model](#)
- [The Service Lifecycle Model](#)
- [Customizing the Predefined Lifecycle Model for Process Objects](#)

The BPM Process Lifecycle Model

The “BPM Process Lifecycle” model defines the following states and transitions:

State	Has valid transitions to...
Requested	Implementing, Rejected
Implementing	Implemented
Implemented	Testing
Testing	Tested
Tested	Available
Available	Retired

State	Has valid transitions to...
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 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 (i.e., inactive). To use CentraSite with ARIS, you must activate the lifecycle model as described in [Activating the Lifecycle Model and Design/Change-Time Policies that ARIS Uses](#).

The Service Lifecycle Model

The “Service Lifecycle” model defines the following states and transitions:

State	Has valid transitions to...
Proposed	In Design, Rejected, Retired
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 as described in [Activating the Lifecycle Model and Design/Change-Time Policies that ARIS Uses](#).

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 [The Design/Change-Time Policies that ARIS Uses](#).



Note: You can only use a customized lifecycle model for Process objects or Service objects if you have the ActiveSOA edition of CentraSite. If you are using the Community Edition, you must use the predefined lifecycle model that CentraSite provides.

The Design/Change-Time Policies that ARIS Uses

CentraSite includes the following predefined policies.

Policy Name	Purpose
ARIS Delete Processes	<p>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.</p> <p>Note that if the Process object in CentraSite is not in a state that permits a transition to the Retired state (i.e, if a transition to the Retired state is not permitted by the Process object's lifecycle model) the policy will fail.</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 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 (i.e, if a transition to the Retired state is not permitted by the Service's lifecycle model) the policy will fail.</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 (i.e, the transition to the Available state is not permitted by the Process object's lifecycle model), the policy will fail.</p>

Policy Name	Purpose
	<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/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/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 ■ PreUpdate ■ OnTrigger
Enforce Unique Name	<p>Ensures that the names of Application Server objects, IS Server objects and IS Connection objects that are created in CentraSite are unique.</p> <ul style="list-style-type: none"> ■ This policy contains the action "Enforce Unique Name", which you can configure to: <ul style="list-style-type: none"> ■ 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: <ul style="list-style-type: none"> ■ PreCreate ■ PreUpdate
Notify ARIS on Service Changes	<p>Notifies the ARIS APG service endpoint when a Service object (native or virtual) in CentraSite is updated.</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 following events occur to a Service object: <ul style="list-style-type: none"> ■ PostUpdate ■ PostStateChange ■ OnTrigger

Policy Name	Purpose
Notify ARIS on Service Completion	<p>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 (e.g, the Productive state).</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 following events occur to a Service object: <ul style="list-style-type: none"> ■ PostStateChange ■ 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.

Activating the Lifecycle Models and Design/Change-Time Policies that ARIS Uses

You use the following procedures to 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. To use CentraSite with ARIS, you must first activate the lifecycle models and policies using the procedures provided in this section.

- [Activating the ARIS-related LCMs and Policies in the ActiveSOA Edition of CentraSite](#)

- [Activating the ARIS-related LCMs and Policies in CentraSite](#)

Activating the ARIS-related LCMs and Policies in the ActiveSOA Edition of CentraSite

If you are using CentraSite ActiveSOA, use the following procedure to activate the lifecycle models and design/change-time policies for ARIS.

▶ To activate the ARIS-related LCM and policies in CentraSite ActiveSOA

- 1 Open CentraSite Control.
- 2 Activate the ARIS lifecycle models using the following steps:
 1. Go to **Administration > Lifecycles > Models**.
 2. Locate the lifecycle model called "BPM Process Lifecycle" and activate it using the procedure described in the section *Activating a Lifecycle Model* in the document *Customizing Lifecycle Management*.
 3. Locate the lifecycle model called "Service Lifecycle" and activate it using the procedure described in the section *Activating a Lifecycle Model* in the document *Customizing Lifecycle Management*.
- 3 Activate the state-change policies using the following steps:
 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. Activate each of the following policies using the procedure described in the topic *Activating a Design/Change-Time Policy* in the section *Functional Scope* of the document *Working with Design/Change-Time Policies*.
 - ARIS Release Processes
 - ARIS Delete Processes
 - ARIS Delete Services
 - ARIS webMethods Integration
 - Reset Lifecycle State to Initial State

For information about these policies, see [The Design/Change-Time Policies that ARIS Uses](#).

- 4 Configure and activate each of the following policies using the procedure described in [Configuring the ARIS Change-Notification Policies](#).
 - Notify ARIS on Service Changes
 - Notify ARIS on Service Completion

■ Notify ARIS on Service Deletion

For information about these policies, see *The Design/Change-Time Policies that ARIS Uses*.

- 5 Configure and activate the "Enforce Unique Name" using the procedure described in *Configuring the Enforce Unique Name Policy*.

For information about this policy, see *The Design/Change-Time Policies that ARIS Uses*.

- 6 Configure and activate the "Reset Lifecycle State to Initial State" using the procedure described in *Configuring the Reset Lifecycle State to Initial State Policy*.

For information about this policy, see *The Design/Change-Time Policies that ARIS Uses*.

Configuring the ARIS Change-Notification Policies

Use the following procedure to configure the "Notify ARIS Service" action in the ARIS change-notification policies (i.e., 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.
 1. Set the following parameters:

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 will submit 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	<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 (see the section <i>Built-In Actions for Design/Change-Time Policies</i> in the document <i>Built-In Design/Change-Time Actions Reference</i> for details). You can use these substitution tokens to customize this message.</p> <pre> <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><i>String</i> The SOAP action that CentraSite will set in the message. If you do not set this parameter, CentraSite will set the SOAP action to the empty string.</p>
Connection Timeout (in milliseconds)	<p><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.</p>
Content Type	<p><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.</p>

2. Click **Save** to save the parameter settings.

- 5 Click **Save** to save the updated policy.
- 6 Activate the policy using the procedure described in the topic *Activating a Design/Change-Time Policy* in the section *Functional Scope* of the document *Working with Design/Change-Time Policies*.

Configuring the 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.
 1. Set the following parameters:

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.
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

2. Click **Save** to save the parameter settings.
- 5 Click **Save** to save the updated policy.
- 6 Activate the policy using the procedure described in the topic *Activating a Design/Change-Time Policy* in the section *Functional Scope* of the document *Working with Design/Change-Time Policies*.

Configuring the 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.

1. Set the following parameter:

Change State To	String Set the parameter to any state of the "BPM Process Lifecycle" model.
-----------------	---

The event scope of this action is:

- OnTrigger

2. Click **Save** to save the parameter setting.

- 5 Click **Save** to save the updated policy.
- 6 Activate the policy using the procedure described in the topic *Activating a Design/Change-Time Policy* in the section *Functional Scope* of the document *Working with Design/Change-Time Policies*.

Activating the ARIS-related LCMs and Policies in CentraSite

InCentraSite, the ActivatePoliciesForARIS command-line utility can be used for:

- Activate the lifecycle models and design/change-time policies that ARIS requires.
- Configure the "Notify ARIS Service" action in the ARIS change-notification policies (i.e., 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 via its user interface.



Note: To run the ActivatePoliciesForARIS utility, you must either have access to the command line on the machine where the CentraSite Application Server Tier is installed or that system must be accessible via HTTP. Additionally, you must belong to the CentraSite Administrator role or a role that has permission to manage system-wide lifecycle models and system-wide design/change-time policies.

- [ActivatePoliciesForARIS from a Java Class](#)
- [ActivatePoliciesForARIS from a Command Line](#)
- [Creating a Script File for Activating the ARIS LCM and Policies](#)

ActivatePoliciesForARIS from a Java Class

Use the following procedure to activate ARIS-related LCMs and Policies from a Java class:

► To run the ActivatePoliciesForARIS command-line utility

- 1 Create a script file as described in [Creating a Script File for Activating the ARIS LCM and Policies](#).
- 2 Execute the script file with the following parameters:

```
yourScriptFile -user yourCSUserID -password yourPassword [-url yourCSURL | [-h  
yourHostname -p yourPort]]
```

Example

```
myScript -user jcallen -password j45Hk19a [-url  
http://myserver:53305/CentraSite/CentraSite | [- h myserver -p 53305]]
```



Note: If the registry/repository component is not running on the same machine as the CentraSite Application Server Tier, or is running on a port other than 53305, you must also include the -url parameter on the command line, followed by the URL for the CentraSite registry/repository (e.g., -url http://myserver:53305/CentraSite/CentraSite).

ActivatePoliciesForARIS from a Command Line

To activate ARIS-related LCMs and Policies, run the utility with the following input parameters:

- [Command Line Under Windows](#)
- [Command Line Under Linux](#)
- [Input Parameters](#)

Command Line Under Windows

Use the following procedure to ActivatePoliciesForARIS under Windows:

1. Open *EnableARISIntegration.cmd* in a text editor.
2. Add the following property statement:

```
<CentraSiteInstallDir>\utilities\ EnableARISIntegration.cmd -user <USERNAME> ↵  
-password <PASSWORD> [-dburl <CENTRASITE-URL> | [-h <HOSTNAME> -p <PORT>]]
```

where, *<CentraSiteInstallDir>* is the CentraSite installation directory. By default, this is the *CentraSite* folder under *<SuiteInstallDir>*.

Command Line Under Linux

Use the following procedure to ActivatePoliciesForARIS under Windows:

1. Open *EnableARISIntegration.sh* in a text editor.
2. Add the following property statement:

```
<CentraSiteInstallDir>/utilities/ EnableARISIntegration.sh -user <USERNAME> ↵  
-password <PASSWORD> [-dburl <CENTRASITE-URL> | -h <HOSTNAME> -p <PORT>]
```

where, *<CentraSiteInstallDir>* is the CentraSite installation directory. By default, this is the *CentraSite* folder under *<SuiteInstallDir>*.

Input Parameters

The following table describes the complete set of input parameters that you can use with the ActivatePoliciesForARIS utility:

Parameter	Description
<i>USERNAME</i>	<i>Required.</i> Your CentraSite user ID.
<i>PASSWORD</i>	<i>Required.</i> The password for your CentraSite user account.
<i>CENTRASITE-URL</i>	<p>The fully qualified URL for the CentraSite registry/repository.</p> <p>If you omit this parameter, the importer assumes that the registry/repository resides at <code>http://localhost:53305/CentraSite/CentraSite</code>.</p> <p>Note: If the registry/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/or <code>-p</code> parameters, the <code>-h</code> and <code>-p</code> parameters will be ignored.)</p>
<i>HOSTNAME</i>	<p>The host name or IP address of the computer where the CentraSite registry/repository component is running.</p> <p>If you omit this parameter, the importer assumes that the registry/repository is running on <code>localhost</code>.</p>
<i>PORT</i>	<p>The port number on which the CentraSite registry/repository is configured to listen for incoming requests.</p> <p>If you omit this parameter, the importer assumes that the registry/repository is listening on the default port, 53305.</p> <p>Note:</p>

Creating a Script File for Activating the ARIS LCM and Policies

The `ActivatePoliciesForARIS` utility is a Java class whose `main()` method executes when you execute `ActivatePoliciesForARIS` 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 `ActivatePoliciesForARIS` as described below.

Creating a Script File for Windows (a .bat 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 ARISPATH=CentraSiteHomeDirectory\rts\aris
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
```



```

set BASEDIR=%~dp0
cd /d %ARISPATH%
set ARIS_CLASSPATH=
for %%I in ("%ARISPATH%\*.jar") do call "CentraSiteHomeDirectory\bin\cfg\lcp.cmd" %%I

set LOCAL_CLASSPATH=%LOCAL_CLASSPATH%;%ARIS_CLASSPATH%
cd /d %REDIST%

%JAVAEXE% -cp %LOCAL_CLASSPATH% com.centrasite.aris.util.ActivatePoliciesForARIS %*
cd /d %BASEDIR%

```

Example

```

@echo off
REM
REM Run ARIS LCM and Policy Activator
REM
set JAVAEXE=D:\Fix7\CentraSiteGE7\jdk1.5.0_12\bin\java
set REDIST=C:\SoftwareAG\CentraSite\redist
set ARISPATH=C:\SoftwareAG\CentraSite\rts\aris
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

set BASEDIR=%~dp0
cd /d %ARISPATH%
set ARIS_CLASSPATH=
for %%I in ("%ARISPATH%\*.jar") do call "C:\SoftwareAG\CentraSite\bin\cfg\lcp.cmd" %%I

set LOCAL_CLASSPATH=%LOCAL_CLASSPATH%;%ARIS_CLASSPATH%
cd /d %REDIST%

%JAVAEXE% -cp %LOCAL_CLASSPATH% com.centrasite.aris.util.ActivatePoliciesForARIS %*
cd /d %BASEDIR%

```

Creating a 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="CentraSiteHomeDirectory/redist"
set arispath="CentraSiteHomeDirectory/aris"
set mainjar="CentraSiteARISIntegration.jar"
set delim='\: '
cd "$redist"
set cl=""
foreach j ( `ls *.jar` )
  if ($cl != "") set cl=${cl}${delim}
  set cl=${cl}${j}
end
set delim='\: '
cd "$arispath"
set acl=""
foreach j ( `ls *.jar` )
  if ($acl != "") set acl=${acl}${delim}
  set acl=${acl}${j}
end
setenv CLASSPATH ${mainjar}${delim}${cl}${acl}
cd "$redist"
$javaexe com.centrasite.aris.util.ActivatePoliciesForARIS $*
```

Example

```
#!/bin/csh
#
# Run ARIS LCM and Policy Activator
#
set javaexe="/.../softwareag/cjp/v16/bin/java"
set redist="/.../softwareag/CentraSite/redist"
set arispath="/.../softwareag/CentraSite/rts/aris"
set mainjar="CentraSiteARISIntegration.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
set delim='\: '
cd "$arispath"
set acl=""
foreach j ( `ls *.jar` )
  if ($acl != "") set acl=${acl}${delim}
  set acl=${acl}${j}
end
setenv CLASSPATH ${mainjar}${delim}${cl}${acl}
cd "$redist"
$javaexe com.centrasite.aris.util.ActivatePoliciesForARIS $*
```

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>
```

