CentraSite

Customizing Lifecycle Management

Version 9.6

April 2014

This document applies to CentraSite Version 9.6.

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

Document ID: IINM-OIINMAG-LCM-96-20140318

Table of Contents

Customizing Lifecycle Management

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.

Lifecycle models can theoretically be complex and go across multiple stages (machines). In CentraSite, the stages are called nodes. A lifecycle node is a machine within a lifecycle management model.

Lifecycle models can be applied to:

- Assets
- Policies
- Lifecycle models

> **Note:** CentraSite provides predefined lifecycle models for several types of objects. For information about these lifecycle models, see *The Predefined Lifecycle Models Installed with CentraSite*.

The description of lifecycle management in CentraSite consists of the following sections:

1 Introduction

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.

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.

Lifecycle Models

A lifecycle model is a state machine that can be maintained in CentraSite and applied to an asset type. In CentraSite, the lifecycles of assets, policies and also lifecycle models themselves can be managed. Lifecycle models are defined by administrators for their organization.

There are two levels of lifecycle models: system-wide models and organization-specific models. A system-wide model applies to objects that can be owned by any organization, whereas an organization-specific model applies only to objects that are owned by a specific organization. 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 will complete their lifecycle without effects. Only new objects will be assigned to the new system-wide model.

Each organization-specific lifecycle model belongs to an organization. System-wide models do not belong to any organization, but apply to all organizations. System-wide models may have organization-specific policies attached to them.

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 creating user belongs to an organization, and that organization's active lifecycle model will be 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.

Global models may have organization-specific policies attached to them. Organizational models are technically hierarchical, as organizations may contain sub-organizations.

A lifecycle model can itself be assigned to a lifecycle model, i.e. to enable a lifecycle model it may be necessary to go through a lifecycle model to activate it for usage.

Lifecycle management is possible for the object types that are classified as assets, as well as for policies and lifecycle models themselves.

The following points should also be noted:

- Organization-specific models are defined by the administrator of the organization.
- An object type may have more than one lifecycle model defined, but only one per organization can be defined as active at any given time.

Stages for Lifecycle Registries

Most software engineering methodologies define fundamental phases, that start with requirements or needs for a software and end with a productive software. These phases apply to typical lifecycle stages like Analysis, Development, Test, Production. In general these stages are represented by dedicated software and hardware environments like server, repositories and tools. A registry adds benefit to all stages as it references, collects and maintains structured information of relevance in each of these stages. Especially for enterprises following the service oriented architecture, the CentraSite provides registry stages that can be used to take strong control over the lifecycle process the on one hand but allow for loosely coupled stage environments on the other hand.

If we take the example of the stages Architecture, Development, Test and Production, the following tasks could apply to them:

- **Architecture**
  If services are developed in a top-down approach, the Architecture registry keeps track of all planning and preparation information. A service request triggers design and approval steps that may lead to the implementation or rejection of the requested service. During the service design process, the service interface as well as metadata may undergo changes. Only the final approval by an authorized role marks the service as ready for implementation. Examples of metadata which may be of special interest in the Architecture stage are:
  - Organization and contact responsible for service provisioning;
  - Date planned to go live;
  - Planned service consumer; dates planned to consume the service.

- **Development**
  For the implementation of the service, it is important that only approved service descriptions are used. Service consumers often start to implement their service usage at the same time and rely on the service description. Multiple instances of Development registries may exist. These registries may be volatile, because development environments exist for the duration of one release and can be set up in a different configuration. Therefore it must always be possible to set up a development registry with the services that are of interest for the projects using this environment.
- **Test**
  In loosely coupled service oriented architectures, the test laboratory often integrates for the first time different service participants which previously worked in dedicated environments. Furthermore the test cycle requires the repeated handover from development to test. The registries can support these dynamics by providing easy-to-use export and import facilities for all service related information and components. As the number of service provider and consumer applications grows, the test environments also get volatile and its registries need the flexibility to be adapted to the different combinations of test participants.
- **Production**
  Finally, the registry in the production environment is the single point of truth for all applications requesting service lookup. Objects should enter this registry only if they have passed defined approval steps and if they are from a prescribed origin, for example the test registry. Metadata in the registry can inform about contacts in case of malfunction. Or the registry can be used to notify contacts in case of planned restrictions in availability.

Given these different tasks and requirements, it is obvious that multiple registries become important for optimal support of specific needs; while the Architecture is interested in services, service versions and participants still in planning, the Production is not. Test registries keep objects and references with physical properties - like endpoints - that are only valid during the test phase and are of no interest either to Architecture or Production. However, it is important that defined transitions exist from one stage to another and that each object can be traced to its current stage.

This four-stage topology is an example for enterprises with a strong top-down approach in their enterprise architecture; simpler or more complex topologies are also possible.

Note: This four-stage topology is an example for enterprises with a strong top-down approach in their enterprise architecture; simpler or more complex topologies are also possible. In practice, any deployment involving more than two stages leads to significant administrative overhead involved in managing multiple registries (in particular, creating LCMs and policies across all instances) and promoting objects from one registry to another.

Defining a Lifecycle Model in CentraSite

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).

States for Registry Objects

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.

Applying Lifecycle Models to Asset Types

You can associate a lifecycle model with one or more asset types. 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 Virtual Types

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 enabled 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	Will follow the lifecycle model assigned to the Base Type
ENABLED	HAS an assigned lifecycle model	HAS an assigned lifecycle model	Will follow the lifecycle model assigned to the Virtual Type
ENABLED	DOES NOT HAVE an assigned lifecycle model	DOES NOT HAVE an assigned lifecycle model	Will not have an assigned lifecycle model
ENABLED	DOES NOT HAVE an assigned lifecycle model	HAS an assigned lifecycle model	Will follow the lifecycle model assigned to the Virtual Type
DISABLED	HAS an assigned lifecycle model	DOES NOT HAVE an assigned lifecycle model	Will not have an assigned lifecycle model
DISABLED	HAS an assigned lifecycle model	HAS an assigned lifecycle model	Will follow 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	DOES NOT HAVE an assigned lifecycle model	DOES NOT HAVE an assigned lifecycle model	Will not have an assigned lifecycle model
DISABLED	DOES NOT HAVE an assigned lifecycle model	HAS an assigned lifecycle model	Will follow the lifecycle model assigned to the Virtual Type

Note: If a virtual type initially inherits its lifecycle model from the base type and it 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 will continue following that model. Only instances of the virtual type that are created after the new lifecycle model is assigned to the virtual type will comply with the virtual type's newly assigned lifecycle model.

For more information about virtual types and the **Inherit Base Type LCM** option, see the section *What is a Virtual Type?* in the document *Object Type Management*. For information about which predefined types in CentraSite are virtual types, see the section *The Predefined Asset Types in CentraSite* in the document *Object Type Management*.

2 Who Can Create and Manage Lifecycle Models?

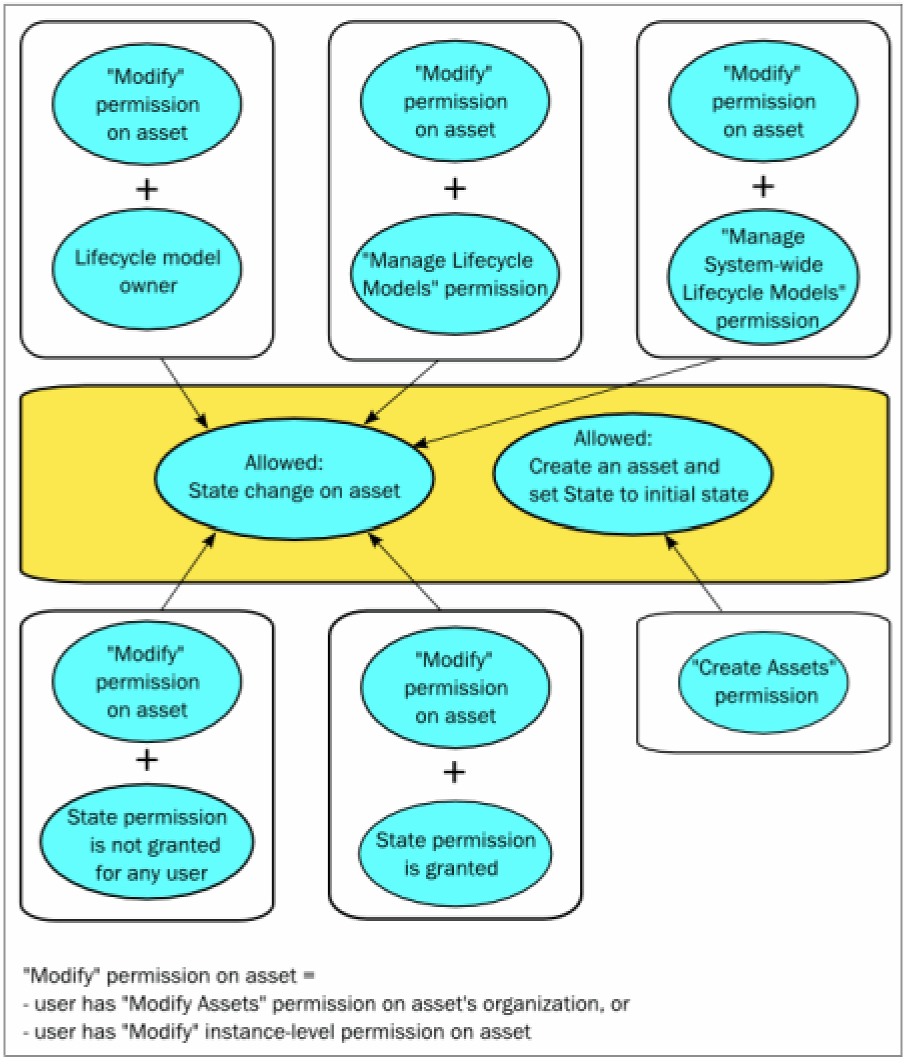
To create and manage (i.e., view, edit, and delete) lifecycle models for an organization, you must belong to a role that has the "Manage Lifecycle Models" permission for the organization.

To create and manage *system-wide* lifecycle models (i.e., lifecycle models that apply to all organizations), you must belong to a role that has the "Manage System-wide Lifecycle Models" permission.

To be able to activate an organization-specific lifecycle model or a system-wide lifecycle model, you must have the "Modify Assets" permission for the organization(s) that own the lifecycle model's assigned asset types. This is because when a lifecycle model is activated, assets of its assigned asset types in the organization(s) will be set to the initial state of the lifecycle model, and this requires the modify permission for the assets.

To be able to delete an organization-specific lifecycle model or a system-wide lifecycle model, you must have the "Modify Assets" permission for the organization(s) that own the lifecycle model's assigned asset types. This is because when a lifecycle model is deleted, assets of its assigned asset types in the organization(s) will be updated to remove the reference to the lifecycle model, and this requires the modify permission for the assets.

Several rules determine who can change the lifecycle state of an asset. These rules are summarized in the following diagram.


"Modify" permission on asset
+
Lifecycle model owner
"Modify" permission on asset
+
"Manage Lifecycle Models" permission
"Modify" permission on asset
+
"Manage System-wide Lifecycle Models" permission
Allowed: State change on asset
Allowed: Create an asset and set State to initial state
"Modify" permission on asset
+
State permission is not granted for any user
"Modify" permission on asset
+
State permission is granted
"Create Assets" permission
"Modify" permission on asset =
- user has "Modify Assets" permission on asset's organization, or
- user has "Modify" instance-level permission on asset

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 that the "Manage Assets" permission does NOT include the rights to change lifecycle states.

To see the full list of predefined roles that by default include the "Manage Lifecycle Models" or "Manage System-Wide Lifecycle Models" permission, see the section *About Roles and Permissions* in the document *Users, Groups, Roles and Permissions* .

For more information about permissions, see the CentraSite online documentation's section *Users, Groups, Roles and Permissions* > *About Roles and Permissions*.

3 Viewing or Editing a Lifecycle Model

Use the following procedure to view or edit the properties of a lifecycle model.

To view and/or edit a lifecycle model

1 In CentraSite Control, go to **Administration > Lifecycles > Models**.

2 By default, the names of all of the available lifecycle models are displayed.

If you want 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 will be updated to show only those lifecycle models whose name contains the specified characters. The wildcard character "%" is supported.

3 Locate the lifecycle model that you want to view or edit.

4 From the lifecycle model's context menu, select **Details**.

5 View or edit the attributes on the Edit Lifecycle Model page as required. For additional information about the attributes on this page, see the relevant steps in ***Creating a Lifecycle Model***.

6 Click **Save**.

You can view multiple lifecycle models as follows:

To view multiple asset types

1 In CentraSite Control, go to **Administration > Lifecycles > Models**.

2 Mark the checkboxes of the lifecycle models whose details you want to view.

3 In the **Actions** menu, click **Details**.

The **Details** view of each of the selected lifecycle models is now displayed.

4 Creating a Lifecycle Model

The definition of a lifecycle model consists of specifying basic attributes of the model such as its name and owning organization, the model's states and transitions, user permissions, and the associated asset types and policies. These are described in the following sections.

Defining the basic attributes of the model

▶ To define the basic attributes of the model

1 In CentraSite Control, go to **Administration > Lifecycles > Models**.

2 Click **Add Model**.

3 In the field **Name**, enter the name of the lifecycle model.

4 In the field **Description**, you can optionally add a comment that describes the purpose of the lifecycle model.

5 In the field **Version**, you can supply a user-defined version number for the model.

6 In the **Organization** drop-down box, choose the organization for which the lifecycle model applies.

The organization names displayed in the drop-down box are the 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.

If you have the "Manage System-wide Lifecycle Models" permission, you will be able to select the "All" option, meaning that the lifecycle model is a system-wide lifecycle model.

⚠ **Important:** Set the system-wide property ("All") with care. You cannot change this assignment after the lifecycle model is created.

Defining the States and State Transitions of the Model

▶ To define the states and state transitions of the model

1 Click the **States** tab, if it is not already selected.

2 Click **Add State**.

3 In the field **State Name**, supply the name of your state.

4 In the field **Description**, you can optionally enter a comment that describes the purpose of the state.

5 In the area labeled **Transitions**, specify a state than can be reached as a result of a transition from the current state. In the column **Target State**, you can select the target state from a drop-down list containing all of the states defined so far.

> **Note:** You may find it more 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, select the "+" icon to create a new line.

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 choosing the triangle icon in the header line of the **Transitions** area.

6 Use the **Add State** button to define additional states as required.

Associating an Asset Type with the Model

When a lifecycle model is assigned to an asset type, and the lifecycle mode is activated (i.e. 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 will be automatically associated with the lifecycle model and will be 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 will be 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. For information about creating a new version of a lifecycle model, see **Updating a lifecycle model**.

To assign a lifecycle model to an asset

1 In CentraSite Control, go to **Administration > Lifecycles > Models**.

2 In the list of lifecycle models displayed, click the required lifecycle model to display its details page.

3 Select the **Associated Types** tab. This displays a list of available asset types that can be associated with the lifecycle model.

4. To define that a particular asset type should be associated with the lifecycle model, select the appropriate asset type from the list shown in the box **Available Types**, then use the arrow buttons to copy the asset type into the box **Selected Types**.

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 types associated with it, the model will automatically be applied to those virtual types that meet the following conditions: 1) the virtual type's **Inherit Base Type LCM** option is enabled and 2) the virtual types does not have a lifecycle model assigned to it. For more information about how CentraSite applies lifecycle models to base types and virtual types, see ***Applying Lifecycle Models to Virtual Types***.

Defining the User and Group Permissions Associated with the States

Now 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 define the user and group permissions associated with the model

1. Start by selecting the **State Permissions** tab.
2. Click **Add Users/Groups** to display the list of currently defined users and groups.
3. From the list of displayed names, choose a user or group for whom you wish to grant or deny permission to change the lifecycle state of an asset.
4. Select the states into which the user or group is allowed to move an asset. You can either check the checkboxes for the states individually, or select them all by checking the corresponding box in the **ALL** column. To deselect a state that is already selected, uncheck its 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.
5. Repeat the previous steps for additional users or groups as required.

Defining Policies Associated with the Model

To define policies associated with the model

1 Finally, you can define that one or more policies will be triggered when a state in the lifecycle is entered.

Before proceeding, click **Save** to ensure that the model that you have defined so far validates (i.e. contains no inconsistencies).

2 Return to the **States** tab and click the **Policies** button for the required state. This displays a list of the currently defined policies for the state, and whether or not the policies are currently active or inactive.

A policy can be assigned to a particular state when you create a policy.

See the document *Working with Design/Change-Time Policies* for details.

In the CentraSite online documentation see the document *Working with Design/Change-Time Policies* for details.

If you want to see the details of any of the given policies, select the policy name. This opens the **Policy Details** dialog.

Creating Lifecycle Stages

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 create a lifecycle stage, proceed as follows:

To create a lifecycle stage

1 In CentraSite Control, choose **Administration > Lifecycles > Stages**.

2 Click **Add Stage**. The **Add Stage** dialog opens.

3 Enter values for the dialog fields as follows:

Fields Name	Contents
Name	The stage name. Here you can enter any text string as the name of the stage.
Description	(Optional) A description of the purpose of the stage.
Host Name	The name of the HTTP host machine where the lifecycle stage is deployed on a CentraSite installation.
Host Port	The HTTP port number of the CentraSite installation on the host machine where the lifecycle stage is deployed.
Use SSL	Mark this checkbox if the communication with the specified host machine should use the SSL protocol for secure communication.

4 Click **OK**.

Defining Stage Transitions

After you have defined your states and state transitions, you can define a 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

1. In the details page of the lifecycle model, click the **Stages** button beside the appropriate state name.
2. From the list of stage names displayed, select the stage or stages that can be the target of a transition from the selected state.
3. Click **OK**.

Validating and Saving the Model

When you have completed all of the above steps, click **Save**. When you do this, the model is validated. If there are no validation errors, the lifecycle model is stored. If any part of the model is invalid, this will be indicated by an appropriate error message.

A 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.

5 Activating a 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	Meaning
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 (i.e. in the state "New"). To activate the model, you must change its state as described below.

⚠ **Important:** 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 will be set to the initial state of the lifecycle model, and this requires the modify permission for the assets.

▶ To activate a lifecycle model

1 Open the Edit Lifecycle Model page for the lifecycle model, if it is not open already.

You reach this page by choosing **Administration > Lifecycles > Models**, then choosing the appropriate model from the displayed list.

2 Click **Change State**.

3 Select the state "Productive", then click **OK**.

While a lifecycle model is in the "Productive" state, you cannot modify it. If you need to modify a lifecycle model, you must create a new version of it, as described in the section **Updating a lifecycle model**.

6 Updating a Lifecycle Model

This section describes how to update a lifecycle model that is stored in CentraSite. To update a lifecycle model, you create a new version of the model, and in the new version you can add states, remove states, rename states, change the relationships between the states and change the list of assigned asset types.

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.

To update a lifecycle model, proceed as follows:

To update a lifecycle model

1 In CentraSite Control, go to **Administration > Lifecycles > Models**.

2 Locate the name of the lifecycle model that you want to update.

3 Create a new version of the lifecycle model by choosing **Create new version** in the context menu.

When you create the 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".

4 Make any changes you require to the new version of the lifecycle model.

5 Set the new version of the lifecycle model to the "Productive" state.

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.

7 Deleting a State from a Model

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 model

1. Display a list of available lifecycle models. If you need procedures for this step, see *Viewing the List of Lifecycle Models*.
2. Locate the required lifecycle model from the list and open its detail view.
3. In the detail view, select the **States** tab if it is not already selected.
4. Mark the checkbox of the state you wish to delete.
5. Click **Delete**.
6. In the other states of the model, remove all transitions to the deleted state.
7. Save your changes.

8 Deleting a Lifecycle Model

If you wish 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 will be 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 will not be deleted. In this case, assets governed by this lifecycle model will continue 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, i.e. 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.

Use the following procedure to delete the most recent version of a lifecycle model.

To delete the most recent version of a lifecycle model

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.
2. If the lifecycle model that you wish to delete is currently in "Productive" mode, select the entry **Change Lifecycle State** from the context menu and change the state to "Retired".
3. Mark the checkbox for the required version of lifecycle model.
4. In the context menu, click **Delete**.

   The selected version will be deleted; all older versions will not be affected.

Use the following procedure to delete a version and all older versions of a lifecycle model.

To delete a version and all older versions of a lifecycle model (i.e. to purge)

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.
2. If any version of the lifecycle model that you wish to purge is currently in "Productive" mode, select the entry **Change Lifecycle State** from the context menu of the version and change the state to "Retired".
3. Mark the checkbox for the version of lifecycle model where you want to begin purging.
4. In the context menu, click **Purge**.

   The selected version and all older versions will be deleted; any newer versions will not be affected.

You can delete multiple lifecycle models in a single step. The rules described above for deleting a single lifecycle model apply also when deleting multiple lifecycle models.

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 will be deleted. The same applies to any other lifecycle models for which you do not have the required permission.

To delete multiple lifecycle models in a single operation

1. In CentraSite Control, go to **Administration > Lifecycles > Models**.
2. Mark the checkboxes of the lifecycle models that you want to delete.

3 In the **Actions** menu, click **Delete**.

9 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.

10 The Predefined Lifecycle Models Installed with CentraSite

The following table lists the predefined lifecycle models that are installed with CentraSite.

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 in any way you like. These lifecycle models are not active when you install CentraSite. If you want to use them, you must activate them.

Model Name	Applies to	Active upon Installation of CentraSite?	Customizable?
Lifecycle Model for Lifecycles	Lifecycle Models objects	Yes	In limited ways. See ***Lifecycle Model for Lifecycle Models (LCM for LCMs)***.
Policy Lifecycle	Policy objects (Both design/change-time policies and run-time policies)	Yes	In limited ways. See the section *Customizing the Predefined Lifecycle Model for Policies* in the section *Functional Scope* in the document *Working with Design/Change-Time Policies*.
BPM Process Lifecycle	Process objects	No	Yes
Service Lifecycle	Service assets	No	Yes