CentraSite

Developing Custom Actions

Version 9.5 SP1

November 2013

Document ID: IINM-OIINMDG-ACTIONSC-95SP1-20140410

Table of Contents

Developing Custom Actions

If you would like a policy to execute a task that is not provided by a built-in action, you can create a custom action to perform the work. For example, a custom action can consist of a Java class or a Groovy script that performs the required task, such as running a test, creating a required attribute or logging an entry in an external database. You can insert custom actions into a policy just like you would insert a built-in action.

The content is organized under the following sections:

Planning to Create Custom Actions	Provides an overview of planning for a custom action.
About Action Categories	Discusses the CentraSite system action categories and custom action categories.
About Action Templates	Describes the contents of the action templates.
About Parameter Templates	Describes the contents of the parameter templates.
Who Can Create and Manage Action Categories or Templates?	Describes the who can create and manage action categories or templates.
Viewing the Action Categories List	Describes how to view the list of action categories in CentraSite.
Adding Custom Actions Using CentraSite's UI	Describes how to create custom actions using CentraSite's UI.
Adding Custom Actions Using APIs	Describes how to create custom actions using APIs.
Viewing or Editing Action Categories or Templates	Describes how to view or edit action categories and templates.
Downloading Rules from System Action Templates	Describes how to download rules from the system action templates.
Deleting Custom Action Categories and Templates	Describes how to delete action categories and templates.
Versioning a Custom Action	Describes how to create new versions of a custom action.
Javadoc: policyAPI	Contains an overview of the policy API.
Sample Custom Actions	Describes the sample custom actions that are provided in your CentraSite installation. One action contains a Java class rule, and the other contains a Groovy script rule.

1 Planning to Create Custom Actions

Creating a custom action consists of the following high-level steps:

1. Create a custom *action category* with which to associate the action.
2. Create an *action template* to specify the scope of objects and events to which the action applies.
3. For an action template to be used in a design or change time policy, you specify the location of your custom *action rule*, which fires when the action executes.
4. Add *parameter templates* to define the parameters that serve as input to the action.

There are two ways you can add custom actions:

- You can use CentraSite Control to create action categories and action templates, as described in ***Adding Custom Actions Using CentraSite's UI***. Then, you create action rules programmatically (as Java programs or Groovy scripts), which is described in ***Creating Action Rules***. You upload the rules into action templates when you create the action templates, using CentraSite Control.

  —OR—

- You can create action categories, action templates and action rules programmatically, as described in the section ***Adding Custom Actions Using APIs***. Then, you upload the rules (Java programs or Groovy scripts) into action templates using JAXR-based calls in the action templates.

2 About Action Categories

Action categories identify the action templates assigned to the category, and the type of action those templates represent. CentraSite includes a set of predefined action categories and templates.

System Action Categories

By default, the policy actions that are installed with CentraSite are grouped into the following categories.

- Design-Time Category
- Change-Time Category
- Run-Time Category
- Global Category
- WS-I Category

For more information on the policy actions that CentraSite ships, see the documents *Built-In Design/Change-Time Actions Reference* and *Built-In Run-Time Actions Reference*.

Custom Action Categories

If you would like to enforce policies using actions that are not provided by your CentraSite, you can create your own categories to define the custom actions. For example, a custom category can consist of user-defined actions to enforce policies. You can enforce policies of the custom actions just like you would enforce policies using the system (built-in) actions.

3 About Action Templates

An action template specifies the object and event types to which the action applies. In addition, an action template for a design or change time policy contains your custom action rule, which fires when the action executes.

Types of Actions

CentraSite supports the following types of actions:

- *Manual Actions* are long-run processes that involve manual user intervention to complete the execution of the action. For example, Approval is a manual action.

  Be aware that although CentraSite allows you to use system-defined manual actions in creating policies, you cannot create a new manual action.
- *Axiomatic Actions* are simple actions used to configure parameters. No code is involved in the execution of axiomatic actions. Axiomatic actions are used in run-time policies.
- *Programmatic Actions* are usually executed by means of program code. Specifically, a programmatic action fires an *action rule* when the action executes. You write an action rule as a Java class or a Groovy script. Programmatic actions are used only in design/change-time policies. CentraSite provides sample action rules, as described in the following section.

  You upload action rules when you create a custom action template, as described in ***Adding an Action Template to the Action Category***.

For more information on the policy actions that CentraSite ships, see the documents *Built-In Design/Change-Time Actions Reference* and *Built-In Run-Time Actions Reference*.

Supported Object Events

Although it is possible to create an action template whose scope encompasses any combination of object types and event types, be aware that not all combinations are enforceable. This is because certain types of events do not occur for certain types of objects.

For example, a PreStateChange event occurs only on Assets, Policies and Lifecycle Models. If you create a policy for a PreStateChange event on a User object, that policy will never execute, because a PreStateChange event will never occur on a User object.

To see the types of events that each object type supports, see the topic *Working with Design/Change-Time Policies > Functional Scope > Supported Object and Event Combinations*.

4 About Parameter Templates

Parameter templates are a part of the action template. The parameter templates assigned to the action template serve as input parameters for the policy action at enforcement time. You can configure the required parameters of an action template in two ways:

- *Default values:* You can define parameters with default values and select one of these values as the default value. When the action template and its associated parameter template are used in a policy, you can restrict the policy action to use only the defined default values.
- *Blank values:* Alternatively, you can define parameters with blank values. When the action template and its associated parameter template are used in a policy, the policy action accepts any desired value.

5 Who Can Create and Manage Action Categories or Templates?

To create and manage action templates for design/change-time policies, you must belong to a role that includes the "Manage System-wide Design/Change-Time Policies" permission. By default, the following predefined roles include the "Manage System-wide Design/Change-Time Policies" permission:

- CentraSite Administrator
- Asset Type Administrator
- Operations Administrator

For more information about permissions, see the section *About Roles and Permissions* in the document *Users, Groups, Roles, and Permissions*.

6 Viewing the Action Categories List

The **Action Templates** page displays the list of action categories and templates defined on your instance of CentraSite.

To view the action categories and templates list

1. In the CentraSite UI, go to **Policies** > **Action Templates**.
2. The action templates list provides the following information about a category or template.

Column	Description
Action Templates	Lists the action templates assigned to each category.
Description	Provides additional comments or descriptive information about an action template.
Type	Indicates whether an action category contains design-time or change-time or global or WS-I compliant action template and whether an action template type is manual or programmatic.

7 Adding Custom Actions Using CentraSite's UI

To add a custom action to CentraSite, perform the following high-level steps:

1. **Create a custom action category.**
   During this step, you specify the type of the category and details for the category. For more information, see *Creating a Custom Action Category*.
2. **Add an action template to the custom action category.**
   During this step, you specify details for the action template, upload its associated action rule (if appropriate), and select the object and event types to which this template applies. For more information, see *Adding an Action Template to a Custom Action Category*.
3. **Add a parameter template to the action template.**
   During this step, you add one or more parameter templates, and specify their input values. For more information, see *Adding a Parameter Template to the Action Template*.

Creating a Custom Action Category

Perform these steps to create a custom action category and save it to CentraSite.

To create a custom category

1 In CentraSite Control, go to **Policies** > **Action Templates**.

2 Click the **Add Action Category** button in the upper-right corner of the **Policy Information** panel.

3 In the **Add Action Category** dialog box, do the following:

   1. Specify a name for the new custom category.

      An action category name does not need to be unique within the CentraSite Registry. However, to reduce ambiguity, you should avoid giving multiple action categories the same name.

      An action category name can contain any character (including spaces).

   2. Choose the type of template that the category will contain (e.g., Design/Change-Time or Run-Time templates).
   3. Click **OK**.

Adding an Action Template to a Custom Action Category

Perform these steps to add an action template to a custom action category and save it to CentraSite.

To add an action template to a custom action category

1. In CentraSite Control, go to **Policies** > **Action Templates**.
2. Click **Add Action Template**.
3. If a custom action category does not yet exist, the **Add Action Category** dialog box is displayed, prompting you to create a custom action category now, as described in *Creating a Custom Action Category*. After you create a custom action category, the Add Action Template page is displayed, and is described in the next step.
4. In the Add Action Template page, do the following:

In this field...	Do the following...
Category	Select the action category for which you want to add the action template.
Name	Enter a name for the new action template. Follow these guidelines: ■ An action template name must be unique. ■ An action template name can contain any character (including spaces).
Description	*Optional*. Type a description for the new action template. This description appears when the user displays a list of action templates in the **Policy Information** panel.
Type	No action is necessary. By default, CentraSite sets the action type to **Programmatic** for a design-time or change-time action, and **Axiomatic** for a run-time action.
Implementation	For a programmatic action (a design-time or change-time action), specify whether the action's rule is a Groovy script or a Java class.
Uploaded File	For a programmatic action, click the **Browse** button and upload the action's rule file. ■ For a Java rule type, upload its Java program .zip file. ■ For a Groovy rule type, upload its .groovy script. For procedures on creating action rules, see *Creating Action Rules*. For procedures on uploading action rules, see *Uploading Action Rules to Action Templates*. **Note:** Alternatively, you can download the rules used by the system action templates, and modify them for use with the custom action template. For procedures, see *Downloading Rules from System Action Templates*.

5 In the **Scope** panel, do the following:

In this field...	Specify...
Object Types	Select the type of objects to which this action template applies.
Event Types	Select the type of events to which this action template applies. **Note:** Not all event types are supported by all objects. For more information, see *Supported Object Events*.

6 Click **Save**.

Upon saving the action template, CentraSite displays the Edit Action Template Detail page. You will use this page's **Parameter Templates** profile to add parameter templates for this action.

Adding a Parameter Template to the Action Template

To complete the action template, you must define its input parameters.

To add a parameter template to the action template

1 If you are beginning this procedure immediately after completing *Adding an Action Template to the Action Category*, skip to **step 4**.

2 In CentraSite Control, go to **Policies** > **Action Templates**.

3 Select the action template for which you want to define parameter templates.

4 In the Edit Action Template Detail page, select the **Parameter Templates** profile.

5 Click the **Add Parameter Template** button.

6 Define the first parameter as follows:

In this field...	Do the following...
Name	Enter a name for the new parameter template.
Type	Select a data type.
Default Value	If you want to specify a default value, type a value in this field. If the selected data type is String, Number or URL, you can specify one or multiple default values. You can specify *multiple* possible default data values from which to choose as follows: 1. Select the **Edit** icon to the right of the **Default Value** field. 2. In the **Add Default Values** dialog box, type a value and click **Add**. Repeat for as many values as you need.

In this field...	Do the following...
	3. Click **OK**. 4. Then, in the **Default Value** field select from the drop-down list the value you want to use as the default value. **Note:** If you would rather fill in required values when this template is used in a policy, leave this field blank.

7 If you need to define additional parameters, click the **Add Parameter Template** button again and repeat the previous step.

8 Click **Save** and then **Close**.

The parameter templates that you added appear under the **Parameter Templates** profile.

8 Adding Custom Actions Using APIs

To create a custom action programmatically, you perform the following high-level steps:

1. **Create a custom action category and template.**
   To do this, you create a Java class that uses the *com.centrasite.jaxr.CentraSiteLifeCycleManager* interface. To view the Javadoc for this interface, see *Javadoc: metamodelAPI*.
2. **Create a Java class action rule or a Groovy script action rule.**
   See ***Creating Action Rules***.
3. **Upload the action rule to the action template.**
   See ***Uploading Action Rules to Action Templates***.

Creating Action Rules

If creating a rule in a Java class:

1 Create a Java action executor class that implements the *com.softwareag.centrasite.policy.api.IActionExecutor* interface. To view the Javadoc for this interface, see *Javadoc: policyAPI*.

> **Important:** The Java executor class must return an AssertionResult object that contains the completion code `ResultStatus.SUCCESS` (if the action was successful) or `ResultStatus.FAILURE` (if the action failed). There are other possible completion codes (for example, `ResultStatus.IN_PROCESS`), however, these codes are used by internal processes and *are not* intended to be returned by user-defined actions. Custom actions that you create must only return a completion code of `ResultStatus.SUCCESS` or `ResultStatus.FAILURE`. For more information, see the Javadoc for the AssertionResult and ResultStatus classes.

2 Create a .zip file that contains the following:

- A folder named `lib`, which should contain a jar file with the action's executor class and the external libraries.
- A folder named `META-INF`, which should contain a property file named `assertion.properties`, which is the build file for the action. It includes an entry of the following format: *com.softwareag.centrasite.policy.rule.class=<fully_qualified_class_name>*

3 Upload the .zip file, as described in ***Uploading Action Rules to Action Templates***.

For example, the sample Java action rule provided in the your CentraSite installation has the following file structure:

- *<CentraSite_Install_Dir>*\demos\Custom actions\Java\META-INF\assertions.properties

This is the build file for the action. It includes an entry of the following format: *com.softwareag.centrasite.policy.rule.class=<fully_qualified_class_name>*.

- `<CentraSite_Install_Dir>`\demos\Custom actions\Java\src\com\softwareag\demo\actions\UniqueNameChecker.java

  This is the sample source file for the action executor. You can modify this file as needed, and compile it using the build file.

- `<CentraSite_Install_Dir>`\demos\Custom actions\Java\build.xml

  This build file has the default target "zip", which will compile the Java file, build a jar out of it and pack it as a zip file.

- `<CentraSite_Install_Dir>`\demos\Custom actions\Java\uniquenamechecker.zip

  This .zip file contains the following:

  - A folder named `lib`, which contains a jar file with the action's executor class and the external libraries.
  - A folder named `META-INF`, which contains the property file `assertion.properties, which is the build file for the action`.

For more information about the sample Java action, see ***Sample Custom Actions***.

If using a Groovy script rule:

- Upload the .groovy file as described in ***Uploading Action Rules to Action Templates***.

Uploading Action Rules to Action Templates

Before you upload a Java action rule, you must first create a .zip file as described in ***Creating Action Rules***.

If you want to upload a Groovy script rule, the .groovy file can contain the following variables. You cannot upload external libraries.

Variable	Description
`entity`	This represents the RegistryObject which is in the context. The following fields are available in this object: - Name - Description - State

Variable	Description
`policyContext`	The PolicyContext in which the policy is running.
`assertion`	The IAssertionInstance that is in execution.
`result`	The result object that has the status and message. Set the result.successMessage or result.failureMessage. Based on the message, the status is inferred. If you fail to set a message, the status will be treated as a successful execution with an empty message.

To upload an action rule

- There are two ways you can upload a rule's .zip file or .groovy file to a custom action template:

  1. If you are uploading the rule to a custom action template you created using the CentraSite UI, you upload the rule (either a Java .zip file or a .groovy file) when you create the action template, on the Add Action Template page. For procedures, see *Adding an Action Template to the Action Category*.

     —OR—

  2. If you are uploading the rule to a custom action template that you created programmatically, you upload the rule (either a Java .zip file or a .groovy file) using a JAXR-based call in the action template.

9 Viewing or Editing Action Categories or Templates

You use the Edit Action Category or Edit Action Template page to examine and/or edit the properties of an action category or action template. When viewing or changing the properties of an action category or template, keep the following points in mind:

- You cannot edit or delete the predefined categories or action templates that are installed with CentraSite.
- You can change any property of an custom category or action template; however cannot modify the type.
- You can edit an category or action template only after deactivating all the policies that use it.
- You can rename an action category at any time.

Viewing or Editing an Action Category

You use the following procedure to view or edit the action category details.

To view or edit the properties of an action category

1. In CentraSite Control, go to **Policies** > **Action Templates**.
2. In the **Policy information** panel, select the action category whose details you want to view or edit.
3. Examine or modify the category's properties on the **Edit Action Category** dialog box as appropriate. For more information about these properties, see *Creating a New Action Category*.

Viewing or Editing an Action Template

You use the following procedure to view or edit the action template details.

To view or edit the properties of an action template

1. In CentraSite Control, go to **Policies** > **Action Templates**.
2. In the **Policy information** panel, select the action template whose details you want to view or edit.
3. Examine or modify the template's properties on the Edit Action Template page as appropriate. For more information about these properties, see *Adding an Action Template to the Action Category*.
4. Select the **Scope** profile. View or edit the object types and event types to which this action template applies as appropriate. To modify the list of object or event types, do the following:

1. Click the **Select** button beside the list of applicable object or event types.
2. Use the controls in the Select Object/Event Types dialog box to adjust the list.
3. Click **Save** to update the modification.

Note: Not all event types are supported by all objects. See *Supported Object Events*.

5 Select the **Parameter Templates** profile. View or edit the parameter fields as appropriate.

6 If you have edited a template's properties, click **Save**. Otherwise, click **Cancel**.

10 Downloading Rules from System Action Templates

You can download the rules associated with CentraSite system action templates.

To download a rule

1. In CentraSite Control, go to **Policies** > **Action Templates**.
2. Locate the action template whose rule you want to download and select its name.
3. Choose the .zip file in the **Uploaded File** field, and then download the rule.

Structure of the Zip File

The structure of the zip file created by the download feature is as follows:

- A folder named lib, which contains a jar file with the action's executor class and the external libraries.
- A folder named META-INF, which contains a property file named `assertion.properties`, which is the build file for the action. It includes an entry of the following format: *com.softwareag.centrasite.policy.rule.class=<fully_qualified_class_name>*

11 Deleting Custom Action Categories and Templates

When you delete custom action categories, action templates and parameter templates, you must delete these items in the below order:

1. Parameter templates
2. Action templates
3. Action categories

Deleting a Parameter Template

Before you attempt to delete a parameter template, you must first delete all of the policies consuming it.

To delete a parameter template

1 In CentraSite Control, go to **Policies** > **Action Templates**.

2 Locate the custom action template whose parameter template you want to delete and select its name.

3 Select the **Parameter Templates** profile.

4 Select the check box beside the parameter template name and click **Delete**.

This temporarily revokes the selected parameter template from the action template.

5 Click **Save** to permanently remove the parameter template from the action template.

Deleting a Custom Action Template

Before you attempt to delete a custom action template, you must first delete all of the policies consuming it. Also, be aware that when you delete a custom action template, CentraSite also deletes all previous versions of the template.

To delete a custom action template

1 In CentraSite Control, go to **Policies** > **Action Templates**.

2 Locate the custom action template that you want to delete.

3 Select the check box beside the action template name and click **Delete**.

Deleting a Custom Action Category

Before you attempt to delete a custom action category, you must first delete all of the policies consuming it.

To delete a custom action category

1. In CentraSite Control, go to **Policies** > **Action Templates**.
2. Locate the custom action category that you want to delete.
3. Select the check box beside the action category name and click **Delete**.

12 Versioning a Custom Action Template

If you need to modify a custom action template, you can create a new version of the existing template and make your changes to the new version. When you create a new version of a custom action template, CentraSite creates an identical copy of the existing template, and then you make your changes to the copy. (Note that the new version of the custom action template will get is own copy of the executable Groovy or Java file.)

Be aware that CentraSite *does not* automatically apply the new custom action template to policies that use existing versions of the custom action. Policies that use existing versions of the action will continue to use the versions that they have. If you want to apply the new version of the action to these policies, you must edit the policies (or create new versions of them) and replace the old version of the action with the newer one.

Similarly, modifying the parameter definitions in a new version of an action template *will not* affect the parameter definitions in any of the existing policies that use the action. Parameter definitions are specific to a version of the template.

When you create a new version of a custom action, be aware that:

- You can only create a new version from the *latest version* of an action. For example, if an action already has versions 1.0, 2.0 and 3.0, CentraSite will only allow you to create a new version of the action from version 3.0.
- Initially, the new version of the action will be identical to the version from which you created it (except for the system-assigned version identifier, which is always incremented by one).
- CentraSite automatically establishes a relationship between the new version of the policy and the previous version. CentraSite uses this relationship to enforce rules related to versioned actions.
- You can only create new versions of *custom* actions that exist on your instance of CentraSite (i.e., actions that you have added to CentraSite). You cannot create new versions of the predefined actions that are installed with CentraSite.

To version a custom action

1. In CentraSite Control, go to **Policies > Action Templates** to display the list of action templates.
2. Locate the most recent version of the custom action for which you want to create a new version.
3. From the context menu for the custom action, click **Create New Version**.
4. Modify the new version of the custom action as necessary and then save it.

   Tip: To make the new version of the custom action easy to distinguish from earlier versions, consider appending the version number to the name of the custom action. This will make the versions easier to tell apart when you view or edit the action list for a policy.

13 Sample Custom Actions

Your CentraSite installation contains two sample custom action rules. One rule is a Java rule, and the other is a Groovy script rule.

The content is organized under the following sections:

Sample Java Action: Enforce Unique Asset Names

Your CentraSite installation contains a sample Java action rule (which is contained in unique-namechecker.zip) that you can use to create a custom action that ensures that the name and version combination of a newly-created asset is unique within the CentraSite catalog. If it is not unique, the action returns Failure, and the asset is not allowed to be created.

To create the custom action, you will use the CentraSite user interface to:

1. Create a custom action category
2. Create a custom action template, to which you upload the sample Java rule
3. Create a Design/Change-Time policy and add the custom action template to it

Create and test the custom action as follows.

To create and test the custom action Enforce Unique Asset Names

1 In CentraSite Control, go to **Policies** > **Action Templates**.

2 Click the **Add Action Category** button.

3 In the **Add Action Category** dialog box, do the following:

   1. Specify a name for the new custom category, for example `My Custom Actions`. An action category name can contain any character (including spaces).
   2. Choose **Design/Change-Time** as the action category type.
   3. Click **OK**.

   The action category that you created appears as a custom category next to an icon in the Policy Information panel.

4 Click **Add Action Template**.

5 On the Add Action Template page, specify the following fields:

In this field...	Do the following...
Category	Select the custom action category you just created.
Name	Enter the name `Enforce Unique Asset Names` for the new action template.
Description	*Optional*. Type a description for the new action template. For example: `Ensures that asset names are unique.`
Implementation	Select **Java**.
Uploaded File	Click the **Browse** button and upload the following rule file: *<CentraSite_Install_Dir>*\demos\Custom actions\Java\uniquenamechecker.zip This .zip file that contains the following: ■ A folder named `lib`, which contains a jar file with the action's executor class and the external libraries. ■ A folder named `META-INF`, which contains a property file named `assertion.properties`, which is the build file for the action. For more information about creating and uploading actions rules, see ***Creating Action Rules***.

6 In the **Scope** panel, specify the following fields:

In this field...	Do the following...
Object Types	Select **Service** as the type of object to which this action template applies.
Event Types	Select **PreCreate** as the type of event to which this action template applies.

7 Click **Save**.

The Edit Action Template Detail page is displayed.

8 Create a policy and add the sample action to the policy as follows:

1. In CentraSite Control, go to **Policies > Design/Change Time**.
2. Click **Add Policy**.
3. In the **Policy Information** panel, enter a name for the new policy, for example, `Ensure Unique Asset Names Policy`. A policy name can contain any character (including spaces).
4. In the **Scope** panel, specify the object and event types to which the policy applies as follows:
   - In the **Object Types** field, select **Service** as the type of object to which this policy applies.
   - In the **Event Types** field, select **PreCreate** as the type of event to which this policy applies.
   - In the **Organization** field, select your organization name as the organization to which this policy belongs (and to whose objects the policy will be applied).
5. Click **Next**.

6. From the **Available Actions** list, choose the custom action **Enforce Unique Asset Names** action that you created.
7. Click **Finish** to save the new policy. The Design/Change-Time Policy Details page is displayed.
8. Activate the policy by choosing the **Change State** button and choosing the **Productive** state.

Sample Groovy Script Action: Service Attribute Checker

Your CentraSite installation contains a sample Groovy action rule (ServiceAttributeChecker.groovy) that you can use to create a custom action that checks for a particular value of a service attribute.

To create and test the custom action, you will use the CentraSite user interface to:

1. Create a custom action category
2. Create a custom action template, to which you upload the sample Groovy script rule
3. Create a Design/Change-Time policy and add the custom action template to it
4. Test the custom action "on demand" (manually) on the policy's detail page

Create and test the custom action as follows.

To create and test the custom action Service Attribute Checker

1 In CentraSite Control, go to **Policies** > **Action Templates**.

2 Click the **Add Action Category** button.

3 In the **Add Action Category** dialog box, do the following:

   1. Specify a name for the new custom category, for example `My Custom Actions`. An action category name can contain any character (including spaces).
   2. Choose **Design/Change-Time** as the action category type.
   3. Click **OK**.

   The action category that you created appears as a custom category next to an icon in the Policy Information panel.

4 Click **Add Action Template**.

5 On the Add Action Template page, specify the following fields:

In this field...	Do the following...
Category	Select the custom action category you just created.
Name	Enter the name `Service Attribute Checker` for the new action template.
Description	*Optional*. Type a description for the new action template. For example: `Validates asset attribute values.`
Implementation	Select **Groovy**.
Uploaded File	Click the **Browse** button and upload the following rule file: *<CentraSite_Install_Dir>*\demos\Custom actions\Groovy\ServiceAttributeChecker.groovy

In the **Scope** panel, specify the following fields and click **Save**:

In this field...	Do the following...
Object Types	Select **Service** as the type of object to which this action template applies.
Event Types	Select **OnTrigger** as the type of event to which this action template applies. This will enable you to test the action "on demand" (manually) in the **Actions** profile of the Design/Change Time Policy Detail page.

6 In the Edit Action Template Detail page, select the **Parameter Templates** profile in order to add the action's parameter templates and click the **Add Parameter Template** button.

7 Set the first parameter template for the action as follows:

In this field...	Do the following...
Name	Enter `Attribute Name`. This parameter will hold the attribute that you want to check (e.g., any CentraSite attribute, which you will assign after you add the action to a policy).
Type	Choose the data type **Attribute** for this parameter template.
Default Value	Leave blank.
Array	Leave blank.
Required	Select this check box.

8 Click the **Add Parameter Template** button again to set the second parameter template as follows:

In this field...	Do the following...
Name	Enter `Possible Attribute Value`. This parameter will hold the value of the attribute that you want to check.
Type	Select the data type **String** for this parameter template.
Default Value	Enter an attribute value that you want to check.
Array	Leave blank.
Required	Select this check box.

9 Click **Save** and then **Close**.

The parameter template that you added appears in the **Parameter Templates** profile.

10 Create a policy and add the sample action to the policy as follows:

1. In CentraSite Control, go to **Policies > Design/Change Time**.
2. Click **Add Policy**.
3. In the **Policy Information** panel, enter a name for the new policy, for example, `Service Attribute Checker Policy`. A policy name can contain any character (including spaces).
4. In the **Scope** panel, specify the object and event types to which the policy applies as follows:
   - In the **Object Types** field, select **Service** as the type of object to which this policy applies.
   - In the **Event Types** field, select **OnTrigger** as the type of event to which this policy applies.
   - In the **Organization** field, select your organization name as the organization to which this policy belongs (and to whose objects the policy will be applied).
5. Click **Next**.
6. From the **Available Actions** list, choose the custom action **Service Attribute Checker** action that you created.
7. Click **Finish** to save the new (as yet incomplete) policy. The Design/Change-Time Policy Details page is displayed.
8. To configure the two parameters for the action, choose the action name on the **Actions** profile. For the **Attribute Name** parameter, choose an attribute. For the **Possible Attribute Value** parameter, select the default value that you defined for the **Possible Attribute Value** parameter in the action template.
9. Click **Save** and then **Close**. The policy detail page is displayed.
10. Activate the policy by choosing the **Change State** button and choosing the **Productive** state.
11. Select the **Actions** profile.
12. To test the action, click the **Run** button. The action checks all services in your organization and displays a pop-up. The pop-up should indicate Success in the Result column for the

service that matches the value you specified in `Possible Attribute Value`. All other services will also be displayed, with Failure in the Result column.

11 After you have completed testing, you can use this custom action in other Design/Change-Time policies. To do this, go to its Scope panel, change the event type from **OnTrigger** to **PreCreate** or **PreUpdate**, and include the action in other Design/Change-Time policies whose event type scope is either PreCreate or PreUpdate.