Oracle® Web Services Manager

Extensibility Guide

10*g* (10.1.3.1.0)

B31010-02

February 2007

ORACLE®

Oracle Web Services Manager Extensibility Guide, 10*g* (10.1.3.1.0)

B31010-02

Primary Author: Laureen Asato, Vrinda Kirloskar

Contents

Preface

This preface provides information on the following topics:

Audience

This manual provides information to programmers and system administrators on how to extend Oracle Web Services Manager (Oracle WSM) with custom policy steps.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

`http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documents

For more information, see the following documents in the Oracle Web Services Manager10*g* (10.1.3.1.0) documentation set:

- *Oracle Web Services Manager Administrator's Guide*
- *Oracle Web Services Manager Deployment Guide*
- *Oracle Web Services Manager Quick Start Guide*
- *Oracle Web Services Manager Installation Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Developing and Deploying Custom Steps

This chapter provides an overview of how to create custom policy steps, and it describes how to create, develop, and deploy custom policy steps.

This chapter contains the following topics:

Policy Step Creation Overview

Oracle Web Services Manager (Oracle WSM) provides the following tools for creating a custom step:

- The Oracle WSM SDK
- A sample step template for developing a custom step
- The Oracle WSM step template schema

Complete the following tasks to develop a custom step:

- Define a step template and provide a unique ID, name, and configurable parameters for the step.
- Extend the AbstractStep class by writing the implementation of the init, execute, and destroy methods, and adding the set and get methods for the step properties.

> **Note:** The execute method is the only required method for implementing a custom step.

- Deploy the custom step.

Ensure that the custom steps you develop are compliant with Web services standards.

Developing a Custom Step

A custom step extends the `com.cfluent.policysteps.sdk.AbstractStep` class and implements the execute method. The following section contains information about the execute method.

About the execute Method

The basic characteristics and functionality of the execute method are as follows:

- Provides the program entry point for the custom step
- Can throw an exception of type com.cfluent.policysteps.sdk

  If a fault is thrown by the execute method during exception processing, this fault is cascaded back to the client application.
- Returns an object of type IResult

About IResult States

The custom step generates the result object using the createResult method. The int argument for the createResult method provides the states of the result. These states indicate the following about the execute method:

- IResult.FAILED: The execute method failed.
- IResult.SUCCEEDED: The execute method succeeded.
- IResult.SUSPENDED: The execute method suspended execution.

> **Note:** Because the execute method is multithreaded (that is, it can be executed by multiple threads simultaneously), you should ensure that the implementation of the execute method is thread safe.

To override the default init method, add initialization code specific to that custom step. You can also add code to the destroy method to ensure a clean end to the lifecycle of the custom step.

The AbstractStep interface provides the following default methods:

```
public final void setEnabled(boolean);
public final boolean getEnabled();
public final void setStepName(java.lang.String);
public final java.lang.String getStepName();
public final com.cfluent.policysteps.sdk.AgentContext getAgentContext();
public final void setAgentContext(com.cfluent.policysteps.sdk.AgentContext);
public void init() throws java.lang.IllegalStateException;
protected void generateFault(com.cfluent.policysteps.sdk.Fault) throws
com.cfluent.policysteps.sdk.Fault;
protected com.cfluent.policysteps.sdk.IResult createResult(int);
```

Defining a Step Template

The Oracle WSM policy steps framework requires that each Oracle WSM step refer to a step template. The step template defines the step ID and the step properties, and provides a brief description of each property.

To add a step template to the Oracle WSM framework, you must create a well-formed XML document. See "Sample Step Template" on page A-1, when creating your XML sample step template. See "Step Template Schema" on page A-2, for the step template schema.

The following table contains descriptions of a set of tags for a typical step template:

Table 1–1 Step Template tags

Tags	Description
csw:StepTemplate name=Step_Name id=XYZA123	Start tag of the XML document that defines the step name and a unique ID. The Oracle WSM system does not assign a unique ID—you must assign one. If the ID you assign is not unique to the system, Oracle WSM will throw an exception.
csw:Description	Provides a brief description of the step.
Csw:Implementation	Identifies which class contains the implementation of the step. The Framework loads the class defined by this step when the step is invoked, at runtime, by reflection.
Csw:Property name=?...? type=?...?	Defines the properties available for the name and the data type of a step property. Data types can be the following: boolean, string, boolean, int, long, float, short, string[], boolean[], int[], long[], short[], string[]
Property/Csw:Description	Provides a brief description of the property defined by the tag above. A Step developer is expected to provide a relevant description of the property being defined.
Property/csw:Value/Absolute	Defines the default value(s) for the property being defined. The value may be Absolute or PropertyRef. A PropertyRef indicates to the Framework to carry over an environmental property, defined by ${<property_key>}.

Deploying a Custom Step

Complete the following tasks to deploy a custom step:

- Add the custom step to the Oracle WSM Policy Manager.
- Add the jar file containing the implementation classes for the step to the /lib directory. See "To add a jar file" on page 1-4 for more information about adding or copying the jar file.

> **Note:** When you copy a new jar file, or make changes to an existing jar file, you must restart the Oracle Application Server.

- Add the custom step to the service pipeline as needed.

To add a custom step

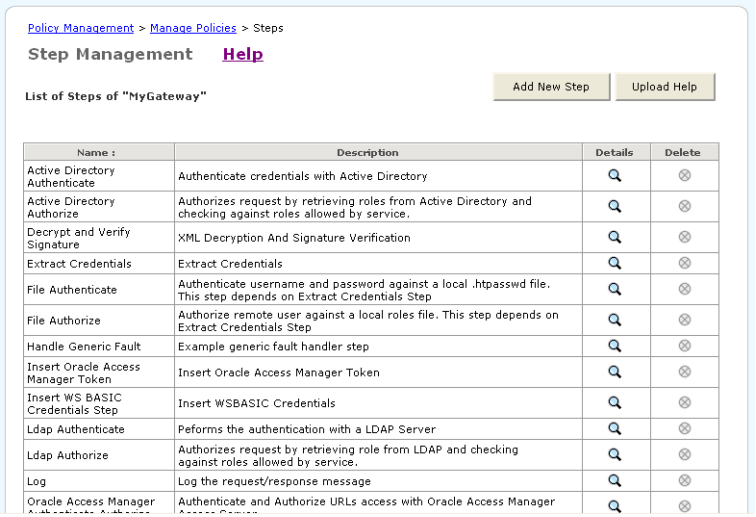
1. Log in to the Web Services Manager Control, and select the **Policy Manager** from the navigation bar.

   The Web Services Manager Control displays the list of currently defined components.

2. In the row for the component where you want to add a custom step, click **Steps**.

   The **Step Management** window displays a list of existing steps (Figure 1–1).

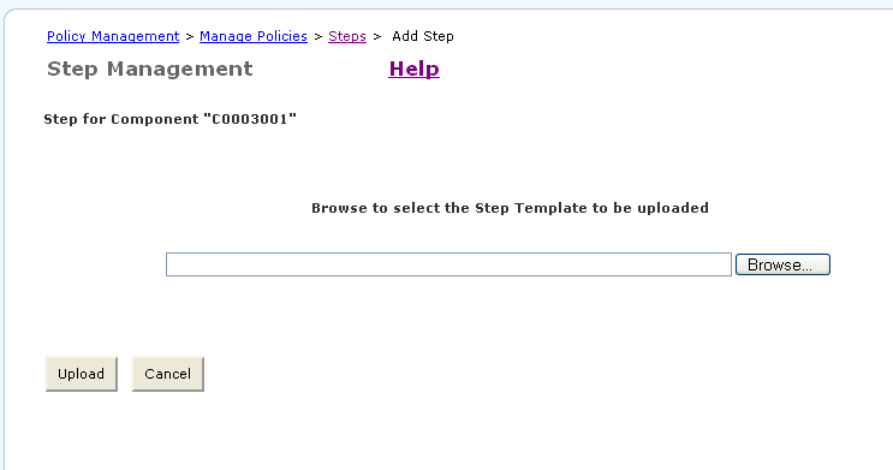
Figure 1–1 List of Policy Steps



3. Click **Add New Step**.

   The Web Services Manager Control displays the Step Management/Add Step page (Figure 1–2).

Figure 1–2 Step Management/ Add Step Page



4. Click **Browse** and select the newly created XML step template.
5. Click **Upload**.

   The **Step Management** screen updates the list of currently available steps, including the newly added step.

> **Note:** Each custom step shows a delete button; however, default steps cannot be deleted.

To add a jar file

You need to copy the Java Archive (jar) containing the implementation classes for the custom step to the machine where the gateway or agent component is installed. Copy the jar file to the following location:

ORACLE_HOME/owsm/lib/custom

To add a step to a policy enforcement point

1. From Web Services Manager Control, select **Policy Manager**.

   Web Services Manager Control displays the list of registered Oracle WSM components.

2. Select the component whose policy you want to modify, and click **Policies**.

   Web Services Manager Control displays the default policy for the component.

3. In the policy row where you want to add a step, click **Edit**.

   The page displays the existing policy pipeline.

4. In the section of the pipeline immediately above where you want to add a step, click **Add Step Below**.

   A list displays the steps available for the selected component, including your new custom step.

5. Select the new step and click **OK**.

6. Configure the step.

7. Click **Next**.

   The step is added to the Service Pipeline.

8. Click **Save**.

9. Commit the changes to the policy by clicking **Commit Policy**.

Note: The new policies go into effect in 10 seconds if you use the default polling frequency. The gateway and agent components automatically retrieve the updated policies from the Oracle WSM Policy Manager.

2

Policy Steps SDK Interfaces & Methods

This chapter contains reference information for the Oracle Web Services Manager (Oracle WSM) Step SDK. The Oracle WSM Step SDK contains the following interfaces and methods:

- IStep
- IContext
- IMessageContext
- AgentContext
- AbstractStep
- Result
- InvocationStatus

IStep

Interface IStep

Known Implementing Classes

AbstractStep

Declaration

```
public interface IStep
```

Description

A step is a basic execution block in the pipelines. A step is a Java Bean that is configured using the Java Beans framework. The AbstractStep class is an abstract class that provides partial implementation of this interface. A step may use resource pooling mechanisms to efficiently handle multiple requests, but it should not keep any per-invocation state.

Table 2–1 Methods of IStep

Method	Description
setEnabled()	Enables or disables this step.
getEnabled()	Gets the boolean value from setEnabled.
setStepName()	Sets the name of this step.
getStepName()	Gets the step name.

Table 2–1 (Cont.) Methods of IStep

Method	Description
getAgentContext()	Gets context information for the policy enforcement point.
setAgentContext()	Sets the agent context.
execute()	Executes this step.
init()	This method is called after the bean is created or after a bean property has changed.
destroy()	Ensures a clean ending of the step lifecycle.
getFaultCodes()	Gets the fault codes that are thrown by this step.
setFaultCodes()	Sets the fault codes that are thrown by this step.

setEnabled()

Declaration

```
public void setEnabled(boolean enabled)
```

Description

Enables or disables this step.

Parameters

Enabled: If true, the step is enabled.

getEnabled()

Declaration

```
public boolean getEnabled()
```

Description

Gets the boolean value from setEnabled.

Returns

One of the boolean values: true or false. If the step is enabled, this method returns the value, true.

setStepName()

Declaration

```
public void setStepName(java.lang.String stepName)
```

Description

Sets the name of this step.

Parameters

stepName: Name of the step.

getStepName()

Declaration

```
public java.lang.String getStepName()
```

Description

Gets the step name.

Returns

The name of this step.

getAgentContext()

Declaration

```
public AgentContext getAgentContext()
```

Description

Gets context information for policy enforcement point.

setAgentContext()

Declaration

```
public void setAgentContext(AgentContext context)
```

Description

Sets the agent context.

Parameters

context (of type AgentContext): Represents the information of an agent during the pipeline execution.

execute()

Declaration

```
public IResult execute(IMessageContext context)
     Throws Fault
```

Description

Executes this step.

Parameters

context: message context containing the request and response.

Returns

IResult: on successful execution of this method.

Fault: Fault type exception, if it gets thrown.

init()

Declaration

```
public void init()
     throws java.lang.IllegalStateException
```

Description

This method is called after the bean is created or after a bean property has changed.

Throws

java.lang.IllegalStateException

destroy()

Declaration

```
public void destroy()
     throws java.lang.IllegalStateException
```

Description

Use the destroy method to ensure a clean ending of the step lifecycle.

setFaultCodes()

Declaration

```
void setFaultCodes(String[]faultCodes)
```

Description

Set the fault codes that are thrown by this step.

Parameters

faultCodes (of type String[]): Fault codes thrown by the step.

getFaultCodes()

Declaration

```
String[]getFaultCodes()
```

Description

Get the fault codes that are thrown by this step.

Returns

The string array of the fault codes thrown by this step.

IContext

Interface IContext

Subinterfaces

IMessageContext

Declaration

```
public interface IContext
```

Description

Information about the pipeline execution context with regard to a component.

Table 2–2 Methods of IContext()

Method	Description
getProperty()	Searches for a specific property and returns the value of the property.
getPropertyNames()	Searches for a specific property or set of properties and returns the name or names of those properties.
containsProperty()	Checks whether a specified property (propName) exists.
setProperty()	Sets the value of a specified property (propName).
removeProperty()	Removes a specified property (propName).

getProperty()

Declaration

```
public java.lang.Object getProperty(java.lang.String propName)
```

Description

Searches for a specific property and returns the value of that property.

Returns

The value for propName.

getPropertyNames()

Declaration

```
public java.util.Iterator getPropertyNames()
```

Description

Searches for a specific property or set of properties and returns the name or names.

containsProperty()

Declaration

```
public boolean containsProperty(java.lang.String propName)
```

Description

Checks for the presence of a specific property.

Returns

One of the boolean values: true or false.

setProperty()

Declaration

```
public void setProperty(java.lang.String propName,
java.lang.Object obj,
boolean persistent)
```

Parameters

propName, obj, persistent.

Description

Sets a property to a specified value.

removeProperty()

Declaration

```
public void removeProperty(java.lang.String propName)
```

Parameters

propName

Description

Removes a specific property.

IMessageContext

Interface IMessageContext

Declaration

```
public interface IMessageContext
extends IContext
```

Description

Provides contextual information about messages that the step handles.

All Superinterfaces

IContext

Inheritance from interface com.cfluent.policysteps.sdk.IContext

containsProperty, getProperty, getPropertyNames, removeProperty, setProperty, setProperty.

Table 2–3 IMessageContext() Static Fields

Field	Description
STAGE_PREREQUEST	Indicates the prerequest processing stage.
STAGE_REQUEST	Indicates the request processing stage.
STAGE_RESPONSE	Indicates the response processing stage.
STAGE_POSTRESPONSE	Indicates the postresponse processing stage.

Table 2–3 (Cont.) IMessageContext() Static Fields

Field	Description
STAGE_SERVICE	Indicates that the message is at service.
STAGE_SERVICE_ DEFINITION	Provides a service definition.
STAGE_SERVICE_WSDL	Identifies the service WSDL.

Table 2–4 Methods of IMessageContext()

Method	Description
getGUID()	Gets the global unique ID.
getServiceID()	Identifies the service to where the message is forwarded.
getService URL()	Gets the remote user, posted with credentials.
getRemoteUser()	Gets the ID of the service requestor.
getRequestMessage()	Gets the SOAP request.
getResponseMessage()	Gets the SOAP response.
setRequestMessage(*)	Sets the request message.
setResponseMessage()	Sets the response message.
getProcessingStage()	Gets the processing stage information.
setProcessingStage()	Sets the processing stage.
getInvocationStatus()	Gets the invocation status.
setUserLocale()	Sets the user locale. The protocol handlers use this method to look for locale information in the message properties and message headers. If the locale information is found, the locale is set. If the locale information is not found, the user locale defaults to the product locale.
getUserLocale()	Gets the user locale. The policy step implementation code uses the user locale to provide the SOAP faults in the user's specified locale.

MessageContext() Fields

STAGE_PREREQUEST

Declaration

```
public static final java.lang.String STAGE_PREREQUEST
```

Description

Indicates the prerequest processing stage.

STAGE_REQUEST

Declaration

```
public static final java.lang.String STAGE_REQUEST
```

Description
Indicates the request processing stage.

STAGE_RESPONSE

Declaration
```
public static final java.lang.String STAGE_RESPONSE
```

Description
Indicates the response processing stage.

STAGE_POSTRESPONSE

Declaration
```
public static final java.lang.String STAGE_POSTRESPONSE
```

Description
Indicates the postresponse processing stage.

STAGE_SERVICE

Declaration
```
public static final java.lang.String STAGE_SERVICE
```

Description
Indicates that the message is at the service.

STAGE_SERVICE_DEFINITION

Declaration
```
public static final java.lang.String STAGE_SERVICE_DEFINITION
```

Description
Provides a service definition.

STAGE_SERVICE_WSDL

Declaration
```
public static final java.lang.String STAGE_SERVICE_WSDL
```

Description
Identifies the service WSDL.

getGUID()

Declaration
```
public java.lang.String getGUID()
```

Description
Gets the global unique ID.

Returns
The global unique ID.

getServiceID()

Declaration
```
public java.lang.String getServiceID()
```

Description
Identifies the service to where the message is forwarded.

Returns
The ID of the service based on the service registration information in Oracle WSM.

getServiceURL()

Declaration
```
public java.lang.String getRemoteUser()
```

Description
Gets the service URL.

Returns
The service URL for this pipeline.

getRemoteUser()

Declaration
```
public java.lang.String getRemoteUser()
```

Description
Gets the ID of the service requestor.

Returns
The ID of the service requestor.

getRequestMessage()

Declaration
```
public com.cfluent.ccore.message.SOAPMessage getRequestMessage()
```

Description
Gets the SOAP request.

Returns
The SOAP request message.

getResponseMessage()

Declaration
```
public com.cfluent.ccore.message.SOAPMessage getResponseMessage()
```

Description
Gets the SOAP response.

Returns
The SOAP response message.

setRequestMessage(*)

Declaration
```
public void setRequestMessage(com.cfluent.ccore.message.SOAPMessage
requestMessage)
```

Description
Sets the request message.

setResponseMessage()

Declaration
```
public void setResponseMessage(com.cfluent.ccore.message.SOAPMessage
responseMessage)
```

Description
Sets the response message.

getProcessingStage()

Declaration
```
public java.lang.String getProcessingStage()
```

Description
Gets the processing stage information.

Returns
The information from the processing stage.

setProcessingStage()

Declaration
```
public void setProcessingStage(java.lang.String stage)
```

Description
Sets the processing stage.

getInvocationStatus()

Declaration
```
public InvocationStatus getInvocationStatus()
```

Description
Gets the invocation status.

Returns
The invocation status.

AgentContext

Declaration
```
public class AgentContext
```

extends
java.lang.Object

Properties
agentID, resolver, properties.

Description
Provides methods to retrieve and work with agent context information.

Constructor
public AgentContext(java.lang.String agentId,

com.cfluent.common.resource.IResourceResolver resolver,

java.util.Map properties)

throws java.io.IOException

Table 2–5 AgentContext Static Fields

Field	Description
SERVER_ID	Component ID.
LOG_LOGBUNDLES	Bundles for the log.
LOG_LOGENABLED	Specifies whether logging is enabled.
COREMAN_ENABLED	Specifies whether the Oracle WSM Monitor is enabled.
POLICYSERVER_ ENDPOINT	Specifies the Oracle WSM Policy Server endpoint.
POLICYSERVER_ ENABLED	Specifies that policy server is enabled.
POLICYPACKS_ FILENAME	Policy packs filename. [Internal code structure filename]

Table 2–6 Methods of AgentContext

Method	Description
getAgentID()	Gets the agent ID as a string.
getProperty()	Gets the property value for a specified propName.
getStringProperty()	Gets the value of the specified property. Returns null if the property does not exist.
getIntProperty()	Gets a property as an integer (primitive).
getResourceResolver()	Gets the resolver to a resource. The resolver is used to get the absolute path to the resource.
getAllProperties()	Gets all the properties as a key-value pair.

AgentContext Fields

SERVER_ID

Declaration

```
public static java.lang.String SERVER_ID
```

Description

Value for the component ID.

LOG_LOGBUNDLES

Declaration

```
public static java.lang.String LOG_LOGBUNDLES
```

Description

Bundles for the log.

LOG_LOGENABLED

Declaration

```
public static final java.lang.String LOG_LOGENABLED
```

Description

Property if logging is enabled.

COREMAN_ENABLED

Declaration

```
public static final java.lang.String COREMAN_ENABLED
```

Description

Property that specifies that the Oracle WSM Monitor is enabled.

POLICYSERVER_ENDPOINT

Declaration

```
public static final java.lang.String POLICYSERVER_ENDPOINT
```

Description

Property for Oracle WSM Policy Server endpoint.

POLICYSERVER_ENABLED

Declaration

```
public static final java.lang.String POLICYSERVER_ENABLED
```

Description

Specifies that Oracle WSM Policy Server is enabled.

POLICYPACKS_FILENAME

Declaration

```
public static final java.lang.String POLICYPACKS_FILENAME
```

Description

Property for policy packs filename.

getAgentID()

Declaration

```
public java.lang.String getAgentID()
```

Description

Gets the agent ID. The agent ID is configured in the config.xml file and is loaded by the constructor as a property during initialization.

Returns

The agent ID as a string.

getProperty()

Declaration

```
public java.lang.Object getProperty(java.lang.String propName)
```

Description

Gets the property value for a specified propName.

Parameters

propName: Name of the property for which the value needs to be retrieved.

Returns

The value of specified property. It returns a null if the property does not exist.

getStringProperty()

Declaration

```
public java.lang.String getStringProperty(java.lang.String propName,
java.lang.String defaultValue)
```

Description

Gets the property value as a string.

Parameters

propName: Name of the property for which the value needs to be retrieved.

Returns

The string representation of the value of the property in argument. It returns a null if the property is not available.

getIntProperty()

Declaration

```
public int getIntProperty(java.lang.String propName)
```

Description

Gets a property as an integer (primitive).

Parameters

propName: Name of the property for which the value needs to be retrieved.

Returns

An integer (primitive) value of the property name in the argument.

getIntProperty()

Declaration

```
public int getIntProperty(java.lang.String propName,
java.lang.string defaultValue)
```

Description

Fetches the value of a property in the argument.

Parameters

propName: Name of the property for which the value needs to be retrieved.

defaultValue: Default value that is returned if the property is not available.

Returns

The value of the property in the argument. It returns the value if the property exists; returns the default value in the argument, if the property is not available.

getResourceResolver()

Declaration

```
IResourceResolver getResourceResolver()
```

Description

Gets the resolver to a resource. The resolver is used to get the absolute path to the resource.

Returns

The object of type IResourceResolver.

getAllProperties()

Declaration

```
public java.util.Map getAllProperties()
```

Description

Gets all the properties as a key-value pair.

Returns

Then object of type Map, containing key-value pairs of the properties.

AbstractStep

com.cfluent.policysteps.sdk.AbstractStep

Declaration

```
public abstract class AbstractStep
```

extends

java.lang.Object

implements

IStep

Description

A default implementation of IStep. The implementation should at least override the IStep execute method.

Table 2–7 Methods of AbstractStep

Method	Description
createResult()	Called after the bean is created or after a bean property has changed.
setEnabled()	If set to true, the step is enabled.
getEnabled()	Returns the value of setEnabled; true if the step is enabled, otherwise false.
setStepName()	Sets the name of this step.

Table 2–7 (Cont.) Methods of AbstractStep

Method	Description
getStepName()	Gets the step name.
getAgentContext()	getAgentContext in interface IStep
setAgentContext()	Sets the agent context.
execute()	Throws FaultExecute for this step. Step implementations should override this method to perform desired tasks.
init()	Throws java.lang.IllegalStateException. This method is called after the bean is created or after a bean property has changed.
getFaultCodes()	Gets the fault codes thrown by this step.
setFaultCodes()	Sets the fault codes thrown by this step.
destroy()	Ensures a clean ending to the step lifecycle.
generateFault()	Generates faults for this step.
createResult()	Creates the result for this step.

createResult()

Declaration

```
protected IResult createResult(int status)
```

Parameters

int status

Description

This method is called after the bean is created or after a bean property has changed.

Table 2–8 Fields for createResult()

Variable	Description
m_name	Protected java.lang.String m_name.
m_enabled	Protected boolean m_enabled.
m_context	Protected AgentContext m_context.

setEnabled()

Declaration

```
public final void setEnabled(boolean enabled)
```

Description

Enables or disables this step.

Parameters

boolean enabled

Specified by

setEnabled in interface IStep.

Parameters
enabled: If true, the step is enabled.

getEnabled()

Declaration
```
public final boolean getEnabled()
```

Description
Checks to see if the step is enabled.

Specified by
getEnabled in interface IStep.

Returns
One of the boolean values: true or false. If the step is enabled, this method returns the value, true

setStepName()

Declaration
```
public final void setStepName(java.lang.String stepName)
```

Description
Sets the name of this step.

Specified by
setStepName in interface IStep.

Parameters
stepName: Name of the step.

getStepName()

Declaration
```
public final java.lang.String getStepName()
```

Description
Gets the step name.

Specified by
getStepName in interface IStep.

Returns
The name of this step.

getAgentContext()

Declaration

```
public final AgentContext getAgentContext()
```

Description

Retrieves the agent context for the step.

Parameters

context

Specified by

getAgentContext in interface IStep.

setAgentContext()

Declaration

```
public final void setAgentContext(AgentContext context)
```

Description

Sets the agent context.

Specified by

setAgentContext in interface IStep.

execute()

Declaration

```
public abstract IResult execute(IMessageContext messageContext)
```

Description

Throws a FaultExecute for this step. Step implementations should override this method to perform desired tasks.

Specified by

execute in interface IStep.

Parameters

messageContext: Message context containing the request and response.

Returns

Depending on result of execution, can return IResult or Fault.

init()

Declaration

```
public void init()
```

Description
Throws java.lang.IllegalStateException. This method is called after the bean is created or after a bean property has changed.

Specified by
init in interface IStep.

Throws
java.lang.IllegalStateException.

See Also
IManageable

generateFault()

Declaration
```
protected void generateFault(Fault fault)
```

Description
Generates faults for this step.

Throws
Fault

destroy()

Declaration
```
public void destroy()
     throws java.lang.IllegalStateException
```

Description
Use the destroy method to ensure a clean end of the step lifecycle.

Specified By
init in interface IStep

setFaultCodes()

Declaration
```
void setFaultCoides(String[] faultCodes)
```

Description
Sets the fault codes thrown by this step.

Specified By
init in interface IStep

Parameters
faultCodes (of type String[]): Fault codes thrown by the step.

getFaultCodes()

Declaration

```
String[] getFaultCodes()
```

Description

Gets the fault codes thrown by this step.

Specified By

init in interface IStep.

Returns

A string array of the fault codes thrown by this step.

createResult()

Declaration

```
protected IResult createResult(int status)
```

Description

Creates the result for this step.

Result

Declaration

```
public class Result()
```

extends

java.lang.Object

implements

IResult

Methods Inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll

Constructor

public Result()

Fields Inherited from interface com.cfluent.policysteps.sdk.IResult

FAILED, SUCCEEDED, SUSPENDED

Table 2–9 Methods of Result

Method	Description
getStatus()	Gets the execution status.
setStatus()	Set the execution status.
getFault()	Returns the fault that caused the request execution failure.

Table 2–9 (Cont.) Methods of Result

Method	Description
setFault()	Sets the fault that caused this request execution to fail. The framework sets this fault if all the applicable fault handlers have failed. Step implementations should only call the generateFault method.
toString()	Result in string form

getStatus()

Declaration

```
public int getStatus()
```

Description

Gets the execution status.

Specified by

getStatus in interface IResult.

Returns

The execution status.

setStatus()

Declaration

```
public void setStatus(int status)
```

Description

Sets the execution status.

Specified by

setStatus in interface IResult.

Parameters

status: Sets the execution status.

getFault()

Declaration

```
public Fault getFault()
```

Description

The fault that caused the request execution failure. The framework sets the fault if all the applicable fault handlers have failed.

Specified by

getFault in interface IResult.

Returns
The fault that caused the request failure.

setFault()

Declaration
```
public void setFault(Fault fault)
```

Description
Sets the fault that caused the request execution failure. The framework sets the fault if all the applicable fault handlers have failed. Step implementations should only call the generateFault method.

Specified by
setFault in interface IResult

Parameters
fault

toString()

Declaration
```
public java.lang.String toString()
```

Overrides
toString in class java.lang.Object

Returns
The result in string format.

InvocationStatus

Declaration
```
public class InvocationStatus
```

extends
java.lang.Object

Implements
java.io.Serializable

Inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll

Description
This class sends step execution status to the Oracle WSM Monitor. The Oracle WSM Monitor uses this information to interpret measurements.

Constructor

public InvocationStatus()

Table 2–10 Fields of InvocationStatus

Field	Description
SUCCEEDED	Invocation Succeeded flag.
FAILED	Invocation failure flag.
FAILEDOVER	Failed Over flag.
PENDING	Measurement or execution pending.
NA	Object not available.

Table 2–11 Methods of InvocationStatus

Method	Description
getServiceName()	Gets the name of the service being invoked.
setServiceName()	Sets the service name.
getTime()	Gets the time.
setTime()	Sets the time.
getAuthenticationStatus()	Gets the authentication status.
setAuthenticationStatus()	Sets the authentication status.
getAuthorizationStatus()	Gets the authorization status.
setAuthorizationStatus()	Sets the authorization status.
getServiceStatus()	Gets the service status.
setServiceStatus()	Sets the service status.
getInvocationStatus()	Gets the invocation status.
setInvocationStatus()	Sets the invocation status.
getSize()	Gets the size.
setSize()	Sets the size.
getLatency()	Gets the default value for overall latency for policy pipelines.
setLatency()	Sets the value for overall latency.
getServiceLatency()	Gets the value for service latency in milliseconds.
setServiceLatency()	Sets the value for service latency in milliseconds.
getFlowID()	Gets the flow ID.
setFlowID()	Sets the flow ID.
getMethod()	Gets the method.
setMethod()	Sets the method.
getCorrelationContext()	Gets the correlation.
setCorrelationContext()	Sets the correlation.
getMessageId()	Gets the message ID.
setMessageId()	Sets the message ID.

Table 2–11 (Cont.) Methods of InvocationStatus

Method	Description
getErrorMessage()	Gets the error message for the failure.
setErrorMessage()	Sets the error message on failure.
toString()	String representation of the object.

InvocationStatus() Fields

SUCCEEDED

Declaration

```
public static final int SUCCEEDED
```

Description

Invocation succeeded flag.

FAILED

Declaration

```
public static final int FAILED
```

Description

Invocation failure flag.

FAILEDOVER

Declaration

```
public static final int FAILEDOVER
```

Description

Failed Over flag.

PENDING

Declaration

```
public static final int PENDING
```

Description

Measurement or execution pending.

NA

Declaration

```
public static final int NA
```

Description

Object not available.

getServiceName()

Declaration

```
public java.lang.String getServiceName()
```

Description

Gets the name of the service being invoked.

Returns

The name of the service being invoked.

setServiceName()

Declaration

```
public void setServiceName(java.lang.String name)
```

Description

Sets the service name.

getTime()

Declaration

```
public long getTime()
```

Description

Gets the time.

setTime()

Declaration

```
public void setTime(long time)
```

Description

Sets the time.

Parameters

time

getAuthenticationStatus()

Declaration

```
public int getAuthenticationStatus()
```

Description

Gets the authentication status.

setAuthenticationStatus()

Declaration

```
public void setAuthenticationStatus(int s)
```

Description

Sets the authentication status.

Parameters

s

getAuthorizationStatus()

Declaration

```
public int getAuthorizationStatus()
```

Description

Gets the authorization status.

Returns

Value for authorization status.

setAuthorizationStatus()

Declaration

```
public void setAuthorizationStatus(int s)
```

Description

Sets the authorization status.

Parameters

s

getServiceStatus()

Declaration

```
public int getServiceStatus()
```

Description

Gets the service status.

Returns

The value of service status.

setServiceStatus()

Declaration

```
public void setServiceStatus(int s)
```

Description
Sets the service status.

Parameters
s

getInvocationStatus()

Declaration
```
public int getInvocationStatus()
```

Description
Gets the invocation status.

Returns
The value for invocation status.

setInvocationStatus()

Declaration
```
public void setInvocationStatus(int s)
```

Description
Sets the invocation status.

Parameters
s

getSize()

Declaration
```
public int getSize()
```

Description
Gets the size.

Returns
The value of size.

setSize()

Declaration
```
public void setSize(int s)
```

Description
Sets the size.

Parameters
s

getLatency()

Declaration
```
public int getLatency()
```

Description
Gets the default value for overall latency used for policy pipelines.

Returns
The value for overall latency.

setLatency()

Declaration
```
public void setLatency(int l)
```

Description
Sets the value for overall latency.

Parameters
int l

getServiceLatency()

Declaration
```
public int getServiceLatency()
```

Description
Gets the service only latency.

Returns
The value for service latency (in milliseconds).

See Also
getLatency()

setServiceLatency()

Declaration
```
public void setServiceLatency(int l)
```

Description
Sets the value for service latency (in milliseconds).

getFlowID()

Declaration

```
public java.lang.String getFlowID()
```

Description

Gets the flow ID.

Returns

The value for flow ID.

setFlowID()

Declaration

```
public void setFlowID(java.lang.String flowid)
```

Description

Sets the flow ID.

Parameters

flowid (Flow ID passed)

getMethod()

Declaration

```
public java.lang.String getMethod()
```

Description

Finds and returns the method on the service.

Returns

The method on the service.

setMethod()

Declaration

```
public void setMethod(java.lang.String inMethod)
```

Description

Sets the method.

Parameters

inMethod

getCorrelationContext()

Declaration

```
public java.lang.String getCorrelationContext()
```

Description
Gets the correlation.

setCorrelationContext()

Declaration
```
public void setCorrelationContext(java.lang.String inCorrelationContext)
```

Description
Sets the correlation.

Parameters
inCorrelationContext

getMessageId()

Declaration
```
public java.lang.String getMessageId()
```

Description
Gets the message ID.

setMessageId()

Declaration
```
public void setMessageId(java.lang.String inMessageId)
```

Description
Sets the message ID.

Parameters
inMessageId

getErrorMessage()

Declaration
```
public java.lang.String getErrorMessage()
```

Description
Gets the associated error message for the failure.

setErrorMessage()

Declaration
```
public void setErrorMessage(java.lang.String m)
```

Description
Sets the error message on failure.

Parameters

m

toString()

Declaration

```
public java.lang.String toString()
```

Description

String representation of this object.

Overrides

toString in class java.lang.Object

A

Samples and Information Sources

This appendix contains an Oracle Web Services Manager (Oracle WSM) sample step template, and the step template schema.

Sample Step Template

The following is a sample step template:

```
<csw:StepTemplate xmlns:csw="http://schemas.confluentsw.com/ws/2004/07/policy"
name="Custom authenticate step" package="customsteps" timestamp="Oct 31, 2005
05:00:00 PM" version="1" id="118970809">
  <csw:Description>Custom step that authenticates the user against the credentials
entered here.</csw:Description>
  <csw:Implementation>customsteps.CustomAuthenticationStep</csw:Implementation>
    <csw:PropertyDefinitions>
      <csw:PropertyDefinitionSet name="Basic Properties">
         <csw:PropertyDefinition name="Enabled" type="boolean">
            <csw:Description>If set to true, this step is
enabled</csw:Description>
            <csw:DefaultValue>
              <csw:Absolute>true</csw:Absolute>
            </csw:DefaultValue>
       </csw:PropertyDefinition>
    </csw:PropertyDefinitionSet>
    <csw:PropertyDefinitionSet name="User Credentials">
     <csw:PropertyDefinition name="Username" type="string" isRequired="true">
  <csw:DisplayName>Username</csw:DisplayName>
       <csw:Description>Username used for authentication</csw:Description>
       <csw:DefaultValue>
          <csw:Absolute>test</csw:Absolute>
       </csw:DefaultValue>
     </csw:PropertyDefinition>
     <csw:PropertyDefinition name="Password" type="string" isRequired="true"
displayType="password">
      <csw:DisplayName>Password</csw:DisplayName>
      <csw:Description>Password used for authentication</csw:Description>
      <csw:DefaultValue>
        <csw:Absolute>test</csw:Absolute>
      </csw:DefaultValue>
     </csw:PropertyDefinition>
  </csw:PropertyDefinitionSet>
 </csw:PropertyDefinitions>
</csw:StepTemplate>
```

Step Template Schema

The following is a step template schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://schemas.confluentsw.com/ws/2004/07/policy"
xmlns:csw="http:// schemas.confluentsw.com/ws/2004/07/policy"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xsd:complexType name="Restriction">
      <xsd:sequence>
         <xsd:element name="length" type="csw:Facet" minOccurs="0"
maxOccurs="unbounded"/>
         <xsd:element name="minLength" type="csw:Facet" minOccurs="0"
maxOccurs="unbounded"/>
         <xsd:element name="maxLength" type="csw:Facet" minOccurs="0"
maxOccurs="unbounded"/>
         <xsd:element name="pattern" type="csw:Facet" minOccurs="0"
maxOccurs="unbounded"/>
         <xsd:element name="enumeration" type="csw:Facet" minOccurs="0"
maxOccurs="unbounded"/>
         <xsd:element name="totalDigits" type="csw:Facet" minOccurs="0"
maxOccurs="unbounded"/>
         <xsd:element name="fractionDigits" type="csw:Facet" minOccurs="0"
maxOccurs="unbounded"/>
         <xsd:element name="minInclusive" type="csw:Facet" minOccurs="0"
maxOccurs="unbounded"/>
         <xsd:element name="maxInclusive" type="csw:Facet" minOccurs="0"
maxOccurs="unbounded"/>
         <xsd:element name="minExclusive" type="csw:Facet" minOccurs="0"
maxOccurs="unbounded"/>
         <xsd:element name="maxExclusive" type="csw:Facet" minOccurs="0"
maxOccurs="unbounded"/>
         <xsd:element name="whiteSpace" type="csw:Facet" minOccurs="0"
maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="base" type="xsd:string" use="required"/>
   </xsd:complexType>

   <xsd:complexType name="Facet">
      <xsd:attribute name="value" type="xsd:string" use="required"/>
   </xsd:complexType>

   <xsd:complexType name="PropertyDefinitionType">
      <xsd:all>
         <xsd:element name="DisplayName" type="xsd:string" minOccurs="0"/>
         <xsd:element name="Description" type="xsd:string" minOccurs="0"/>
         <xsd:element name="DefaultValue">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:choice>
                     <xsd:element name="PropertyRef" type="xsd:string"/>
                     <xsd:element name="ResourceRef" type="xsd:string"/>
                     <xsd:element name="Absolute" type="xsd:string"/>
                  </xsd:choice>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="SimpleType" minOccurs="0" maxOccurs="1">
            <xsd:complexType>
               <xsd:all>
```

```
                  <xsd:element name="Restriction" type="csw:Restriction"
minOccurs="0"/>
               </xsd:all>
               <xsd:attribute name="name" type="xsd:string" use="required"/>
            </xsd:complexType>
         </xsd:element>
      </xsd:all>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="type" type="xsd:string" use="required"/>
      <xsd:attribute name="displayType" type="xsd:string" use="optional"/>
      <xsd:attribute name="isMultivalued" type="xsd:string" use="optional"/>
      <xsd:attribute name="isRequired" type="xsd:string" use="optional"/>
   </xsd:complexType>

   <xsd:element name="StepTemplate">
      <xsd:complexType>
         <xsd:all>
            <xsd:element name="Description" type="xsd:string" minOccurs="0"/>
            <xsd:element name="Implementation" type="xsd:string"/>
            <xsd:element name="PropertyDefinitions">
               <xsd:complexType>
                <xsd:sequence>
                   <xsd:element name="PropertyDefinitionSet" minOccurs="0"
maxOccurs="unbounded">
                     <xsd:complexType>
                       <xsd:sequence>
                        <xsd:element name="PropertyDefinition"
                          type="csw:PropertyDefinitionType" minOccurs="0"
maxOccurs="unbounded"/>
                          </xsd:sequence>
                          <xsd:attribute name="name" type="xsd:string"
use="optional"/>
                       </xsd:complexType>
                     </xsd:element>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="Faults" minOccurs="0">
               <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="Fault" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
                 </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
         </xsd:all>
         <xsd:attribute name="id" type="xsd:string" use="required"/>
         <xsd:attribute name="name" type="xsd:string" use="required"/>
         <xsd:attribute name="package" type="xsd:string" use="required"/>
         <xsd:attribute name="timestamp" type="xsd:string" use="required"/>
         <xsd:attribute name="version" type="xsd:nonNegativeInteger"
use="optional"/>
      </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

Index

A

C

D

E

I

J

O

P

R

S

T