



SESM Configuration Management

The SESM installation program assigns initial values to all of the key SESM attributes, using a combination of default values and values you provide during the installation. This chapter describes how to change these initial configuration values. Topics in this chapter are:

- [Introduction, page 3-1](#)
- [Using the SESM Remote Management Tool, page 3-3](#)
- [Directly Editing MBean Configuration Files, page 3-13](#)
- [Changing web.xml and webdefault.xml, page 3-18](#)

Introduction

This section introduces terms and concepts related to configuring SESM applications.

Java Management Extensions

SESM configuration is based on the Java Management Extensions (JMX) specification and its related JMX MBean standards. For descriptions of these standards, go to:

<http://java.sun.com/products/JavaManagement>

When you install the Jetty component from the SESM installation package, you are also installing a JMX server from Sun Microsystems. You can substitute any JMX-compliant server.

The JMX server, sometimes known as the MBean server, is a registry for objects which are exposed to management operations by an agent. Any object that is registered with the JMX server becomes visible to the agent. (For SESM, the agent is the Cisco ConfigAgent.) MBeans are registered by the ConfigAgent or by other MBeans.

The Cisco ConfigAgent is a JMX-compliant agent provided by Cisco. ConfigAgent configures MBeans by reading and implementing values from MBean configuration files. ConfigAgent is an MBean, started by the SESM web application. The contents of the MBean configuration files control ConfigAgent activity.

MBean Description

MBeans are Java classes that follow a model described in the JMX standards. An MBean represents the management interface for a resource. The management interface is the set of all necessary information and controls that a management application needs to operate on the resource.

SESM uses attributes in MBeans to:

- Configure components and the communication connections between those components.

Read-write attributes in the SESM MBeans let deployers configure the application. For example, the SESM MBean configures the SESM mode; the SSG MBean configures communication between SSG and the SESM web application, the AAA MBean configures communication between RADIUS servers and the SESM web application, and so on. Container-specific parameters are also defined as MBeans. For example, Cisco created a logging MBean for the Jetty server.

- Provide metrics about a running application.

Read-only attributes in the SESM MBeans let deployers monitor a running application. For example, the SESM MBean includes attributes for the current number of active sessions, the highest number of active sessions handled by this application so far, the number of authenticated sessions, the number of failed authentications, and so on. The SSG MBean includes attributes for the number of requests received, the number of access reject messages received, the number of timeouts, and so on.

Methods for Changing MBean Attribute Values

To change the configuration attributes in an SESM application's MBeans, you can:

- Use the SESM remote management tool—You can change the value of most MBean attributes while the SESM application is running by using the SESM remote management tool. These changes take effect immediately on the running application. You can optionally store the changes in the MBean configuration file so that they persist over application restarts.

See the [“Using the SESM Remote Management Tool”](#) section on page 3-3 for more information.

- Directly edit the application's MBean configuration file—You can change the value of some attributes by directly editing the appropriate MBean configuration file. These changes take effect the next time you start the application.

See the [“Directly Editing MBean Configuration Files”](#) section on page 3-13 for more information.

Monitoring Features

To monitor a running application, use the SESM remote management tool. You can view the current value of any read-only attribute using this tool. You can optionally set a refresh interval that automatically refreshes the window with new metric values every set number of seconds.

Using the SESM Remote Management Tool

This section describes how to use the SESM remote management tool. Topics are:

- [SESM Remote Management Overview, page 3-3](#)
- [Accessing an Application's Agent View, page 3-4](#)
- [Using the Agent View, page 3-8](#)
- [Using the MBean View, page 3-9](#)
- [Monitoring an Application, page 3-12](#)

SESM Remote Management Overview

The SESM Remote Management tool provides a way to monitor and change the attributes in a running SESM application. It also provides a way to optionally store changes in the application configuration files, so that the changes persist across restarts.

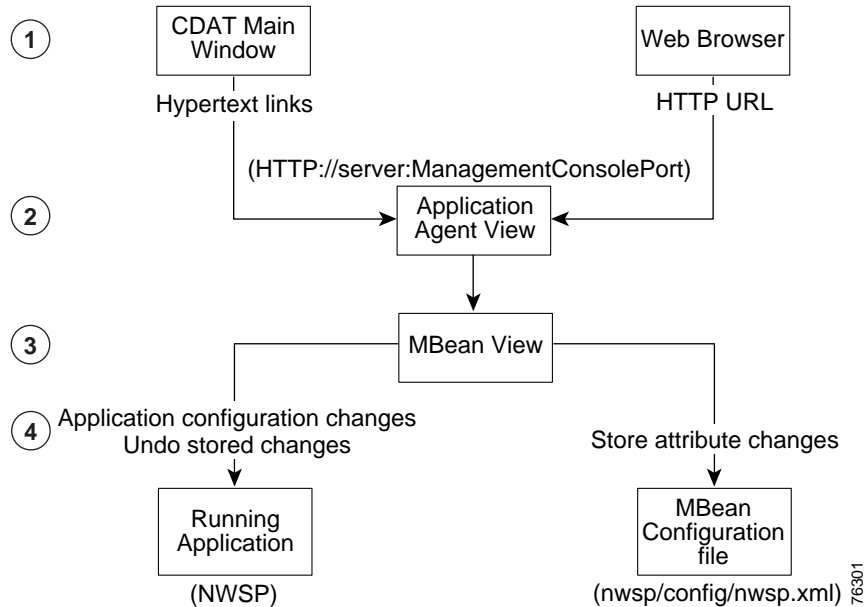


Note

The SESM ManagementConsole is an adaptation of the HTML adaptor server included with the Sun example JMX server. The Cisco adaptations in this release add persistence features to the server. Plans for future SESM releases include an improved user interface.

An application's Agent View is the window into SESM remote management. [Figure 3-1](#) summarizes how to access the Agent View and the tasks you can perform from it.

Figure 3-1 Remote Management Summary



- 1 Each SESM application has a management console, known as the Agent View. You can access an application's Agent View in two ways:
 - Click a link configured on the CDAT main window—From this central location, you can conveniently access the Agent Views for all of the SESM applications.
 - Enter the URL for the application's management console in a web browser.
- 2 An application's Agent View lists all of the MBeans in the running application. From the Agent View, you can access MBean Views.
- 3 An MBean View provides access to all of the attributes in the MBean.
- 4 From the MBean View, you can perform the following actions on attribute values:
 - View current attribute values for the running application.
 - Apply changes to most Read/Write attributes. Applied changes take immediate effect on the running application.
 - Store changes in the application's configuration file. Stored changes persist for future restarts of the application.

Undo (revert) changes sequentially from the most recent store to the first store made in the session. The Undo action only affects the running application, even though it undoes the stored changes. To persist an undo, you must store the change.

Accessing an Application's Agent View

This section describes how to configure, start, and access an Agent View. Topics are:

- [Configuring the ManagementConsole MBean, page 3-5](#)
- [Starting and Removing the Management Console, page 3-5](#)
- [URLs for Accessing Agent Views, page 3-6](#)
- [CDAT Main Window, page 3-6](#)
- [Configuring Links to Agent Views on the CDAT Main Window, page 3-7](#)

Configuring the ManagementConsole MBean

All of the SESM applications include the ManagementConsole MBean, which configures and starts an Agent View for the application. [Table 3-1](#) describes the attributes in the ManagementConsole MBean.

Table 3-1 SESM Portal Application—ManagementConsole MBean

Attribute Name	Explanation
Port	<p>Specifies the management console port for this application.</p> <p>In the installed configuration files, the port value is a system property named:</p> <pre>management.portno</pre> <p>All of the installed startup scripts set this system property to the following value:</p> <pre>application.portno + 100</pre> <p>For example, if the application.portno is 8080, the management.portno is 8180.</p> <p>This runtime setting overrides any value you enter in the configuration file. To change the value of this attribute, edit the start script.</p>
AuthInfo	<p>AuthInfo provides a level of access control on the Management Console. When a user attempts to access the management console port from a web browser, a logon window appears. The user must enter a user ID and password that matches values specified in AuthInfo.</p> <p>Each application has a ManageConsole MBean that configures the login values for that application's management console. You can configure different user IDs and passwords for each application or use the same values for all applications.</p> <p>You can specify multiple sets of AuthInfo information to allow multiple users access to a management console.</p> <p>The AuthInfo array has two elements:</p> <ol style="list-style-type: none"> 1. User ID—Enter a user ID that you want to have access to the management console. The default value in all of the MBean configuration files is <code>MgmtUser</code>. 2. Password—Enter a password that will be required to access the management console. The default value in all of the MBean configuration files is <code>MgmtPassword</code>. <p>You can add, change, and delete AuthInfo values in the configuration files or on the management console.</p> <p>Note If you use the management console to change or delete the user ID or password that you used to log on to the console, the console redisplay the logon prompts. You must log in again using the new authentication values.</p>

Starting and Removing the Management Console

All of the SESM applications are configured to start a management console on application startup.

If you do not want to start a management console for an application, comment out the following lines in the application's MBean configuration file:

```
<Action jmxname="com.cisco.sesm:name=ManagementConsole">
  <Call name="start"/>
</Action>
```

URLs for Accessing Agent Views

You can access an Agent View by typing its URL in the address field of a web browser.

The URL for accessing the Agent View must include the name of the host on which the application is running and the configured management console port number (for example, the value for management.portno). An example URL for the NWSP Agent View is:

```
http://server1:8180
```

CDAT Main Window

The CDAT main window provides the most convenient way to access Agent Views. This window contains links to all of the Agent Views for all of the SESM applications that you install. You can add additional links as you develop more applications. [Figure 3-2](#) shows the CDAT main window.

Figure 3-2 CDAT Main Window



To use the CDAT main window to access an Agent View, follow these procedures:

Step 1 Start CDAT. The CDAT startup script is located in:

```
jetty
  bin
    startCDAT
```

Step 2 Open a web browser.

Step 3 Direct the browser to the CDAT main window ([Figure 3-2](#)).

The URL for accessing CDAT must include the server name where the CDAT is running and the configured CDAT port. The default port used by the installation program is 8081. An example URL for the CDAT main window is:

```
http://server1:8081
```

Step 4 Click the hot text for the Agent View that you want to access.

Configuring Links to Agent Views on the CDAT Main Window

The SESM installation process adds a link to the CDAT main window for Agent Views to each SESM application that you install. You can add additional links as you install more instances of the applications or if you develop customized applications.

To add additional links or to change the URLs behind the existing links, edit the links attribute in the MainServlet MBean. See the [“MainServlet MBean” section on page 6-2](#) for information about the links attribute.

Using the Agent View

The Agent View displays the MBeans in a running application. Figure 3-3 shows the Agent View for a NWSP application running in LDAP mode.

Figure 3-3 Agent View

Agent View

[JMX RI/1.0]

Filter by object name:

This agent is registered on the domain **com.cisco.sesm**. Admin
 This page contains **22** MBean(s).

List of registered MBeans by domain:

- **JMImplementation**
 - [type=MBeanServerDelegate](#)
- **com.cisco.sesm.ignore**
 - [name=ManagementAdaptor](#)
- **com.cisco.sesm.jmx**
 - [name=Version](#)
- **com.cisco.sesm**
 - [agent=Configuration](#)
 - [context=sesm](#)
 - [name=Directory](#)
 - [name=Directory.type=Connection.instance=Primary](#)
 - [name=Directory.type=Connection.instance=Secondary](#)
 - [name=Logger](#)
 - [name=ManagementConsole](#)
 - [name=SESM](#)
 - [name=SSG](#)
 - [name=Version](#)
 - [name=WebApp](#)
- **org.mortbay.jetty**
 - [Debug=0](#)
 - [name=Jetty.Server=0](#)

769302

Table 3-2 explains the actions you can perform from the AgentView.

Table 3-2 Actions from the Agent View

Name	Description
Admin button	Click the Admin button at the top of the window to add a new MBean to the application. Note You should not need to add new MBeans to installed applications.
MBean links	Click an MBean in the list to navigate to the MBean View.

Using the MBean View

The MBean View displays the attributes in an MBean. Figures 3-4 and 3-5 show the MBean View for the WebApp MBean in NWSP. Table 3-3, which follows the figures, explains the numbered callouts in these figures.

Figure 3-4 MBean View—Top Portion

MBean View
[JMX RMI 1.0]

- MBean Name: com.cisco.sesm.name=WebApp
- MBean Java Class: com.cisco.sesm.webapp.config.WebAppMBean

Reload Period in seconds: 1 2

[Back to Agent View](#)

MBean description:
com.cisco.sesm.webapp.config.WebAppMBean

List of MBean attributes: 3

Name	Type	Access	Value
confirmAtAccountLogoff	boolean	RW	<input checked="" type="radio"/> True <input type="radio"/> False
confirmAtServiceLogoff	boolean	RW	<input checked="" type="radio"/> True <input type="radio"/> False
confirmAtServiceLogon	boolean	RW	<input type="radio"/> True <input checked="" type="radio"/> False
credentialMaxLength	int	RW	<input type="text" value="30"/>
defaultURI	java.lang.String	RW	<input type="text" value="/home"/>
dimensions	com.cisco.sesm.webapp.config.DimensionData[]	RW	Type Not Supported: [[Lcom.cisco.sesm.webapp.config.DimensionData;@21302
serviceLogonURI	java.lang.String	RW	<input type="text" value="/serviceLogon"/>
serviceNotGivenURI	java.lang.String	RW	<input type="text" value="/status"/>
serviceStartURI	java.lang.String	RW	<input type="text" value="/serviceStart"/>
serviceSubscriptionURI	java.lang.String	RW	<input type="text" value="/subscriptionManage"/>
sessionTimeout	int	RW	<input type="text" value="7200"/>

4

7-6306

Figure 3-5 MBean View—Bottom Portion

Apply

List of MBean operations: 5

[Description of addDimensionAttribute](#)

void addDimensionAttribute (java.lang.String param0
 (java.lang.String param1
 (java.lang.String param2

[Description of addDimension](#)

void addDimension (int param0
 (java.lang.String param1
 (java.lang.String param2

[Description of addDimension](#)

void addDimension (int param0
 (java.lang.String param1

[Description of undo](#)

void undo 6

[Description of store](#)

void store 7

76306

Table 3-3 Actions from the MBean View

Figure Key	Name	Description
1	Reload interval Reload button	A reload obtains new information from the application and reloads the page. <ul style="list-style-type: none"> The reload interval specifies the number of seconds between automatic reloads. You can change the reload interval here. The change takes effect immediately. If the reload interval is 0 (the default), use the Reload button to manually reload the view.
2	Unregister button	Makes the MBean inaccessible to the running application. Do not use this button.

Table 3-3 Actions from the MBean View (continued)

Figure Key	Name	Description
3	MBean attributes	<p>Lists all of the attributes in the MBean. From this section, you can:</p> <ul style="list-style-type: none"> • Display a short description of the attribute—Click the attribute name. For more detail about any attribute, see the appropriate chapter in this manual. • Change the value of read-write attributes • Monitor metrics (read-only attributes) <p>To change an attribute value, do one of the following, depending on the attribute type:</p> <ul style="list-style-type: none"> • Integers and strings—Type the attribute value in the Value column. • Booleans—Choose the desired radio button. • Arrays: <ul style="list-style-type: none"> – If the Value column contains the phrase “Type Not Supported”—Choose one of the buttons from the MBean Operations section. – If the Value column contains a hotlink phrase “view the values of <i>attribute</i>”—Click the link, which takes you to another page that lists the array elements and current values. Use the appropriate operation in the MBean Operations section to add or change element values.
4	Apply button	Sends the attribute changes to the running application. The change takes effect immediately on the running application unless you receive an error message stating otherwise.
5	MBean operations	Lists operations that you can perform against the MBean. The list is different for each MBean. However, all MBeans include the Store and Undo operations, described below.
6	Undo button	<p>Reverts the running application to the state before the last store. All store operations are captured and can be undone, in sequential order starting with the last change first. You can reverse a previously stored undo. Table 3-4 shows how the Undo operation works.</p> <p>Note The Undo operation applies to the running application only. To save an Undo action to the configuration files (that is, undo the changes stored in the configuration file), click the Store button again.</p>
7	Store button	<p>Saves the attribute changes in the appropriate configuration file (for example, nwsp.xml). This action persists the changes for future application restarts. The Store button has the following effects on the MBean in the configuration file:</p> <ul style="list-style-type: none"> • Deletes any <SystemProperty> or <Property> tags used in the MBean in the originally-installed configuration file. The Store button saves the currently defined value of all attributes in the MBean, regardless of how those values were derived. The Store operation is not aware of property definitions or values assigned by the startup script. • Deletes comments in the MBean. • Includes all read-write attributes in the MBean, whereas the installed configuration files might include only the most commonly changed attributes. • Deletes a <Call> tag inside a <Configure> tag. If the <Call> element sets an attribute value, the rewritten MBean contains the attribute assignment performed in a different way. However, if the <Call> element is performing an action other than setting an attribute value, the action is lost. The correct way to call methods is to use the <Action> tag.

Table 3-4 Sequential Store and Undo Operations

Action	Attribute Value in the Running Application	Attribute Value in the Configuration File
Startup	5	5
Change the value to 10 in the MBean View	5	5
Apply the change	10	5
Store the change	10	10
Undo	5	10
Store	5	5

Monitoring an Application

The SESM application MBeans include read-only attributes that provide activity, performance and memory metrics. You can monitor these metrics from the same MBean View that you use to change the values of read-write attributes.

Some useful monitoring features on the MBean View are:

- **Reload period**—Set an automatic refresh rate by changing the reload period. The browser automatically refreshes the attributes values at the rate specified by the reload period. The default reload period is 0, which turns off the automatic refresh feature.
- **Reload button**—If you do not set an automatic reload period, you can refresh the read-only values at any time by clicking the Reload button.

Figure 3-6 shows metrics in the SESM MBean in the NWSP application.

Figure 3-6 Metrics in the SESM MBean in the NWSP Application

MBean View

[JMX RI/1.0]

- **MBean Name:** com.cisco.sesm.name=SESM
- **MBean Java Class:** com.cisco.sesm.core.model.SESMMBean

Reload Period in seconds:

[Back to Agent View](#)5

MBean description:

SESM Mode configuration.

List of MBean attributes:

Name	Type	Access	
activeAuthenticatedSessions	int	RO	0
activeSessions	int	RO	0
authenticatedSessions	int	RO	0
authenticationFailures	int	RO	0
authenticationSPI	java.lang.String	RO	com.cisco.sesm.spis.radius.RADIUSAuth
authenticationTime	long	RO	0
authorizationSPI	java.lang.String	RO	com.cisco.sesm.spis.dess.DESSAuthorizati
authorizationTime	long	RO	0
autoConnect	boolean	RW	<input type="radio"/> True <input checked="" type="radio"/> False

76307

Directly Editing MBean Configuration Files

You can use a text editor to directly edit any of the SESM MBean configuration files. This section includes information related to direct editing. Topics are:

- [Restarting Applications after Editing](#), page 3-14
- [MBean Configuration File Names](#), page 3-14
- [MBean Configuration File Format](#), page 3-15
- [SystemProperty and Property Tags in Configuration Files](#), page 3-17

Restarting Applications after Editing

If you change configuration values by directly editing the configuration files, you must stop and restart the SESM application and its Jetty server before the changes take effect. If you deployed SESM in LDAP mode, you also must stop and restart RDP.

See [Chapter 9, “Running SESM Components,”](#) for instructions on stopping and starting applications.

MBean Configuration File Names

The MBean configuration files are XML files in a format defined in `xmlconfig.dtd`, a Cisco DTD. These files set configurable attributes in SESM. The SESM installation program assigns values for all of the key attributes in these files, using a combination of default values and values you provide during the install.

[Table 3-5](#) lists all of the MBean configuration files in SESM deployments.

Table 3-5 Summary of MBean Configuration Files

Component	File Path Name	Description
Container (Jetty)	jetty config nwsp.jetty.xml wap.jetty.xml pda.jetty.xml cdat.jetty.xml captiveportal.jetty.xml messageportal.jetty.xml	These files configure Jetty server containers See the “Jetty Container MBeans” section on page 4-2 for information.
SESM web portals	nwsp config nwsp.xml wap config wap.xml pda config pda.xml	These files configure the following items for the web portals: <ul style="list-style-type: none"> • SESM mode and other deployment options. • Communication between SESM applications and SSG, as appropriate. • Communication between SESM applications and RADIUS servers. • Logging and debugging for the SESM application. See the “SESM Portal Application MBeans” section on page 5-1.
Captive portal solution	captiveportal config captiveportal.xml messageportal config messageportal.xml	These files configure captive portal options and behavior. See Chapter 11, “Deploying a Captive Portal Solution,” for more information.
RDP	rdp config rdp.xml	This file configures: <ul style="list-style-type: none"> • RDP options and RDP communication with SSG • Optionally, RDP communication with a RADIUS server • Logging and debugging for RDP See the “RADIUS Data Proxy MBeans” section on page 7-3.

Table 3-5 Summary of MBean Configuration Files (continued)

Component	File Path Name	Description
CDAT	cdat config cdat.xml	This file configures: <ul style="list-style-type: none"> System resource usage for the CDAT application Logging and debugging for the CDAT application Agent View links on the CDAT main window See the “ CDAT Application MBeans ” section on page 6-1 for more information.
SPE	applicationName config dessauth.xml For example: nwsp/config/dessauth.xml pda/config/dessauth.xml wap/config/dessauth.xml rdp/config/dessauth.xml cdat/config/dessauth.xml	The dessauth.xml files configure LDAP directory security and connection attributes, SPE caching, and SPE logging. Each SESM application has its own version of the dessauth.xml file. See the “ SPE Attributes ” section on page 8-1 for more information.

MBean Configuration File Format

This section summarizes the MBean file format Mdefined in xmlconfig.dtd. The purpose of this summary is to provide enough information for you to easily edit the configuration files.

Use the following example as a reference while reading the format guidelines that follow. The example configures the Logger, Version, and ManagementConsole MBeans for SESM portals.

```
<XmlConfig>

<!-- ===== -->
<Instantiate order="1"
    class="com.cisco.sesm.jmx.LoggerMBean"
    jmxname="com.cisco.sesm:name=Logger" />

<Instantiate order="5"
    class="com.cisco.sesm.jmx.VersionMBean"
    jmxname="com.cisco.sesm.jmx:name=Version" />

<Instantiate order="99"
    class="com.cisco.sesm.jmx.AgentView"
    jmxname="com.cisco.sesm:name=ManagementConsole" />

<!-- ===== -->
<Configure jmxname="com.cisco.sesm:name=Logger">
  <Set name="debug" type="boolean">false</Set>
  <Set name="debugPatterns"></Set>
  <Set name="debugThreads"></Set>
  <Set name="debugVerbosity">LOW</Set>
  <Set name="logDateFormat">yyyyMMdd:HHmmss.SSS</Set>
  <Set name="logFile"><Property name="application.home"
default="." />/logs/yyyy_mm_dd.application.log</Set>
  <Set name="logFrame" type="boolean">false</Set>
  <Set name="logThread" type="boolean">false</Set>
  <Set name="logStack" type="boolean">false</Set>
  <Set name="logToErr" type="boolean">false</Set>

```

```

    <Set name="trace"      type="boolean">true</Set>
    <Set name="warning"   type="boolean">true</Set>
  </Configure>
<!-- ===== -->
<Action jmxname="com.cisco.sesm:name=ManagementConsole">
  <Call name="start"/>
</Action>

<!-- ===== -->
<Configure jmxname="com.cisco.sesm:jmx:name=Version">
  <Set name="verbose" type="boolean">>false</Set>
</Configure>

<!-- ===== -->
<Configure jmxname="com.cisco.sesm:name=ManagementConsole">
  <Set name="port" type="int"><Property name="management.portno" default="8180"/></Set>
  <Set name="authInfo">
    <Array class="com.sun.jdmk.comm.AuthInfo">
      <Item>
        <New class="com.sun.jdmk.comm.AuthInfo">
          <Set name="password">MgmtPassword</Set>
          <Set name="login">MgmtUser</Set>
        </New>
      </Item>
    </Array>
  </Set>
</Configure>

```

The following guidelines explain the basic format of the MBean configuration files.

- The MBean configuration file contains a single `<XmlConfig>` element containing one or more `<Instantiate>`, `<Configure>`, and `<Action>` elements.
- An `<Instantiate order = x>` element causes the ConfigAgent to construct and initialize the named MBean or class of MBeans.

The value assigned to the order attribute controls the order in which objects are initialized by the ConfigAgent. The lowest value is initialized first and the highest value is initialized last. For example, in the `nwsp.xml` file, the logger MBean uses the value 1, to ensure that it is initialized first.

After being initialized, an MBean registers itself with the MBean server. When ConfigAgent detects the newly registered object, it then configures the object.

- An `<Action>` element calls methods on an MBean.
- Each `<Configure>` element describes the configuration for either:
 - A single MBean, identified with the name attribute
 - A class of MBeans, identified with the class attribute

ConfigAgent can match a registered MBean by both class and name.

- The `<Set>` tag within a `<Configure>` element identifies an MBean attribute. The format for the `<Set>` tag is:

```
<set name="attributeName" [type="dataType"]>value</set>
```

Where:

attributeName is the MBean parameter name whose value is being set. Do not change any *attributeName*.

dataType is the required data type of the value you specify. Do not change *dataType* unless the change is related to application development. The *dataType* can be: none (which defaults to string), string, int, boolean, URL, an Array element, a Map element, or a New element.

value is the attribute value. You can edit the value, making sure that the value you provide conforms to the data type specified.

- The <Call> tag calls a method defined within the class or the object's class. If the method expects arguments, they are specified within the call tag as well.

Any <Call> tag inside a <Configure> tag disappears if you persist the MBean with the remote management tool. If the <Call> element is setting an attribute value, the rewritten MBean contains the attribute assignment performed in a different way. However, if the <Call> element was used to perform an action other than setting an attribute value, the action is lost. The correct way to call methods is to use the <Action> tag.

- The <Arg> tag inside a call tag can be set to any of the following:
 - Literal values.
 - Objects that are created by a New element or returned by a Call element. Call and New elements might contain Set, Put, Call, Array, or Map elements after any Arg elements. These nested elements are applied to the created or returned object.
- The <Action> tag calls a method defined within the class.
- A <SystemProperty> or <Property> tag might appear inside a <Set> or <Call> tag.



Note The default values assigned in these tags are not used if a value is assigned in the start script. You must remove any use of the setting in the appropriate startup script for the default values in the configuration files to take effect. See the next section, “[SystemProperty and Property Tags in Configuration Files](#)”, for more information.

Cisco ConfigAgent performs the following management functions for MBeans.

- Constructs and initializes an MBean—The <Instantiate> tag causes ConfigAgent to construct and initialize an MBean. Most MBeans are initialized by other objects (for example, other MBeans) and not by ConfigAgent.

After initialization, an MBean registers with the JMX server.

- Configures an MBean—The <Configure> tag causes ConfigAgent to configure an MBean. ConfigAgent can configure existing MBeans and MBeans that are registered later. ConfigAgent configures an MBean if there is a matching entry in the XML file for that MBean. The <Set> tag sets attribute values for the MBean.
- Performs actions on an MBean—The <Action> tag causes ConfigAgent to perform the specified action. For example, ConfigAgent can start an MBean.

SystemProperty and Property Tags in Configuration Files

The installed MBean configuration files use <SystemProperty> and <Property> tags as the value for some attributes. Both tags use the features of a Java system property. The difference between the two tags is:

- SystemProperty tags—The property value applies to the Java virtual machine (VM). All applications running in the same container are configured to use the same value.

- Property tags—The property value applies to the name space of an application. Multiple applications running in the same VM can use properties of the same name but configured with different values.

**Note**

If you use the Store button in the SESM remote management tool to persist a change to an MBean attribute, any <SystemProperty> and <Property> tags in that MBean are written over. The Store button saves the currently defined value of all attributes in the MBean, regardless of how those values were derived. The management tool does not know about properties in the configuration file, nor does it know about values defined in startup scripts.

The value of a property is set as follows:

- You can specify a value on the command line at run time. The command line value overrides all other values. The -D argument to the java command defines the value of a property. For example:

```
startNWSP.sh -jvm -DpropertyName=value
```

- You can specify a value in the start script using the -D option to the java command. For example, the following lines from the installed start script (START.sh or START.cmd) assigns values to some properties.

```
$JAVA $SERVER -Xms64m -Xmx64m \  
-classpath $CLASSPATH \  
-Dinstall.root=$INSTALLDIR \  
-Djetty.home=$JETTYDIR \  
-Dapplication.home=$APPDIR \  
-Dapplication.portno=$PORTNO \  
-Dapplication.ssl.portno=$SSLPORTNO \  
-Dmanagement.portno=$MGMTPORTNO \  

```

For a description of how the start script derives values for the variables used in the assignments, see [Table 9-1 on page 9-5](#).

- If a value is not specified by either of the above methods at application run time, the application uses a default value specified in the MBean configuration file, in the <SystemProperty> or <Property> tag.

Changing web.xml and webdefault.xml

J2EE configuration files, such as web.xml and webdefaults.xml, define how the applications run in the J2EE environment. These files conform to Java specifications, as described in the Java Servlet Version 2.3 specifications from Sun Microsystems.

The *Cisco Subscriber Edge Services Manager Web Developer Guide* describes application-specific parameters in the J2EE configuration files. For information about other parameters, see the Java Servlet Version 2.3 specifications. To download these specifications, go to:

<http://java.sun.com/aboutJava/communityprocess/first/jsr053>

Table 3-6 shows the J2EE configuration files used to configure SESM web portals.

Table 3-6 Summary of J2EE Configuration Files

Component	File Path and Name	Description
Container (Jetty)	jetty config webdefault.xml	This file sets attributes for the Jetty server's handling of HTTP requests and how they map to servlets and JSPs.
SESM application	applicationName webapp WEB-INF web.xml	This file defines J2EE application parameters, including parameters related to Java Server Pages (JSPs). There is a separate web.xml file for each web application.
SESM application	applicationName webapp WEB-INF web-jetty.xml	This file is required for the port-bundle host key feature. See “Container Requirement for the Port-Bundle Host Key Feature” section on page 4-1 for more information.

