



Configuring Components after Installation

This chapter describes all of the configurable attributes in the Subscriber Edge Services Manager (SESM) software components. Use this chapter to change or fine-tune attributes after installation.

This chapter includes the following topics:

- [Configuration Overview, page 4-1](#)
- [Configuring the J2EE Jetty Container, page 4-7](#)
- [Configuring an SESM Web Application, page 4-16](#)
- [Configuring RDP, page 4-31](#)
- [Configuring CDAT, page 4-35](#)
- [Configuring DESS, page 4-37](#)
- [Configuring Specific Features, page 4-42](#)

Configuration Overview

This section provides an overview of the configuration files and the configuration technology used by SESM. It includes the following topics:

- [Changing Configuration Information, page 4-1](#)
- [Configuration Technology, page 4-2](#)
- [Configuration Files, page 4-3](#)

Changing Configuration Information

You can change any configuration information by manually editing the configuration files. If you change configuration information, you must stop and restart the SESM web application and the Jetty server. If you deployed SESM in DESS mode, you also must stop and restart RDP. See [Chapter 5, “Running SESM Components,”](#) for instructions.

Configuration Technology

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>

The configuration elements involved in SESM are:

- **MBeans**—MBeans are Java classes that follow a model described in the MBean 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 MBeans to configure components and the communications connections between those components. For example, an SESM MBean configures the SESM mode; an SSG MBean configures communication between SSG and the SESM web application, an 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.

- **JMX 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 Jetty component in the SESM installation package includes a JMX server. You can substitute any JMX-compliant server.

- **Cisco ConfigAgent**—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.
- **MBean Configuration Files**—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. You can change the value of any attribute by editing the appropriate MBean configuration file.

Cisco ConfigAgent

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 itself with the JMX server.

- **Configures an MBean**—The `<Configure>` tag causes ConfigAgent to configure an MBean.

When the ConfigAgent detects a newly registered MBean, ConfigAgent configures that MBean if there is a matching entry in the XML files for that MBean.

The `<Set>` tag sets attribute values for the MBean.

- **Starts an MBean**—The `<Call>` tag causes ConfigAgent to start an MBean.

The contents of the MBean configuration files control ConfigAgent activity.

Configuration Files

Two types of configuration files are used in SESM:

- J2EE configuration files—These are standard J2EE files that conform to Java servlet specifications. Examples are `web.xml` and `webdefaults.xml`.
- MBean configuration files—These XML files conform to a format defined by Cisco. These files are named `application.xml`.

J2EE Configuration Files

The 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.

Administrators do not usually need to change the J2EE configuration files. Therefore, the contents of these files are not documented in this guide. However, web developers might require changes to these files. 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 4-1 lists the J2EE configuration files used to configure SESM web applications. The table includes a brief description of each file and shows the installed location of the file.

Table 4-1 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 web application (NWSP)	nwsp docroot WEB-INF web.xml	This file defines J2EE application parameters, including parameters related to Java Server Pages (JSPs).
SESM captive portal application	captiveportal docroot WEB-INF web.xml	This file defines J2EE application parameters for the captive portal application.
CDAT	cdat docroot WEB-INF web.xml	This file defines J2EE application parameters for CDAT.

MBean Configuration Files

Administrators edit the MBean configuration files to change values of configurable attributes for SESM software components. The installation program assigns initial values for all of the key attributes in these files, using a combination of default values and values you provide during the install. You can change the value of any attribute by editing the appropriate MBean configuration file.

The MBean configuration files conform to `xmlconfig.dtd`, a Cisco DTD. See the “[MBean Configuration File Format](#)” section on page 4-5 for a summary of the MBean configuration file format. See [Appendix C, “DTD for MBean Configuration Files”](#) for the complete DTD.

Each software component in an SESM deployment has its own MBean configuration files. [Table 4-2](#) lists all of the MBean configuration files used in an SESM deployment. The table describes the file location relative to the installation directory and a brief description of the file.

Table 4-2 Summary of MBean Configuration Files

Component	File Path and Name	Description
Container (Jetty)	<pre>jetty config nwsp.jetty.xml cdat.jetty.xml yourapp.jetty.xml</pre>	<p>You can configure the Jetty server instance associated with each application differently. These files configure:</p> <ul style="list-style-type: none"> • Logging and debugging for the Jetty server. This log file name is <code>nnn.jetty.log</code>. • HTTP listener, which configures: <ul style="list-style-type: none"> – The application that is running in the container and the application port. – The web server’s standard HTTP request log. This log file name is <code>nnn.request.log</code>.
SESM web application (NWSP)	<pre>nwsp config nwsp.xml</pre>	<p>This file configures:</p> <ul style="list-style-type: none"> • SESM deployment options • Communication between an SESM web application and SSG • Communication between an SESM web application and RADIUS servers • Attributes for a captive portal application • Logging and debugging for the SESM web application. The log file name is <code>nnn.application.log</code>. • A management port for development and testing purposes
SESM captive portal application	—	Captive portal attributes are included in the MBean configuration file for the SESM web application.
RDP	<pre>rdp config rdp.xml</pre>	<p>This file configures:</p> <ul style="list-style-type: none"> • RDP options and packet handlers • RDP communication with SSG • Optionally, RDP communication with a RADIUS server • Logging and debugging for RDP • A management port for development and testing purposes.

Table 4-2 Summary of MBean Configuration Files (continued)

Component	File Path and 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 • A management port for development and testing purposes.
DESS	dess-auth config config.xml	This file configures attributes used by the executing classes in the Dess and Auth application programming interfaces (APIs). The Dess and Auth APIs provide the underlying support for communication between an LDAP directory and the RDP, CDAT, and SESM web applications. If these applications are installed on the same machine, the same config.xml file applies to all of the applications. This file contains attributes that control: <ul style="list-style-type: none"> • Directory security • Directory connections • Caching • Logging

For detailed descriptions of all attributes in the MBean configuration files, see the following tables:

- [Table 4-3 on page 4-10, “Attributes in the Container MBean Configuration Files”](#)
- [Table 4-4 on page 4-18, “Attributes in the Application MBean Configuration File”](#)
- [Table 4-6 on page 4-33, “Attributes in the RDP MBean Configuration File”](#)
- [Table 4-7 on page 4-36, “Attributes in the CDAT MBean Configuration File”](#)
- [Table 4-8 on page 4-38, “Attributes in the Dess-Auth MBean Configuration File”](#)

MBean Configuration File Format

This section summarizes the MBean file format defined in `xmlconfig.dtd`. The purpose of this summary is to provide enough information for you to easily edit the MBean files. For the full text of the DTD, including extensive comments, see [Appendix C, “DTD for MBean Configuration Files.”](#)

Use the following example as a reference while reading the format guidelines that follow. The example configures the debugging MBean for an SESM application.

```
<Instantiate order="1"
  class="com.cisco.aggbu.jmx.LoggerMBean"
  jmxname="com.cisco.aggbu:name=Logger" />

</Instantiate>
```

```

<!-- ===== -->
<Configure jmxname="com.cisco.aggbu:name=Logger">
  <Set name="debug" type="boolean"><SystemProperty name="nwsp.debug"
    default="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"><SystemProperty name="application.log"
    default="./logs"/>/>yyyy_mm_dd.application.log</Set>
  <Set name="logFrame" type="boolean">>false</Set>
  <Set name="logStack" type="boolean">>false</Set>
  <Set name="logThread" type="boolean">>true</Set>
  <Set name="logToErr" type="boolean"><SystemProperty name="nwsp.logToErr"
    default="false"/></Set>
  <Set name="trace" type="boolean">>true</Set>
  <Set name="warning" type="boolean">>true</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 `<Configure>` elements.
- 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

Each `<Configure>` element must contain one of the above attributes, or both. `ConfigAgent` matches a registered MBean by both class and name, so that two `<Configure>` elements might be applied to an MBean.

The `<Instantiate order = x>` tag 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.

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

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

Where:

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

dataType is the required data type of the value you specify. If *dataType* is not explicitly identified, the default *dataType* is string. It is best not to change any *dataType*.

value is the parameter 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.

Containers and Applications

This section defines containers and applications, and describes the relationship between them.

SESM applications and CDAT are J2EE web applications. The J2EE web server is the *container* for the applications that run in it. For example, the Jetty server is the container for the installed NWSP application.

One-to-One Relationship

The SESM core model, the NWSP sample application, and CDAT are designed and configured with the assumption that there is a one-to-one relationship between the web server container and each web application. That is, each application runs in its own web server container. If you are running two instances of the same application, or two different applications, you are running two web servers.

This one-to-one relationship means that you can configure the J2EE server differently for each application. For example, you can turn on logging for one application and turn it off for another.

Configuration File Locations

Each SESM web application (and also CDAT) has two MBean configuration files associated with it. The two files are:

- Application MBean configuration file—Configures the application. For example:

```
nwsp
  config
    nwsp.xml
```

```
cdat
  config
    cdat.xml
```

- Container MBean configuration file—Configures the J2EE server for the application. The container's config directory holds an MBean configuration file for *each* application. For example:

```
jetty
  config
    nwsp.jetty.xml
    cdat.jetty.xml
    newapplication.jetty.xml
```

This modular approach has several advantages:

- It makes it easy to switch containers. If you change the J2EE container, you must make changes to the container MBeans, such as changing class or object names, or even adding more MBeans.
- It clearly defines the process that each MBean is configuring. For example, both the container and the application have logging and debugging MBeans.

The RDP and DESS components are not web applications. Therefore, the jetty directory does not contain an MBean configuration file for those components.

Container Attributes

This section describes the attributes in the J2EE container MBean configuration files. These files are located in the container's config directory. For example:

```
jetty
  config
    nwsp.jetty.xml
    cdat.jetty.xml
```

The container MBean configuration files configure the following MBeans:

- Log MBean—Enables or disables the Jetty server logging mechanism and configures the information to appear in the jetty log files.
- Debug MBean—Enables or disables the Jetty server debugging mechanism.
- HTTP Server MBean—Configures the following:
 - The port that the Jetty server listens on for HTTP requests from subscribers and the listener thread pools. Two listeners are used, a main listener and a listener for requests on the Secure Sockets Layer (SSL). Each listener has one pool.
 - The web application to which the requests should be sent. The installed sample files identify two sample applications: the NWSP application and the captive portal application.
 - A request log, which records all HTTP requests.

Table 4-3 describes the attributes in the container MBean configuration files.

Table 4-3 *Attributes in the Container MBean Configuration Files*

Object Name	Attribute Name	Explanation
LogMBean	append	Indicates if messages overwrite existing contents (false) or are appended to the existing file (true). Installed default: true
	filename	Specifies the log file name and path, as follows: <i>application.log/yyyy_mm_dd.jetty.log</i> Where: <ul style="list-style-type: none"> <i>application.log</i>—Is a Java system property. The same system property is used for all log files, so that they are all created in the same directory. Table 5-1 on page 5-4 describes how the start script sets <i>application.log</i>. <i>yyyy_mm_dd</i>—Is the year, month, and day that the file was created. <i>.jetty.log</i>—Is a constant identifying the Jetty log files.
	logTimezone	Installed default: empty
	logDateFormat	Controls the format of the date stamp in the log messages. Installed default: yyyyMMdd:HHmmss.sss
	logLabels	Controls whether or not logging messages include frame details. Installed default: false
	logOneLine	Installed default: false
	logStackSize	Controls whether or not logging messages include an indication of stack depth. Installed default: false
	logStackTrace	Controls whether or not logging messages include trace information. Installed default: false

Table 4-3 Attributes in the Container MBean Configuration Files (continued)

Object Name	Attribute Name	Explanation
DebugMBean	debug	Controls whether or not debugging messages are produced. Installed default: false
	debugPatterns	By specifying one or more patterns, you turn on a filtering mechanism that excludes any message that does not match the pattern. The patterns are file, class, or method names. Pattern matching is based on substring matches. For example, if you specify the pattern RADIUS, the software focuses on RADIUS messages. To specify multiple patterns, separate the patterns using a comma. Installed default: empty
	debugTriggers	Installed default: empty
	verbose	Specifies the level of detail reported in debugging messages. The range of allowed values is 0 (no details) to 255 (all details). Installed default: 0
	suppressStack	Controls whether or not stack information is included in debug messages. Installed default: false
	suppressWarnings	Controls whether or not warning messages are included in debug messages. Installed default: false

Table 4-3 Attributes in the Container MBean Configuration Files (continued)

Object Name	Attribute Name	Explanation
HttpServer MBean— AddListener for HTTP.SocketListener	port	<p>Sets the port number that the web server listens on. The installed value is a Java system property named:</p> <p style="text-align: center;"><code>application.portno</code></p> <p>Note The startup script sets this system property. Unless you alter the startup script, the default value in the MBean configuration file is ignored during application startup.</p> <p>To change the value of <code>application.portno</code>, edit the application-specific startup script. The SESM installation program sets <code>application.portno</code> in the startup script to the NWSP port that you provided during the installation process.</p> <p>If you are running in captive portal mode, this port value must be 80, whether you explicitly set it here by removing the reference to the Java system property or change the value of <code>application.portno</code> in the startup script.</p>
	minThreads	<p>Sets the minimum number of threads that this listener will maintain during periods of low load. This listener will always have system resources allocated for this number of threads.</p> <p>Installed default: 5</p>
	maxThreads	<p>Sets the maximum number of threads that this listener can allocate resources for, even during peak loads. This listener can have up to this number of threads.</p> <p>Installed default: 255</p>
	maxIdleTimeMs	<p>Specifies how long a thread can be idle (not used) before the listener deallocates it. The unit is milliseconds.</p> <p>Installed default: 60000</p>
	maxReadTimeMs	<p>Specifies the time that a read on a request can block. This is how long the web server waits for a request to come from a client after the client opens a socket connection. When <code>maxReadTimeMs</code> is exceeded, the web server closes the socket connection.</p> <p>Installed default: 60000</p>

Table 4-3 Attributes in the Container MBean Configuration Files (continued)


Object Name	Attribute Name	Explanation
HttpServer MBean— AddListener for HTTP.SunJsseListener	port	<p>Sets the port that the secure socket layer (SSL) listener uses. The installed value is a Java system property named:</p> <pre>application.ssl.portno</pre> <p>Note The startup script sets this system property. Unless you alter the startup script, the default value in the MBean configuration file is ignored during application startup.</p> <p>The generic startup script derives a value for <code>application.ssl.portno</code> based on the value of <code>application.portno</code>, as follows:</p> <pre>application.ssl.portno = application.portno - 80 + 443</pre> <p>To change the value of <code>application.ssl.portno</code>, edit the generic startup script.</p>
	MinThreads	<p>Sets the minimum number of threads that this listener will maintain during periods of low load. This listener will always have system resources allocated for this number of threads.</p> <p>Installed default: 5</p>
	MaxThreads	<p>Sets the maximum number of threads that this listener can allocate resources for, even during peak loads. The listener can allocate up to this number of threads.</p> <p>Installed default: 255</p>
	MaxIdleTimeMs	<p>Specifies the length of time a thread can be idle (not used) before the listener deallocates it. The unit is milliseconds.</p> <p>Installed default: 50000</p>
	Keystore	<p>Sets the pathname of the SSL keystore file. The keystore file is a binary file created by keytool. A sample keystore file is included in the installation. The name and location of the sample is:</p> <pre>jetty.home/config/nwspkeystore</pre> <p>Where:</p> <ul style="list-style-type: none"> <code>jetty.home</code>—Is a Java system property. The NWSP start script derives the value of <code>jetty.home</code> from an expected (installed) directory structure. To change the value of <code>jetty.home</code>, edit the start script. Unless you alter the start script, the default value for <code>jetty.home</code> specified in this MBean configuration file is ignored at run time. <p> Caution A keystore file is required for deployments that use HTTPS. HTTPS does not function without a valid keystore file. The <code>nwspkeystore</code> file included with the SESM installation works, but you should replace it with a keystore valid for your specific deployment. See the “HTTPS Description” section on page A-2 for more information</p>
	Password	Must match the value in the keystore file referenced above.
KeyPassword	Must match the value in the keystore file referenced above.	

Table 4-3 Attributes in the Container MBean Configuration Files (continued)

Object Name	Attribute Name	Explanation
HttpServerMBean— LogSink Configures a log file that records the incoming HTTP requests.	This is a positional argument.	The logSink class has one argument, which specifies the name and location of the request log. The installed value is: <i>application.log/yyyy_mm_dd.request.log</i> Where: <ul style="list-style-type: none"> • <i>application.log</i>—Is a Java system property. whose value is set in the generic startup script. The same system property is used for all log files, so that they are all created in the same directory. See Table 5-1 on page 5-4 for a description of how the start script sets <i>application.log</i>. • <i>yyyy_mm_dd</i>—Is the year, month, and day that the file was created. The installation program uses the appropriate pathname delimiter for the installation platform. • <i>.request.log</i>—Is a constant identifying an HTTP request file.
	retainDays	Indicates the number of days to keep an old log file before deleting it. Installed default: 90
	append	Indicates whether or not to append messages to an existing file or to create a new file for each application instance. Installed default: true

Table 4-3 Attributes in the Container MBean Configuration Files (continued)

Object Name	Attribute Name	Explanation
<p>HttpServer MBean— <Call AddWebApplication></p> <p>The first call to this class adds the NWSP application to run on the web server.</p>	<p>These are positional arguments.</p>	<p>AddWebApplication has five positional arguments:</p> <ol style="list-style-type: none"> 1. The first positional argument specifies the virtual host name for the web server application. 2. The second positional argument specifies the context path for locating the web server application. For example, / or /pathname/*. 3. The third positional argument identifies the location of the application. The XML file sets this value to: <i>application.home/docroot</i> Where: <i>application.home</i> is a Java system property. 4. The fourth positional argument identifies the location of the webdefault.xml file for this application. The XML file sets this value to: <i>jetty.home/config/webdefault.xml</i> Where: <i>jetty.home</i> is a Java system property 5. The fifth positional argument specifies whether or not web archive (WAR) files are used. Valid values are TRUE and FALSE. Set this value to FALSE, since NWSP and CDAT are not WAR files. <p>The first three arguments define the location of the web server application. <i>host/context/application</i></p> <p>The NWSP start script derives the values for <i>application.home</i> and <i>jetty.home</i> from an expected (installed) directory structure. To change the value of <i>application.home</i> or <i>jetty.home</i>, edit the start script.</p>

Table 4-3 Attributes in the Container MBean Configuration Files (continued)

Object Name	Attribute Name	Explanation
<p>HttpServer MBean— <Call AddWebApplication></p> <p>The second call to this class adds the captive portal application to run on the web server.</p>	<p>These are positional arguments.</p>	<p>AddWebApplication has five positional arguments:</p> <ol style="list-style-type: none"> 1. The first positional argument is not used when calling the captive portal application. 2. The second positional argument specifies the context path for locating the web server application. For example, / or /pathname/*. (context) 3. The third positional argument identifies the location of the captive portal application. The XML file sets this value to: <i>install.root/captiveportal/docroot</i> Where: <i>install.root</i> is a Java system property 4. The fourth positional argument identifies the location of the webdefault.xml file for the captive portal application. The XML file sets this value to: <i>jetty.home/config/webdefault.xml</i> Where: <i>jetty.home</i> is a Java system property 5. The fifth positional argument specifies whether web archive (WAR) files are used. Valid values are TRUE and FALSE. Set this value to FALSE, since NWSP and CDAT are not WAR files. <p>The first three arguments define the location of the captive portal application. <i>host/context/application</i></p> <p>The NWSP start script derives the values for <i>install.root</i> and <i>jetty.home</i> from an expected (installed) directory structure. To change the value of <i>application.home</i> or <i>jetty.home</i>, edit the start script.</p>

Configuring an SESM Web Application

This section describes how to configure an SESM web application, using the NWSP application as an example. The section includes the following topics:

- [SESM Application Attributes, page 4-17](#)
- [Associating SSGs and Subscriber Requests, page 4-27](#)

Also see the “[Sample Application MBean Configuration File](#)” section on page F-3.

SESM Application Attributes

This section describes the SESM application MBean configuration file. This file is located in the application's config directory. For example:

```
nwsp
  config
    nwsp.xml
```

The application MBean configuration file configures the following MBeans:

- **Logger**—The `com.cisco.aggbu.jmx.LoggerMBean` configures both logging and debugging tools. The logging tool logs SESM web application activity. The debugging mechanism produces messages useful to developers in debugging applications.
- **ManagementConsole**—This MBean configures a management console port for development and testing purposes. On this port, you can see the currently set values for all attributes in all of the MBean configuration files.
- **SSD**—This MBean configures SESM features and options, including the SESM mode.
- **SSDDemoMode**—This MBean configures SESM in demo mode.
- **SSG**—The `SSG` MBean configures communication between SESM web application and SSG. These components communicate using the RADIUS protocol, so this MBean includes RADIUS protocol attributes. The MBean also includes attributes that determine which SSG should handle a subscriber request.
- **AAA**—The `AAA` MBean configures communication between SESM web application and the RADIUS servers.
- **captiveportal**—This MBean configures captive portal information, including the URL that the captive portal redirects to, which should be the SESM web application.
- **context parameters**—Context parameters are used by an application for any arbitrary reason. The NWSP application uses context parameters to control web page content based on location.

[Table 4-4](#) explains the configurable attributes in the MBeans listed above.

Table 4-4 Attributes in the Application MBean Configuration File

Object	Attribute Name	Explanation
ManagementConsole	Port	<p>Specifies a port for a management console.</p> <p>The management console displays the current settings of all attributes in all of the MBean configuration files. The console is useful in development and testing environments.</p> <p>Note The ManagementConsole is the HTML adaptor server included with the Sun example JMX server. However, the HTML adaptor server is not production quality. For example, configuration changes that you make using the management console are not persistent. You should remove the HTML adaptor server from your configuration before transitioning the SESM deployment to public use.</p> <p>To remove the JMX HTML adaptor server, comment out the following lines in the configuration files:</p> <pre><Configure jmxname="com.cisco.aggbu:name=ManagementConsole"> <Call name="start"/> </Configure></pre> <p>The port attribute is set to a Java system property named:</p> <pre>management.portno</pre> <p>All of the installed startup scripts set this Java 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 first. The user must enter a user ID and password that matches the values specified here.</p> <p>AuthInfo requires two positional arguments:</p> <ol style="list-style-type: none"> 1. User ID—Enter a user ID that will be required to access 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>.

Table 4-4 Attributes in the Application MBean Configuration File (continued)

Object	Attribute Name	Explanation
Logger MBean	debug	<p>Turns debugging on (value is true) or off (value is false).</p> <p>The following parameters control the contents of debug messages that the application generates: logFrame, logStack, logThread, debugPatterns, and debugThreads.</p> <p>When debug is false, the application does not generate debug messages but it can still generate logging messages. The following parameters control the types of logging messages produced: trace and warning.</p> <p>Installed default: false</p>
	debugPatterns	<p>By specifying one or more patterns, you turn on a filtering mechanism that excludes any message that does not match the pattern. The patterns are file, class, or method names. Pattern matching is based on substring matches. For example, if you specify the pattern RADIUS, the software focuses on RADIUS messages. To specify multiple patterns, separate the patterns using a comma.</p> <p>Installed default: empty, which means that you receive all messages.</p>
	debugThreads	<p>Specifies a specific thread name for which to show debugging messages. You can specify multiple thread names, separating them using a comma. By default, no thread name is specified.</p> <p>Because each user interaction with the SESM web application takes place in a thread named for that user, this parameter can be used to focus the logging trace on a specific user activity. (This feature is not implemented in SESM Release 3.1(1).)</p> <p>Installed default: empty</p>
	debugVerbosity	<p>Specifies the level of detail in debugging messages. When the debug attribute is set to false, this attribute is ignored. Values are:</p> <ul style="list-style-type: none"> • MAX • MED • LOW <p>Installed default: LOW</p>
	logDateFormat	<p>Specifies format of dates in the log file.</p> <p>Installed default: yyyyMMdd:HHmmss.SSS</p>

Table 4-4 Attributes in the Application MBean Configuration File (continued)

Object	Attribute Name	Explanation
LoggerMBean (continued)	logFile	Specifies the log file name and location. The installed default is: <i>application.log/yyyy_mm_dd.application.log</i> Where: <ul style="list-style-type: none"> <i>application.log</i>—Is a Java system property. The same system property is used for all log files, so that they are all created in the same directory. See Table 5-1 on page 5-4 for a description of how the start script sets <i>application.log</i>. <i>yyyy_mm_dd</i>—Is the year, month, and day that the file was created. <i>application.log</i>—Is a constant identifying the application log files.
	logFrame	Controls whether or not to log the calling member function. Installed default: false
	logStack	Controls whether or not to log stack traces. Installed default: false
	logThread	Controls whether or not to log thread IDs. Installed default: true
	logToErr	Controls whether or not to route log messages to stderr, in addition to the log file. This parameter is useful for monitoring the SESM web application at the command line. Displaying output to stderr is not recommended for production deployments. Installed default: true
	trace	Controls whether or not to log trace messages. These messages indicate entry and exit to code points. Installed default: true
	warning	Controls whether or not to log warning messages (nonfatal exceptions). Installed default: true

Table 4-4 Attributes in the Application MBean Configuration File (continued)

Object	Attribute Name	Explanation
SSD	mode	<p>An SESM web application runs in one of the following modes. The SESM installation program sets the mode according to the options you choose during installation.</p> <ul style="list-style-type: none"> • RADIUS—In this mode, the SESM web application communicates with SSG and a RADIUS server. • Demo—SESM runs in this mode when you choose the Demo option during installation. In this mode, the SESM web application does not communicate with other components. Rather, it simulates communication by reading data from a Merit flat file. This mode is intended for demonstrations only, when network components such as SSG, RADIUS, or an LDAP server are not available. • DESS—In this mode, the SESM web application communicates with SSG and an LDAP directory. The LDAP directory communication relies on a Cisco application programming interface known as directory-enabled service selection (DESS). <p>The MBean configuration file defines a Java system property for mode:</p> <pre>ssd.mode</pre> <p>This system property is different from most of the other system properties used in the MBean configuration files, in that, by default, the startup script does <i>not</i> set this system property. Therefore, the application runs in the mode specified in the MBean configuration file unless you explicitly override that value at run time.</p> <p>To change the mode, you can:</p> <ul style="list-style-type: none"> • Reinstall the software. • Edit the MBean configuration files, changing the mode and other attributes, as appropriate. • Use the mode option on the SESM application startup script command line. This command line option provides a way to quickly switch between modes for testing purposes. You might need to alter the start script to access a different set of MBean configuration files for each mode, or use some other method to ensure that the attributes match the mode you are using. The syntax is: <ul style="list-style-type: none"> – on Solaris: <code>jetty/bin/startNWSP.sh -mode {Demo RADIUS DESS}</code> – on Windows: <code>jetty\bin\startNWSP.cmd {Demo RADIUS DESS}</code> <p>Note The best way to change the SESM mode is to re-install the software. Several other configuration attributes must be aligned with the mode for SESM to run properly in the selected mode. Also, you might not have all of the appropriate components to run in a mode other than the one you installed. For example, a demo installation does not install the DESS component.</p>
	singleSignOn	<p>Enables (true) or disables (false) the single sign-on feature.</p> <p>If single sign-on is enabled, the SESM web application does not ask a PPP subscriber to authenticate (log on). Instead, the SESM web application uses the SSG's PPP authenticated identity. Installed default: false</p>

Table 4-4 Attributes in the Application MBean Configuration File (continued)

Object	Attribute Name	Explanation
SSD (continued)	profileCache Period	Specifies the time in seconds that a service or group object must be idle in the cache before its resources are deallocated from memory. Installed default: 600
	autoConnect	In RADIUS mode, this parameter is ignored. The automatic connection feature is always available, regardless of parameter settings. In RADIUS mode, the SSG always performs automatic service connections for all services marked as auto connect in a subscriber's profile. In DESS mode, the SSG performs automatic connections if it has the service list. If SSG does not have the service list, you can set this autoConnect parameter to allow the SESM application to perform the automatic connections. The Add Services option, which is set during RDP installation, controls whether or not SSG has a service list in DESS mode. The Add Services option configures RDP to either: <ul style="list-style-type: none"> Return a service list to SSG—In this case, SSG performs automatic connections for services marked as auto connect in a subscriber's profile. Not return a service list to SSG—In this case, SSG cannot perform automatic connections. The advantage to this configuration is that it saves memory on the SSG host (the NRP on the Cisco 6400 UAC). If you configure RDP so that it does not return a service list to SSG, change the value of this autoConnect parameter to true to enable automatic connections by the SESM web application.
SSDDemoMode	demoDataFile	Specifies the file that contains data for the demo mode. The installed value is: <i>application.home/config/demo.txt</i> Where: <i>application.home</i> is a Java system property The NWSP start script derives the value for <i>application.home</i> from an expected (installed) directory structure. To change the value of <i>application.home</i> , edit the start script.

Table 4-4 Attributes in the Application MBean Configuration File (continued)

Object	Attribute Name	Explanation
SSG—Global attributes The global attributes apply to all SSGs that the SESM web application might communicate with. To determine how an SSG was configured, use the show run command on the SSG host.	throttle	The maximum number of simultaneous requests that SESM web applications can send to SSG. This is a RADIUS protocol attribute. The RADIUS protocol queues additional requests and issues them as SSG returns responses or timeout messages for previous requests. You cannot override this global value. Installed default: 20
	PORT	The global value for RADIUS ports on the SSG hosts. This value must match the value that was configured on the SSG host using the following command: <pre>ssg radius-helper authenticationPort</pre> You can create subnet entries in the MBean configuration file to override this global value for specific SSGs.
	TIMEOUTSECS	The number of seconds the SESM web application waits before timing out RADIUS packets that it sends to SSG. You cannot override this global value. Installed default: 5
	RETRIES	The number of times the SESM web application resends a RADIUS packet to SSG if no response is received. You cannot override this global value. Installed default: 3
	SECRET	The global value for the RADIUS protocol shared secret used for communication between the SESM web application and the SSGs. This value must match the value entered on the SSG host using the ssg radius-helper key command. You can create subnet entries in the MBean configuration file to override this global value for specific SSGs.
	MASK	The global value for the mask that the SESM web application applies to incoming subscriber IP addresses to derive an IP address for the SSG. You can create subnet entries in the MBean configuration file to override this global value for specific subnets.
	BUNDLE_LENGTH	The global value for the port bundle length that SSGs use when the host key feature is enabled. Currently, this value can be either: <ul style="list-style-type: none"> • 0—A value of 0 indicates that SSGs are not using the host key feature. • 4—The port bundle length is the number of bits that SSG uses to indicate bundled slots. For example, a value of 4 indicates 16 bundled slots. This value must match the value used in the following command on the SSG host: <pre>ssg port-map length</pre>

Table 4-4 Attributes in the Application MBean Configuration File (continued)

Object	Attribute Name	Explanation
SSG—Subnet entries Use subnet entries to override the global values or to map client subnets to specific SSGs when the host key feature is not being used.	Subnet entries use positional arguments.	The call to setSubnetAttribute has four positional arguments: <ol style="list-style-type: none"> 1. <i>subnetAddress</i> is the subnet for which you are explicitly setting a value, overriding the globally set value. 2. <i>subnetMask</i> is the mask that can be applied to the subscriber's IP address to derive the subnet. 3. <i>argumentName</i> is the argument that you are explicitly setting. 4. <i>argumentValue</i> is the value for <i>argumentName</i>. See the “Associating SSGs and Subscriber Requests” section on page 4-27 for more information.
AAA This MBean defines communication between the SESM web application and the RADIUS server, which occurs only when the SESM application is running in RADIUS mode.	Connection	The Configure element in the AAA MBean includes a connection attribute whose value is either: <ul style="list-style-type: none"> • ServiceProfile—The MBean for this connection type includes the servicePassword attribute. • GroupProfile—The MBean for this connection type includes the groupPassword attribute. The connection name identifies the type of request.
	throttle	The maximum number of simultaneous requests that SESM web applications can send to a RADIUS server. This is a RADIUS protocol attribute. The RADIUS protocol queues additional requests and issues them as the server returns responses or timeout messages for previous requests. Installed default: 256
	timeOut	The number of seconds the SESM web application waits before timing out RADIUS packets that it sends to the AAA server. Installed default: 4
	retryCount	The number of times the SESM web application resends packets to the AAA server if no response is received. Installed default: 3
	primaryIP	The IP address or the host name of the primary AAA server.
	primaryPort	The port number that the primary RADIUS server listens on. Default: 1812
	secret	The shared secret used between the RADIUS server and the SESM web application. The shared secret must be the same for the primary and secondary servers. It must match the secret specified when you configured SESM as a NAS client on the RADIUS server. Default: cisco.
secondaryIP	The IP address or the host name of the secondary AAA server. If you are not using a secondary RADIUS server, enter the same value used for the primary server.	

Table 4-4 Attributes in the Application MBean Configuration File (continued)

Object	Attribute Name	Explanation
AAA (continued)	secondaryPort	The port number that the secondary RADIUS server listens on. If you are not using a secondary RADIUS server, enter the same value used for the primary server. Default: 1812
	servicePassword	The password that the SESM web application uses to request service and group profiles from the RADIUS server. This password must match the value that was configured on the SSG host with the following command: <code>ssg service-password password</code> The service-password value must be the same on all SSGs. Default: <code>servicecisco</code>
	groupPassword	The password that the SESM web application uses to request group profiles from the RADIUS server. Default: <code>groupcisco</code>
Captive Portal	captureToURL	The URL of the NWSP application to which the captive portal application redirects the subscriber's browser. The captive portal application captures the original URL that was requested by the subscriber and forwards it to the SESM web application along with the redirect. The SESM web application can then honor the subscriber's originally requested URL after authentication occurs.

Table 4-4 Attributes in the Application MBean Configuration File (continued)

Object	Attribute Name	Explanation
Options context parameters	useIcons	Controls whether the application uses icons or text when it displays, on the web page, the services that a subscriber is authorized to use. Default: TRUE
	confirmAtService Logon	Controls whether or not the application prompts the user for confirmation before it acts on a request to start a service. Default: FALSE
	confirmAtService Logoff	Controls whether or not the application prompts the user for confirmation before it acts on a request to log off. Default: TRUE
	confirmAtAccount Logoff	Controls whether or not the application prompts the user for confirmation before it acts on a request to log off of the SESM application. Default: TRUE
	sessionTimeOut	The number of seconds of inactivity allowed before the application closes a session. This value overrides the timeout value in the nwsp.jetty.xml file. Default: 7200
Arbitrary context parameters	locations and brands	<p>Defines specific locations and brands and the attributes associated with each one.</p> <p>The NWSP application uses the location context parameter to define an initial URL and meaningful symbols (rivers and churches) related to the location. It uses the brand context parameter to define an initial URL and an email address.</p> <p>You can define additional context parameters, for any arbitrary use, by copying the format used in the nwsp.xml file to define the location and brand parameters. See the “Sample Application MBean Configuration File” section on page F-3 section for context parameter examples.</p> <p>To define a context parameter, use separate XML elements to define the following:</p> <ul style="list-style-type: none"> • The context parameter (for example, location) • The related subcontext parameters (for example, London, Paris, New York) • The attributes that are associated with each subcontext value (for example, URL values, river values, and church values) <p>For new context parameters to be meaningful, the SESM web application must be changed to do something with the new parameters. You can add new subcontext parameters (new locations or new brands) without changing the web application.</p>

Associating SSGs and Subscriber Requests

A typical SESM deployment consists of multiple SSGs. An SESM web application must know which SSG is handling each subscriber request. This section describes how to configure the associations between a subscriber request and its SSG. It includes the following topics:

- [Using Host Key with Identical SSG Configurations](#), page 4-27
- [Using Host Key with Varying SSG Configurations](#), page 4-28
- [Specifically Mapping SSGs to Subscriber Subnets](#), page 4-29
- [Format of Global and Subnet Attribute Elements](#), page 4-30

Using Host Key with Identical SSG Configurations

The easiest way to associate the correct SSG with each subscriber request is to use the host key port bundle feature on all SSGs, and configure certain attributes identically on all of the SSG hosts. We recommend using host key unless you need backward compatibility with SSD Release 2.5(1).

**Note**

To use the host key port bundle feature, the Cisco 6400 NRP must be running Cisco IOS Release 12.2(2)B or later and the SSG host key feature must be configured appropriately.

When the host key feature is enabled on an SSG, the SSG replaces the subscriber IP address in the request with a software token (or key) when it forwards the request to SESM. The SESM application uses this key in its responses to SSG, and the SSG does an internal translation to an actual host object.

The key is a unique combination of an SSG IP address from a range of IP addresses and a port number from a range of port numbers, as follows:

IP_address:port

The IP address and port ranges are configured on each SSG. The key uniquely identifies each subscriber currently logged on to SESM, even when multiple subscribers are using the same IP address.

To use the host key feature to associate SSGs, follow these procedures:

1. Enable and configure the host key feature on all of the SSGs, as described in the [“Configuring the Host Key Port Bundle Feature on SSG”](#) section on page B-2.
2. Configure the same values on all of the SSG hosts for the following attributes:
 - Port—The SSG port on the SSG host. Specify the port that SSG uses to listen for RADIUS requests from an SESM application. Configure this value on the SSG host with the following command:

```
ssg radius-helper authenticationPort
```

- Shared secret—The shared secret used for communication between SSG and an SESM application. Configure this value on the SSG host with the following command:

```
ssg radius-helper key
```

- Port bundle length—The number of bits that SSG uses for port bundling when the host key feature is enabled. This value must be 0 or 4. Configure this value on the SSG host with the following command:

```
ssg port-map length
```

3. Enter these globally configured values when the SESM installation program prompts you for them. These values are reflected in global elements in the <Configure name="SSG"> section of the application MBean configuration file, as the following example illustrates.

Example Using Host Key

When SSG has the host key feature enabled and configured, you can set all parameters globally.

```
<Configure name="com.cisco.aggbu:name=SSG">
<Call name="setGlobalAttribute"><Arg>PORT</Arg><Arg>1812</Arg></Call>
<Call name="setGlobalAttribute"><Arg>SECRET</Arg><Arg>cisco</Arg></Call>
<Call name="setGlobalAttribute"><Arg>MASK</Arg><Arg>255.255.255.255</Arg></Call>
<Call name="setGlobalAttribute"><Arg>BUNDLE_LENGTH</Arg><Arg>4</Arg></Call>
</Configure>
```

In this example, all SSGs are configured to use a port of 1812 and a shared RADIUS secret of `cisco`. The `BUNDLE_LENGTH` of 4 indicates that host key is configured on all SSGs.

The `MASK` attribute specifies the mask that SESM applies to the client (source) IP address in a received message to determine the client's subnet, and, from that, the SSG IP address. However, when host key is being used, the client (source) IP address is the SSG IP address. The SESM installation program provides the default mask of 255.255.255.255.

Using Host Key with Varying SSG Configurations

If host key is enabled on all SSGs, but some are configured differently, you can configure the global case and then specifically configure the exceptions. For example, if all but one SSG is assigned the same shared secret, you can configure the shared secret attribute globally, and then add one subnet entry to configure the different secret for the one SSG.

The installation program lets you provide one set of SSG global attribute values and one subnet entry. It records these attribute values in the <Configure name="SSG"> section of the application MBean configuration file, as illustrated in the following example.

Example Using Host Key with One Non-Complying SSG

In this example, host key is enabled on all SSGs. In addition, all SSGs are using the same port, secret, and client IP address mask, except that one SSG uses a different port. In this case, you can set all parameters globally, and then use one subnet entry to define:

- The client subnet being serviced by the SSG that uses the nonconforming port.
- The port value that overrides the globally-set port value.

In the following example, the SSG that services subnet 10.1.1.0 uses port 1245.

```
<Configure name="com.cisco.aggbu:name=SSG">
<Call name="setGlobalAttribute"><Arg>PORT</Arg><Arg>1812</Arg></Call>
<Call name="setGlobalAttribute"><Arg>SECRET</Arg><Arg>cisco</Arg></Call>
<Call name="setGlobalAttribute"><Arg>MASK</Arg><Arg>255.255.255.255</Arg></Call>
<Call name="setGlobalAttribute"><Arg>BUNDLE_LENGTH</Arg><Arg>4</Arg></Call>
<Call name="setSubnetAttribute"><Arg>10.1.1.0</Arg><Arg>255.255.255.0</Arg><Arg>PORT
</Arg><Arg>1245</Arg></Call>
</Configure>
```

Specifically Mapping SSGs to Subscriber Subnets

Each request arriving at an SESM web application contains a source, or client, IP address. SESM uses this client IP address to determine which SSG should handle each request.

- If the configuration file explicitly provides an SSG IP address for a subnet or a specific client IP address, SESM uses that SSG. You code an explicit IP address in a <subnet> element. The MASK value in the subnet element specifies whether the element applies to a subnet or to a specific subscriber IP address. The <IP> parameter in the subnet element specifies the SSG IP address.

For example, the following subnet entry explicitly sets the SSG IP address to 10.6.7.1 for subnet 10.2.0.0:

```
<Call name="setSubnetAttribute">
<Arg>10.2.0.0</Arg><Arg>255.255.0.0</Arg><Arg>IP</Arg><Arg>10.6.7.1</Arg></Call>
```

- If an explicit IP address for the SSG is not provided, SESM masks the subscriber's IP address to determine the SSG that should handle the request. Use masking as follows:
 - If host key is enabled—The host key feature replaces the original client IP address with the IP address of the SSG. (The port bundle key appended to the address preserves a unique identity for each subscriber). Since the client IP address is the SSG IP address, a global setting for MASK of 255.255.255.255 correctly results in the client IP address being used as the SSG IP address.
 - If the SSG uses the first IP address in a particular set of client subnets—Specify the mask that SESM web application can apply to the client IP address to derive the SSG IP address. For example, if, for all 10.x.0.0 client subnets, the SSG IP address is 10.x.0.1, you would specify a subnet of 10.0.0.0 and a mask of 255.0.0.0.
 - If the SSG IP is the first IP in all client subnets—You can set a global value for mask. For example, for all subscriber addresses x.y.z.n, if the SSG always has an IP address of x.y.0.1, then use a global mask of 255.255.0.0.



Note

Set the widest global or subnet mask possible. Each SSG IP address consumes some resources on the machine where the SESM application is running. (Each one uses an open file descriptor.) For example, even when the Cisco 6400 UAC is using host key, a mask of 255.255.255.0 is desirable, so that the SESM uses a single SSG IP address rather than 254 different SSG IP addresses. A mask of 255.255.255.255 is the least efficient, but it is the default setup.

Example Mapping Client Subnets to SSGs

In this example, host key is not being used. In this case, you must explicitly define the mapping from subscriber subnet to the SSG IP address.

```
<Configure name="com.cisco.aggbu:name=SSG">
<Call name="setGlobalAttribute"><Arg>PORT</Arg><Arg>1645</Arg></Call>
<Call name="setGlobalAttribute"><Arg>SECRET</Arg><Arg>cisco</Arg></Call>
<Call name="setGlobalAttribute"><Arg>MASK</Arg><Arg>255.255.255.255</Arg></Call>
<Call name="setGlobalAttribute"><Arg>BUNDLE_LENGTH</Arg><Arg>0</Arg></Call>
<Call name="setSubnetAttribute"><Arg>10.1.1.0</Arg><Arg>255.255.255.0</Arg><Arg>IP
</Arg><Arg>10.21.1.2</Arg></Call>
<Call name="setSubnetAttribute"><Arg>10.1.2.0</Arg><Arg>255.255.255.0</Arg><Arg>IP
</Arg><Arg>10.21.2.2</Arg></Call>
<Call name="setSubnetAttribute"><Arg>10.1.3.0</Arg><Arg>255.255.255.0</Arg><Arg>IP
</Arg><Arg>10.21.3.2</Arg></Call>
<Call name="setSubnetAttribute"><Arg>10.1.4.0</Arg><Arg>255.255.255.0</Arg><Arg>IP
</Arg><Arg>10.21.4.2</Arg></Call>
</Configure>
```

Format of Global and Subnet Attribute Elements

You can set the attributes that associate an SSG with subscriber requests globally, by client subnet, or for a specific client IP address, as follows:

- Global attribute elements—A global setting applies to all SSGs. For example, a global shared secret setting means that all SSGs are configured using the same secret. The global attributes are: PORT, SECRET, MASK, and BUNDLE_LENGTH.
- Subnet attribute elements—The subnet attributes apply to a specific subnet and override the global attribute value. The subnet attributes are optional; if any of them are not specifically coded, the global attribute value is used. Subnet attributes that you can supply are: PORT, SECRET, MASK, BUNDLE_LENGTH, and IP. The IP attribute is the IP address of the SSG for a specified subnet.

You can also specify some optional session information in a subnet entry, using context parameter values. See [Table 4-5](#).

- A specific client IP address is specified in a subnet element.

The format for the global attribute entries is illustrated in the following examples:

```
<Configure name="com.cisco.aggbu:name=SSG">
<Call name="setGlobalAttribute"><Arg>PORT</Arg><Arg>1645</Arg></Call>
<Call name="setGlobalAttribute"><Arg>SECRET</Arg><Arg>cisco</Arg></Call>
<Call name="setGlobalAttribute"><Arg>MASK</Arg><Arg>255.255.255.0</Arg></Call>
<Call name="setGlobalAttribute"><Arg>BUNDLE_LENGTH</Arg><Arg>0</Arg></Call>
</Configure>
```

The format for subnet entries is:

```
<Call name="setSubnetAttribute">
<Arg>subnetAddress</Arg>
<Arg>subnetMask</Arg>
<Arg>argumentName</Arg>
<Arg>argumentValue</Arg>
</Call>
```

Where:

subnetAddress is the subnet for which you are explicitly setting a value, overriding the globally set value.

subnetMask is the mask that can be applied to the subscriber's IP address to derive the subnet.

argumentName is the argument that you are explicitly setting. See [Table 4-5](#).

argumentValue is the value for *argumentName*. See [Table 4-5](#).

Table 4-5 Argument Names and Values for Subnet Entries

<i>argumentName</i> Value	<i>argumentValue</i> Explanation
PORT	The SSG port for the specified subnet. Overrides the globally-set SSG port.
MASK	The mask used on the subscriber's IP address to derive the subnet. Overrides the globally-set mask.
SECRET	The shared secret used between SESM and SSG. Overrides the globally-set shared secret.

Table 4-5 Argument Names and Values for Subnet Entries (continued)

<i>argumentName</i> Value	<i>argumentValue</i> Explanation
BUNDLE_LENGTH	<p>The host key bundle length used on the SSG. Overrides the globally-set bundle length.</p> <p>The bundle length is the number of bits that SSG uses for the port bundle feature. For example, a value of 4 indicates 16 bundled slots. A value of 0 indicates that the SSG is not using the host key and port bundle mechanism.</p> <p>This value must match the value used in the following command on the SSG host:</p> <pre>ssg port-map length</pre> <p>To determine how SSG has configured the port bundle length, use the show run command on the SSG host.</p>
IP	Explicitly sets the IP address for the SSG that services the specified <i>subnetAddress</i> .
SESSION_LOCATION	<p>The location associated with the specified subnet. Valid values are defined as subcontext parameters under the location context parameter in the nwsp.xml configuration file. The installed file defines the following locations: London, Paris, and New York.</p> <p>For the context parameters to have meaning, the SESM web application must support them. The NWSP application uses the location context parameter to define an initial URL and meaningful symbols related to the location.</p>
SESSION_BRAND	<p>The brand of service associated with the specified subnet. Valid values are defined as subcontext parameters under the brand context parameter in the nwsp.xml configuration file. The installed file defines the following brands: acme, cisco, silver, and gold.</p> <p>For the context parameters to have meaning, the SESM web application must support them. The NWSP application uses the brand context parameter to define an initial URL and an email address.</p>

Configuring RDP

This section describes how to configure the RDP application. The section includes the following topics:

- [RDP Modes, page 4-31](#)
- [RDP Attributes, page 4-32](#)

Also see the “[Sample RDP MBean Configuration File](#)” section on page F-13.

RDP Modes

RDP can run in two modes:

- **Non-proxy mode**—In this mode, RDP uses the DESS API to obtain authentication and authorization information from the LDAP directory.
- **Proxy mode**—In this mode, RDP sends authentication requests to a RADIUS server. It uses the DESS API to obtain authorization information from the LDAP directory.

You choose the mode during RDP installation. The content of the `rdp.xml` file is significantly different depending on the mode. Therefore, to change the mode, we recommend reinstalling the RDP component. (Choose a Custom installation to reinstall a single component.)

RDP Attributes

The MBean configuration file for RDP is located in:

```
rdp
  config
    rdp.xml
```

The `rdp.xml` file configures the following MBeans:

- **Logger**—The `com.cisco.aggbu.jmx.LoggerMBean` configures both logging and debugging tools. The logging tool logs RDP application activity. The debugging mechanism produces messages useful to developers in debugging applications. See the *Cisco Subscriber Edge Services Web Developer Guide* for more information about debugging an application.
- **RDPPacketFactory**—This MBean creates RDP packets that analyze and process requests from SSG. Each request becomes a series of packets. Each type of packet is handled by a different packet handler.
- **RDP**—The RDP MBean listens for requests sent through SSG.
- **ManagementConsole**—This MBean configures a management console port. Administrators can go to this console port on a web browser and see the currently set values for all attributes in all of the MBean configuration files.
- **AAA**—This MBean applies only when RDP is running in Proxy mode. In that mode, RDP is a RADIUS proxy server. The RDP AAA MBean defines the proxy server attributes.

[Table 4-6](#) explains the configurable attributes in these MBeans.

Table 4-6 Attributes in the RDP MBean Configuration File

MBean	Attribute Name	Explanation
Logger		See the description for Logger MBean in Table 4-4 on page 4-18 .
RDPPacketFactory		<p>The only attributes in this MBean that administrators are expected to change are the password attributes associated with service profile requests. These password attributes are used to identify a service request as one of the following: a single service request, a service group request, or a next hop table request. SSG sets the password in the request; RDP interprets the password. You must configure the values on both sides, as follows:</p> <ul style="list-style-type: none"> • On SSG, you set the values for these three passwords using IOS commands. • On RDP, you set the values for the three passwords as described here. <p>If the password in a request from SSG does not match one of the three values you set on the RDP side, the request is discarded.</p> <p>You can find the password attributes in this MBean by searching the file for the following string:</p> <pre data-bbox="448 764 623 785"><arg>PASSWORD:</pre> <p>Note There are no security implications to these attributes. It might be helpful to think of them as identifying keys, rather than passwords.</p> <p>The three password attributes are:</p> <ul style="list-style-type: none"> • ServiceRequest—Requests containing this password are handled by the ServiceRequest packet handler. The ServiceRequest packet handler uses the DESS API to obtain a list of authorized services for a subscriber. On the SSG side, set this password using the following command: <pre data-bbox="448 1062 911 1083">ssg service-password servicePassword</pre> • GroupRequest—Requests containing this password are handled by the GroupRequest packet handler. The GroupRequest packet handler forwards requests to a RADIUS server to obtain a list of authorized services for the group of which the subscriber is a member. Group requests are relevant only when RDP is configured in proxy mode. • NextHopRequest—Requests containing this password are handled by the ProxyNextHop packet handler. The Proxy NextHop packet handler passes authentication requests to the AAAMBean when the RDP is configured in proxy mode, or through DESS to the directory when the RDP is not in proxy mode. On the SSG side, set this password using the following command: <pre data-bbox="448 1394 1049 1415">ssg next-hop download nextHopTableName password</pre> <p>See Appendix E, “RDP Packet Handlers,” for more information about how RDP processes requests from SSG.</p>

Table 4-6 Attributes in the RDP MBean Configuration File (continued)

MBean	Attribute Name	Explanation
RDP	secret	Enter the RADIUS client shared secret to be used for communication between SSG and RDP. It must be a different value from the shared secret used for RDP to RADIUS communication. The installation program's displayed default is cisco.
	localIPAddress	Enter the IP address or host name of the RDP. Note This value cannot be localhost (127.0.0.1)
	localPort	Enter the port on which the RDP will listen. The installation program's displayed default is 1812.
	minThreads	Sets the minimum number of threads that RDP will maintain during periods of low load. RDP will always have system resources allocated for this number of threads. Installed default: 10
	maxThreads	The total number of simultaneous requests that the RDP can handle. If the RDP is receiving more requests than the current setting, and the RDP host machine is not processor-bound, then you can increase this number for a potential performance improvement. Installed default: 256
	maxIdleTimeMs	The number of milliseconds that a thread can remain idle before the system deallocates its resources. Installed default: 10000
ManagementConsole	See the description for " ManagementConsole " in Table 4-4 on page 4-18 .	

Table 4-6 Attributes in the RDP MBean Configuration File (continued)

MBean	Attribute Name	Explanation
AAA This MBean applies only when RDP is configured in Proxy mode.	Connection	The Configure tag for the AAA MBean includes a connection attribute whose value is either: <ul style="list-style-type: none"> • NextHop • Proxy The RDP proxy handlers use the connection name to identify the AAA server to proxy the request to.
	throttle	The maximum number of simultaneous requests that RDP can send to a RADIUS server. This is a RADIUS protocol attribute. The RADIUS protocol queues additional requests and issues them as the RADIUS server returns responses or timeout messages for previous requests. Installed default: 256
	timeOut	The number of seconds RDP waits before timing out RADIUS packets that it sends to the AAA server. Installed default: 4
	retryCount	The number of times RDP resends packets to the AAA server if no response is received. Installed default: 1
	primaryIP	Enter the IP address or the host name of the primary RADIUS AAA server that you want RDP to communicate with.
	primaryPort	Enter the port number on the primary RADIUS server host that the RADIUS server listens on.
	AAASecret	Enter the RADIUS client shared secret used between RDP and the RADIUS server. The shared secret must be the same for both servers. The installation program's displayed default value is <code>cisco</code> .
	secondaryIP	Enter the IP address or the host name of the secondary RADIUS server. If you are not using a secondary RADIUS server, enter the same value used for the primary server.
	secondaryPort	Enter the port number on the secondary RADIUS server host that the RADIUS server listens on. If you are not using a secondary RADIUS server, enter the same value used for the primary server.

Configuring CDAT

This section describes how to configure the CDAT application. The section includes the following topics:

- [Cookies Required, page 4-36](#)
- [CDAT Attributes, page 4-36](#)

Also see the “[Sample CDAT MBean Configuration File](#)” section on page F-16.

Cookies Required

Make sure that the cookies feature is enabled on the browser where you are running CDAT. If the CDAT application seems to log itself off unexpectedly, check your cookies setting.

CDAT Attributes

The CDAT MBean configuration file is located in:

```
cdat
  config
    cdat.xml
```

The cdat.xml file configures the following MBeans:

- **Logger**—The Logger MBean configures both logging and debugging tools. The logging tool logs CDAT application activity. The debugging mechanism produces messages useful for debugging.
- **ManagementConsole**—This MBean configures a management console port. Administrators can go to this console port on a web browser and see the currently set values for all attributes in all of the MBean configuration files.
- **CDAT**—The CDAT MBean configures resource attributes for the CDAT application.

[Table 4-7](#) explains the configurable attributes in this MBean.

Table 4-7 Attributes in the CDAT MBean Configuration File

MBean Name	Attribute Name	Explanation
Logger		See the description for Logger MBean in Table 4-4 on page 4-18 .
ManagementConsole		See the description for ManagementConsole in Table 4-4 on page 4-18 .
CDAT	sessionTimeout	The maximum period of inactivity allowed during a CDAT login, after which the user will be logged out. Values are in seconds. A negative value will prevent the user from ever being logged out. Changes will only take effect for subsequent logins. Default: 600
	maxVariables	The maximum number of page/page instance variables allowed for each CDAT session. This number affects how many pages can be visited before their state is lost, though it is not a one-to-one mapping. If you see many StateTimedOut errors, you should increase this number. Default: 40
	queryMaxResults	The maximum number of results to return from any one directory query. Changes will take immediate effect. A value of zero will remove any limits. Default: 500
	queryTimeout	The timeout (in milliseconds) for directory queries. Changes will take immediate effect. A value of zero will cause an infinite timeout. Default: 0

Configuring DESS

This section describes how to configure the DESS component. The section includes the following topics:

- [DESS Attributes, page 4-37](#)
- [Extending the Directory Schema and Installing Initial RBAC Objects, page 4-40](#)

Also see the “[Sample DESS MBean Configuration File](#)” section on page F-17.

DESS Attributes

The MBean configuration file for DESS is located in:

```
dess-auth
  config
    config.xml
```

This file applies to applications that incorporate the Dess and Auth APIs:

- SESM web applications deployed in DESS mode
- RDP

If these applications are installed on the same machine, the same config.xml file applies to both of them. If the applications are installed on different machines, the DESS component is installed with each of them, and each config.xml file can contain different attribute values.

The config.xml file for DESS contains the following MBean:

- Directory—The Directory MBean configures security, location, logging, and caching attributes for executing classes in the Dess and Auth APIs.

[Table 4-8](#) explains the configurable attributes in this MBean.

Table 4-8 Attributes in the Dess-Auth MBean Configuration File

Object Name	Attribute Name	Explanation
Directory MBean	factory	The full class name of the JNDI connection factory.
	poolSize	The number of active connections allowed to the LDAP server used for authorization.
	URL	The URL of the LDAP server used for authorization.
	principal	The name used when connecting to the LDAP server.
	credentials	The credentials (such as password) used for connecting to the LDAP server.
	context	The default LDAP context used for LDAP operations.
	alwaysGetAllAttributes	If set to true then all the attributes of an LDAP entry are returned for every query.
	traceFileName	The name of the directory log file.
	traceLevel	Should be one of: NONE, ERROR, BRIEF, VERBOSE, or DEBUG.
	printTraceToConsole	If set to true, the application sends trace messages to the console as well as writing them into the log file.
	stackTrace	If set to true, print a stack trace with each trace message.
cacheMaxObjects	Specifies the maximum number of software objects to hold in the cache. Objects represent subscribers, services, privileges, roles, and so on. When the cache contains cacheMaxObjects, old objects are deleted from cache, regardless of available cache space. Set this value high to allow the available cache space to be the determining factor for cache management. Installed default: 50000	

Table 4-8 Attributes in the Doss-Auth MBean Configuration File (continued)

Object Name	Attribute Name	Explanation
Directory MBean	cacheMinFreeMem	<p>Specifies the percentage of Java virtual memory that must remain available (that is, not used by the cache) after the application is loaded into memory. You can calculate the specific amount of memory available for the cache as follows:</p> $cacheSize = (JavaVM - applCodeSize) * (100\% - cacheMinFreeMem)$ <p>Where:</p> <p><i>JavaVM</i> is the maximum virtual memory size specified at application startup time with the <i>jvm</i> argument. The installed startup scripts use the following values:</p> <ul style="list-style-type: none"> • The startNWSP script uses 64 MB • The runrdp script uses 20 MB <p><i>applCodeSize</i> is the application size. The NWSP is approximately 18 MB. <i>cacheMinFreeMem</i> specifies the percentage of Java virtual memory that must remain available after the application is loaded into memory. The installed default value is 10. If NWSP and RDP applications are installed on the same machine, the same <i>cacheMinFreeMem</i> attribute value applies to both applications.</p> <p>For example, using all of the installed default values, the <i>cacheSize</i> for the NWSP application is 90% of 14 MB, or 12.6 MB:</p> $cacheSize = (32\text{ MB} - 18\text{ MB}) * (100\% - 10\%)$ <p>Installed default: 10</p>
	cacheSessionTimeout	<p>Specifies the timeout of inactive client sessions in seconds.</p> <p>Installed default: 600</p>
	cacheExpireInterval	<p>Specifies the interval in seconds after which the cache attempts to expire objects.</p> <p>Note Do not set this attribute to 0. A value of 0 causes <i>every</i> request to go to the directory, bypassing caching and any memory storage from a recent request for the same object. A value of 0 would degrade performance substantially.</p> <p>Installed default: 600</p>
	cacheObjectTimeout	<p>Specifies the number of seconds before objects time out.</p> <p>Installed default: 600</p>

Extending the Directory Schema and Installing Initial RBAC Objects

An SESM deployment running in DESS mode requires the following update activities on the LDAP directory:

- Extend the directory schema. These extensions include the `dess` and `auth` classes and attributes that will hold the SESM data. For more information about the extensions, see the *Cisco Distributed Administration Tool Guide*.
- Install initial RBAC objects. Some initial top-level rules and roles must be created in the directory before an administrator can log into CDAT and create additional objects.

The DESS installation process optionally performs these two update activities. If you did not choose these options during the installation, you must do them before running CDAT or an SESM application running in DESS mode.



Note

If the SESM components are distributed among different servers, which means that DESS might be installed in more than one location, you only need to perform these update activities one time against the LDAP directory.

To perform these updates after the initial DESS installation, use either of the following procedures:

- Use the installation process to perform the updates by running a custom installation of the DESS component.
- Perform the updates manually using native administration tools and commands.

Using a Custom Installation to Update the Schema and Install RBAC Objects

To use the custom installation process to extend the directory schema and install initial RBAC objects, follow these procedures:

-
- Step 1** Make sure the LDAP directory server is running.
 - Step 2** Make sure you know the following user IDs and passwords:
 - A user ID and password that allows you to update the directory schema
 - A user ID and password that allows you to update the container (organization and organizational unit) that you created for SESM data.
 - Step 3** Execute the SESM installation program on a server that has network access to the LDAP directory.
 - Step 4** When the installation program prompts for setup type, choose **Custom**.
 - Step 5** When the installation program prompts for the components to install, choose **DESS**.
 - Step 6** When the installation program prompts for directory connection information, provide correct information to access the directory. This includes the names of the organization and organizational unit you created to hold the SESM data.
 - Step 7** When the installation program displays the options, click the **Update schema** and **Install RBAC** check boxes.
-

Using LDIF Commands to Update the Directory Schema

To use LDIF commands to manually update the directory, follow these procedures:

-
- Step 1** Make sure the LDAP directory server is running.
 - Step 2** Make sure you have a user ID and password for the directory that allows you to update the schema.
 - Step 3** Obtain the required updates from the following location under your installation directory. Choose NDS or Netscape, depending on the LDAP directory you are using:

```
dess-auth
  schema
    NDS
    Netscape
```

You need to apply the contents of all of the ldf files found under the NDS or Netscape directories:

```
authattr.ldf
authclas.ldf
dessattr.ldf
dessclas.ldf
Policy15.ldf
```

- Step 4** Use the **ldapmodify** command to apply all of the preceding files to your directory. On successful completion, you have applied all of the required updates.
-

Using Manual Tools to Create Initial RBAC Objects

Some initial RBAC rules and roles must be loaded into the directory before any administrator can log into CDAT to create additional objects. The easiest way to load these top level objects is to allow the installation program to do it. However, you can also obtain them by loading the sample RBAC data files that are installed with DESS or by using your own data generating tool. See the *Cisco Distributed Administration Tool Guide* for information about the initial RBAC objects and loading the sample data.

Configuring Specific Features

Table 4-9 summarizes how to enable or disable some of the major features in an SESM deployment.

Table 4-9 Configuration Requirements for Specific Features

Feature	Configuration Requirements
Single sign-on	<p>On SESM Host</p> <p>Edit the following line in the application MBean configuration file (for example, nwsp/config/nwsp.xml):</p> <pre><Set name="singleSignOn" type="boolean">true</Set></pre>
Automatic connections	<p>On SSG Host</p> <p>No action required.</p> <p>On SESM Host—RADIUS Mode</p> <p>In RADIUS mode, the autoconnect feature is always on, regardless of parameter settings. In RADIUS mode, the SSG always performs automatic service connections for all services marked as auto connect in a subscriber's profile.</p> <p>In Subscriber's Profile—RADIUS Mode</p> <p>The A attribute in a subscriber's profile marks a service as one that should be automatically connected for the subscriber. For example:</p> <pre>user5 Password = "cisco" Service-Type = Framed-User, Account-Info = "Ainternet-green"</pre> <p>On SESM Host—DESS Mode</p> <p>In DESS mode, the SSG performs automatic connections if it has the service list. If SSG does not have the service list, the SESM application can perform the automatic connections. During RDP installation, the Add Services option configures RDP to either:</p> <ul style="list-style-type: none"> Return a service list to SSG—In this case, RDP includes the subscriber's service list and related information in replies to SSG, and SSG performs automatic connections for services marked for autoconnection in the subscriber's profile. <p>The service information consumes memory on the SSG host.</p> Not return a service list to SSG—In this case, SSG cannot perform automatic connections. The advantage to this configuration is that it saves memory on the SSG host. <p>In this case, you can configure the SESM application to perform automatic connections. The following line in the application MBean configuration file (for example, nwsp/config/nwsp.xml) controls whether the SESM web application performs automatic connections:</p> <pre><Set name="autoConnect" type="boolean">false</Set></pre> <p>Change the value to <code>true</code> to enable automatic connections by the SESM web application.</p> <p>To change the setting of the RDP service list option, either reinstall RDP or edit the configuration files to enable the correct set of packet handlers. See Appendix E, "RDP Packet Handlers," for information about the packet handlers that are used in the various configurations.</p> <p>In Subscriber's Profile—DESS Mode</p> <p>See the <i>Cisco Distributed Administration Tool Guide</i> for instructions about marking services for autoconnection in subscriber profiles.</p>

Table 4-9 Configuration Requirements for Specific Features (continued)

Feature	Configuration Requirements
Application Interface Options	<p>On SESM Host</p> <p>Edit the following lines in the application MBean configuration file (for example, nwsp/config/nwsp.xml):</p> <pre><Put name="useIcons" type="boolean">TRUE</Put> <Put name="confirmAtServiceLogon" type="boolean">FALSE</Put> <Put name="confirmAtServiceLogoff" type="boolean">TRUE</Put> <Put name="confirmAtAccountLogoff" type="boolean">TRUE</Put> <Put name="sessionTimeout" type="String">7200</Put></pre>
Captive portal	<p>On SSG Host</p> <p>Enable the TCP redirect feature using the http-redirect Cisco IOS commands.</p> <pre>ssg http-redirect group captive-portal-appl server 10.1.2.50 80 ssg http-redirect unauthorized-user group captive-portal-appl</pre> <p>Note The format of the http-redirect commands might change in the next release of Cisco IOS.</p> <p>On SESM Host</p> <p>To enable the captive portal application, choose the Run Captive Portal option when you install the NWSP application. This option sets the captiveportal.home Java system property in the generic startup script.</p> <p>To disable captive portal, edit the generic startup script (for example, jetty/bin/start.sh) and remove the captiveportal.home system property.</p> <p>To change the name of the captive portal application being called, edit the third argument in the Call element in the <i>container</i> MBean configuration file (for example, jetty/config/nwsp.jetty.xml):</p> <pre><!-- Captive portal web application --> <Call name="addWebApplication"> <Arg></Arg> <Arg></Arg> <Arg><SystemProperty name="install.root" default="."/>/captiveportal/docroot</Arg> <Arg><SystemProperty name="jetty.home" default="."/>/config/webdefault.xml</Arg> <Arg type="boolean">FALSE</Arg> </Call></pre> <p>To configure the SESM web application to which the captive portal application redirects subscribers, edit the following element in the <i>application</i> MBean configuration file (for example, nwsp/config/nwsp.xml):</p> <pre><Configure name="com.cisco.aggbu:name=captiveportal"> <Set name="captureToURL">http://localhost:80/decorate/pages/home.jsp</Set> </Configure></pre>
Walled garden	<p>In SSG</p> <p>Enter Cisco IOS ssg commands. For example:</p> <pre>>ssg open-garden opengarden-xyz.com >local-profile opengarden-xyz.com > attribute 26 9 251 "R10.1.1.0;255.255.255.255" > attribute 26 9 251 "D10.1.1.10" > attribute 26 9 251 "Oxyz.com;zap.com"</pre> <p>In SESM</p> <p>Create JSPs that implement the desired interface. See the <i>Cisco Subscriber Edge Services Manager Web Developer Guide</i> for information.</p>

Table 4-9 Configuration Requirements for Specific Features (continued)

Feature	Configuration Requirements
Retail pages and service ads	<p>In SSG No configuration required.</p> <p>In SESM Create JSPs that implement the desired interface. See the <i>Cisco Subscriber Edge Services Manager Web Developer Guide</i> for information.</p>
Host Key	<p>In SSG Enable the SSG host key feature using the Cisco IOS ssg port-map commands.</p> <pre data-bbox="300 611 1117 684"> ssg port-map enable ssg port-map source ip loopback 0 ssg port-map destination range lowPort to highPort ip SSDaddress </pre> <p>Disable the host key feature using the following command:</p> <pre data-bbox="251 758 508 783"> ssg port-map disable </pre> <p>In SESM Edit the BUNDLE_LENGTH attributes in the application MBean configuration file (for example, nwsp/config/nwsp.xml):</p> <pre data-bbox="251 936 1208 961"> <Call name="setGlobalAttribute"><Arg>BUNDLE_LENGTH</Arg><Arg>0</Arg></Call> </pre> <p>In the SSG MBean, the BUNDLE_LENGTH attributes must match the bundle lengths specified on the SSG side.</p> <p>Note A BUNDLE_LENGTH of zero indicates that host key is not being used.</p>