



## 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 6-1](#)
- [Configuring the J2EE Jetty Container, page 6-7](#)
- [Configuring an SESM Portal Application, page 6-14](#)
- [Configuring RDP, page 6-30](#)
- [Configuring CDAT, page 6-36](#)
- [Configuring SPE, page 6-37](#)
- [Configuring Specific Features, page 6-41](#)
- [Configuring a Customized SESM Application, page 6-44](#)

### Configuration Overview

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

- [Changing Configuration Information, page 6-1](#)
- [Configuration Technology, page 6-2](#)
- [Configuration File Summary, page 6-3](#)

### Changing Configuration Information

The installation program assigns initial values to all of the key attributes in the MBean configuration files, using a combination of default values and values you provide during the install. To change these initial values, administrators can manually edit 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 LDAP mode, you also must stop and restart RDP. See [Chapter 7, “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 File Summary

This section includes the following topics:

- [J2EE Configuration Files, page 6-3](#)
- [MBean Configuration Files, page 6-4](#)
- [MBean Configuration File Format, page 6-5](#)
- [Java System Properties in the MBean Configuration Files, page 6-6](#)

### 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.

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 6-1](#) shows the J2EE configuration files used to configure SESM web portals.

**Table 6-1 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	applicationName docroot WEB-INF web.xml	This file defines J2EE application parameters, including parameters related to Java Server Pages (JSPs).  There is a separate <code>web.xml</code> file for each web application.

## MBean Configuration Files

Table 6-2 lists all of the MBean configuration files in an SESM deployment.

**Table 6-2 Summary of MBean Configuration Files**

Component	File Path and Name	Description
Container (Jetty)	jetty config nwsp.jetty.xml wap.jetty.xml pda.jetty.xml cdat.jetty.xml captiveportal.jetty.xml messageportal.jetty.xml yourapp.jetty.xml	These files configure the Jetty server instance associated with each application. These files configure: <ul style="list-style-type: none"> <li>Logging and debugging for the Jetty server. This log filename is <i>nnn.jetty.log</i>.</li> <li>HTTP listener, which configures: <ul style="list-style-type: none"> <li>The application that is running in the container and the application port.</li> <li>The web server's standard HTTP request log. This log filename is <i>nnn.request.log</i>.</li> </ul> </li> </ul>
SESM web portals	nwsp config nwsp.xml wap config wap.xml pda config pda.xml captiveportal config captiveportal.xml messageportal config messageportal.xml	This file configures: <ul style="list-style-type: none"> <li>SESM deployment options</li> <li>Communication between an SESM web application and SSG</li> <li>Communication between an SESM web application and RADIUS servers</li> <li>Logging and debugging for the SESM web application. The log filename is <i>nnn.application.log</i>.</li> <li>Captive portal options and behavior. See <a href="#">Chapter 8, "Deploying a Captive Portal Solution,"</a> for more information.</li> </ul>
RDP	rdp config rdp.xml	This file configures: <ul style="list-style-type: none"> <li>RDP options, including 3-key authentication and packet handlers</li> <li>RDP communication with SSG</li> <li>Optionally, RDP communication with a RADIUS server</li> <li>Logging and debugging for RDP</li> </ul>
CDAT	cdat config cdat.xml	This file configures: <ul style="list-style-type: none"> <li>System resource usage for the CDAT application</li> <li>Logging and debugging for the CDAT application</li> </ul>
SPE	dess-auth config config.xml	This file configures attributes used by the executing classes in the SPE application programming interfaces (APIs). The SPE APIs provide the underlying support for communication between an LDAP directory and the RDP, CDAT, and SESM portal applications. If these applications are installed on the same machine, the same config.xml file applies to all of the applications.  This file configures LDAP directory security and connection attributes, SPE caching, and SPE logging.

## 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.
- 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, and Call elements after any Arg elements. These nested elements are applied to the created or returned object.
- A <SystemProperty> tag might appear inside a <Set> or <Call> tag. See the next section (“[Java System Properties in the MBean Configuration Files](#)”) for more information.

## Java System Properties in the MBean Configuration Files

The MBean configuration files use Java system properties as the value for some attributes. The value of a Java system property is set as follows:

1. 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 system property.
2. You can specify a value in the startup script. For example, the following lines from the installed start scripts (START.sh or START.cmd) set some system properties.

```
$JAVA -Xmx64m -Xms64m \
  -classpath $CLASSPATH \
  -Djetty.home=$JETTYDIR \
  -Dapplication.home=$APPDIR \
  -Dapplication.log=$LOGDIR \
  -Dapplication.portno=$PORTNO \
```

For a description of how the start script derives values for the environment variables used in the assignments, see [Table 7-1 on page 7-5](#).

3. If a value is not specified by either of the above methods at run time, the application uses a default value specified in the MBean configuration file.

In the MBean configuration files, the <SystemProperty> tag might appear inside a <Set> or <Call> tag. The format is:

```
<SystemProperty name="propertyName" default="value" />
```

Where:

*propertyName* is the Java system property name.

*value* is the default value used if no value is assigned at run time.



**Note**

If a system property is defined in the startup script, the default values in the MBean configuration files are not used, unless you delete the setting in the startup script.

# Configuring the J2EE Jetty Container

This section includes the following topics:

- [Containers and Applications, page 6-7](#)
- [J2EE Container Configuration Attributes, page 6-8](#)

## 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 the following advantages:

- 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 adding more MBeans.
- Defines the process that each MBean is configuring. For example, both the container and the application have logging and debugging MBeans.

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

## J2EE Container Configuration 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**—Enables the Jetty server logging mechanism and configures the information to appear in the jetty log files.
- **DebugMBean**—Enables or disables the Jetty server debugging mechanism.
- **Jetty**—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 6-3](#) describes the attributes in the container MBean configuration files. For an example file, see the [“Sample Container MBean Configuration File”](#) section on page F-1.



Table 6-3 Attributes in the Container MBean Configuration Files

Object Name	Attribute Name	Explanation
Log	append	Indicates if messages overwrite existing contents (false) or are appended to the existing file (true). Installed default: true
	filename	Specifies the log filename and path, as follows: <i>application.log/yyyy_mm_dd.jetty.log</i> Where: <ul style="list-style-type: none"> <li><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. <a href="#">Table 7-1 on page 7-5</a> describes how the start script sets <i>application.log</i>.</li> <li><i>yyyy_mm_dd</i>—Is the year, month, and day that the file was created.</li> <li><i>.jetty.log</i>—Is a constant identifying the Jetty log files.</li> </ul>
	logTimezone	Installed default: empty
	logDateFormat	Controls the format of the date stamp in the log messages. Installed default: yyyyMMdd:HHmms.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
	logTags	Installed default: false
	logTimeStamps	Installed default: false
retainDays	Indicates the number of days to keep an old log file before deleting it. Installed default: 31	

Table 6-3 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 6-3 Attributes in the Container MBean Configuration Files (continued)

Object Name	Attribute Name	Explanation
Jetty 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><b>Note</b> 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.</p> <p>Installed value: The SESM installation program sets the <code>application.portno</code> in the startup script to the NWSP port that you provided during the installation process.</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 6-3 Attributes in the Container MBean Configuration Files (continued)


Object Name	Attribute Name	Explanation
Jetty 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><b>Note</b> 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 path name 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> <p><i>jetty.home</i>—Is a Java system property. The NWSP start script derives the value of <i>jetty.home</i> from an expected (installed) directory structure. To change the value of <i>jetty.home</i>, edit the start script. Unless you alter the start script, the default value for <i>jetty.home</i> specified in this MBean configuration file is ignored at run time.</p> <p> <b>Caution</b> A keystore file is required for deployments that use HTTPS. HTTPS does not function without a valid keystore file. The nwspkeystore file included with the SESM installation works, but you should replace it with a keystore valid for your specific deployment. See the “<a href="#">HTTPS Description</a>” 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 6-3 Attributes in the Container MBean Configuration Files (continued)

Object Name	Attribute Name	Explanation
Jetty 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: <code>application.log/yyyy_mm_dd.request.log</code> Where: <ul style="list-style-type: none"> <li>• <code>application.log</code>—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 <a href="#">Table 7-1 on page 7-5</a> for a description of how the start script sets <code>application.log</code>.</li> <li>• <code>yyyy_mm_dd</code>—Is the year, month, and day that the file was created. The installation program uses the appropriate path name delimiter for the installation platform.</li> <li>• <code>.request.log</code>—Is a constant identifying an HTTP request file.</li> </ul>
	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 6-3 Attributes in the Container MBean Configuration Files (continued)

Object Name	Attribute Name	Explanation
Jetty— <Call AddWebApplication>  This call adds the NWSP application to run on the web server.	These are positional arguments.	AddWebApplication has five positional arguments: <ol style="list-style-type: none"> <li>1. The first positional argument specifies the virtual host name for the web server application.</li> <li>2. The second positional argument specifies the context path for locating the web server application. For example, / or /pathname/*.</li> <li>3. The third positional argument identifies the location of the application. The value is: <i>application.home/docroot</i> Where: <i>application.home</i> is a Java system property.</li> <li>4. The fourth positional argument identifies the location of the webdefault.xml file for this application. The value is: <i>jetty.home/config/webdefault.xml</i> Where: <i>jetty.home</i> is a Java system property</li> <li>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, because NWSP and CDAT are not WAR files.</li> </ol> The first three arguments define the location of the web server application. <i>host/context/application</i>  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.

## Configuring an SESM Portal Application

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

- [SESM Application Attributes, page 6-14](#)
- [Associating SSGs and Subscriber Requests, page 6-25](#)

## 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 traces business events in the SESM portal. 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.
- **SESM**—This MBean configures SESM features and options, including the SESM mode.
- **SESMDemoMode**—This MBean configures SESM in demo mode.
- **DESSMode**—This MBean configures SPE attributes used by the SESM application.
- **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.
- **WebApp**—The WebApp MBean configures options of the SESM portal application, including:
  - Attributes that control the behavior of the application
  - Attributes that control captive portal service redirections handled by NWSP
  - Context parameters, which are used by an application for any arbitrary reason. The `nwsp.xml` file contains an example of using context parameters to control web page content based on location.

[Table 6-4](#) explains the configurable attributes in the MBeans listed above. For an example file, see the [“Sample Application MBean Configuration File”](#) section on page F-3.

Table 6-4 Attributes in the Application MBean Configuration File

Object	Attribute Name	Explanation
Logger	debug	<p>Turns debugging on or off. Note that logging is on regardless of this value.</p> <ul style="list-style-type: none"> <li>• False—The application produces trace messages but not debug messages. The trace messages record business activity performed by the SESM portal. This setting is the normal, recommended setting for production environments. The trace messages provide important information for diagnosing configuration problems.</li> <li>• True—The application produces trace and debug messages. This setting is intended for development environments to debug portal code behavior. The logging of debug messages can affect performance; hence, this setting is not recommended for production environments.</li> </ul> <p>The following parameters control the contents of debug messages that the application generates: logFrame, logStack, logThread, debugPatterns, and debugThreads.</p> <p>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"> <li>• MAX</li> <li>• MED</li> <li>• LOW</li> </ul> <p>Installed default: LOW</p>
	logDateFormat	<p>Specifies format of dates in the log file.</p> <p>Installed default: yyyyMMdd:HHmmss.SSS</p>



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

Object	Attribute Name	Explanation
Logger (continued)	logFile	<p>Specifies the filename and location for the logging (tracing) of business events performed by the SESM application. The installed default is:</p> <p style="text-align: center;"><i>application.log/yyyy_mm_dd.application.log</i></p> <p>Where:</p> <ul style="list-style-type: none"> <li>• <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 <a href="#">Table 7-1 on page 7-5</a> for a description of how the start script sets <i>application.log</i>.</li> <li>• <i>yyyy_mm_dd</i> —Is the year, month, and day that the file was created.</li> <li>• <i>application.log</i>—Is a constant identifying the application log files.</li> </ul>
	logFrame	<p>Controls whether or not to log the calling member function.</p> <p>Installed default: false</p>
	logStack	<p>Controls whether or not to log stack traces.</p> <p>Installed default: false</p>
	logThread	<p>Controls whether or not to log thread IDs.</p> <p>Installed default: true</p>
	logToErr	<p>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.</p> <p>Installed default: true</p>
	trace	<p>Controls whether or not to log trace messages. These messages indicate entry and exit to code points.</p> <p>Installed default: true</p>
	warning	<p>Controls whether or not to log warning messages (nonfatal exceptions).</p> <p>Installed default: true</p>

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

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><b>Note</b> 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>&lt;Configure jmxname="com.cisco.aggbu:name=ManagementConsole"&gt; &lt;Call name="start" /&gt; &lt;/Configure&gt;</pre> <p>The port value is 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"> <li>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>.</li> <li>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>.</li> </ol>

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

Object	Attribute Name	Explanation
SESM	mode	<p>An SESM portal runs in one of the following modes.</p> <ul style="list-style-type: none"> <li>• RADIUS—In this mode, the SESM web application communicates with SSG and a RADIUS server.</li> <li>• LDAP—In this mode, the SESM web application communicates with SSG and an LDAP directory.</li> <li>• Demo—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 directory are not available.</li> </ul> <p>The value for mode is a Java system property named:</p> <pre>sesm.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. The installation program sets the default value to match the type of installation you perform (RADIUS, LDAP, or Demo.)</p> <p>To change the mode, you can:</p> <ul style="list-style-type: none"> <li>• Reinstall the software.</li> <li>• Edit the MBean configuration files, changing the mode and other attributes, as appropriate.</li> <li>• 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"> <li>– On Solaris: <code>jetty/bin/startNWSP.sh -mode {Demo   RADIUS   LDAP}</code></li> <li>– On Windows: <code>jetty\bin\startNWSP.cmd {Demo   RADIUS   LDAP}</code></li> </ul> </li> </ul> <p><b>Note</b> The best way to change the SESM mode is to reinstall 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 SPE component.</p>

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

Object	Attribute Name	Explanation
SESM (continued)	singleSignOn	<p>Enables or disables the single sign-on feature.</p> <ul style="list-style-type: none"> <li>• True—Subscribers only need to authenticate during a session. Single sign-on offers the following advantages: <ul style="list-style-type: none"> <li>– Subscribers can stop the browser or navigate away from the SESM portal pages, and then return to the SESM pages later and not be required to reauthenticate.</li> <li>– Subscribers do not need to reauthenticate if SESM automatic memory management clears sessions from the SESM portal.</li> <li>– Point-to-point protocol (PPP) clients do not need to authenticate to the SESM portal. Instead, the SESM portal uses the PPP authenticated identity from SSG.</li> </ul> </li> <li>• False—Subscribers are required to reauthenticate for all of the cases described above.</li> </ul> <p>Installed default: true</p>
	autoConnect	<p>Specifies if SESM should send connection requests to SSG for the services marked for auto connection in the subscriber's profile. Values are:</p> <ul style="list-style-type: none"> <li>• False—SESM does not send connection requests to SSG</li> <li>• True—SESM sends connection requests to SSG</li> </ul> <p>In RADIUS mode, set this attribute to false, because SSG automatically makes the connections immediately after authentication. You do not need SESM to request those connections.</p> <p>In LDAP mode, the SSG performs automatic connections if it obtains a service list from the RDP. If SSG does not obtain the service list from RDP, you should set this attribute to true.</p> <p>The Add Services option, which is set during RDP installation, controls whether or not the RDP returns a service list to SSG. The Add Services option configures RDP to either:</p> <ul style="list-style-type: none"> <li>• Return a service list to SSG—SSG performs automatic connections for services marked as auto connect in a subscriber's profile. In this configuration, set the autoConnect attribute to false.</li> <li>• Not return a service list to SSG—SSG cannot perform automatic connections. The advantage to this configuration is that it saves memory on the SSG device. In this configuration, set the autoConnect attribute to true.</li> </ul>
	profileCachePeriod	<p>Specifies the time in seconds that a service or group object must be idle in the cache before its resources are deallocated from memory.</p> <p>Installed default: 600</p>
	sessionCachePeriod	<p>The minimum time in seconds that an SESM session can be in memory without being accessed.</p> <p>Installed default: 1200</p>

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

Object	Attribute Name	Explanation
SESM (continued)	confirmMutex Disconnect	<p>Controls the action of the SESM portal if a subscriber is currently connected to a service in a mutually exclusive service group and then selects another service in that group.</p> <ul style="list-style-type: none"> <li>• True—The SESM portal displays an error message to the subscriber stating that the current service must be disconnected before selecting the newly selected service.</li> <li>• False—The SESM portal sends a request to SSG to disconnect the current service before sending the request to connect to the newly selected service.</li> </ul> <p>Installed default: false</p>
SESMDemoMode	demoDataFile	<p>Specifies the file that contains data for demo mode. The installed value is:</p> <p><i>application.home/config/demo.txt</i></p> <p>Where:</p> <p><i>application.home</i> is a Java system property</p> <p>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.</p>
DESSMode	tokenCheckInterval	<p>The time in seconds between checking the authorization tokens.</p> <p>Default: 300 seconds</p>
	tokenMaxAge	<p>The length of time in seconds a token can remain in cache without being used before it is deleted.</p> <p>Default: 600 seconds</p>

Table 6-4 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 <b>show run</b> command on the SSG host.	throttle	<p>The global value for the maximum number of simultaneous requests that SESM portals can send to an SSG. The RADIUS protocol queues additional requests and issues them as the SSG returns responses or timeout messages for previous requests.</p> <p>If set correctly, this throttle attribute prevents the situation in which the SSG receives requests at a faster rate than it can handle, causing the SESM application to time out waiting for responses. Set the throttle value according to the ability of the SSG device to process access requests from a client. Try adjusting this value lower if the SESM portal is timing out while waiting for responses from the SSG.</p> <p>You cannot override the global value. (The same throttle value applies to all SSGs.)</p> <p>Installed default: 20</p>
	PORT	<p>The global value for RADIUS ports on the SSG hosts. This value must match the value that was configured on the SSG device using the following command:</p> <pre>ssg radius-helper authenticationPort</pre> <p>You can create subnet entries in the MBean configuration file to override this global value for specific SSGs.</p>
	TIMEOUTSECS	<p>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.</p> <p>Installed default: 5</p>
	RETRIES	<p>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.</p> <p>Installed default: 3</p>
	SECRET	<p>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 device using the <b>ssg radius-helper key</b> command.</p> <p>You can create subnet entries in the MBean configuration file to override this global value for specific SSGs.</p>
	MASK	<p>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.</p> <p>You can create subnet entries in the MBean configuration file to override this global value for specific subnets.</p>
	BUNDLE_LENGTH	<p>The global value for the port bundle length that SSGs use when the port-bundle host key feature is enabled.</p> <p>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:</p> <pre>ssg port-map length</pre> <p>Default: You set this value during installation.</p>

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

Object	Attribute Name	Explanation
SSG global attributes (continued)	PORT_BUNDLE_ HOST_KEY_ SWITCH	<p>The global value indicating whether or not the port-bundle host key feature is enabled on the SSGs. If BUNDLE_LENGTH is zero, then the value of this switch is important.</p> <ul style="list-style-type: none"> <li>• True—The SSGs have port-bundle host key enabled with a 0 bundle length.</li> <li>• False—The SSGs do not have port-bundle host key enabled.</li> </ul> <p><b>Note</b> If BUNDLE_LENGTH is non-zero, this switch is ignored, because a nonzero value implies the use of the host key feature.</p>
SSG Subnet entries  Use subnet entries to override the global values or to map client subnets to specific SSGs when the port-bundle host key feature is not being used.	Subnet entries use positional arguments.	<p>The call to <code>setSubnetAttribute</code> has four positional arguments:</p> <ol style="list-style-type: none"> <li>1. <i>subnetAddress</i> is the subnet for which you are explicitly setting a value, overriding the globally set value.</li> <li>2. <i>subnetMask</i> is the mask that can be applied to the subscriber's IP address to derive the subnet.</li> <li>3. <i>argumentName</i> is the argument that you are explicitly setting.</li> <li>4. <i>argumentValue</i> is the value for <i>argumentName</i>.</li> </ol> <p>See the <a href="#">“Associating SSGs and Subscriber Requests”</a> section on page 6-25 for more information.</p>

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

Object	Attribute Name	Explanation
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 that identifies the type of request. Values are: <ul style="list-style-type: none"> <li>ServiceProfile—The MBean for this connection type includes the servicePassword attribute.</li> <li>ServiceGroupProfile—The MBean for this connection type includes the serviceGroupPassword attribute.</li> </ul>
	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: <code>cisco</code> .
	secondaryIP	The IP address or host name of the secondary AAA server. If you are not using a secondary RADIUS server, reenter the primary server.
	secondaryPort	The port number that the secondary RADIUS server listens on. If you are not using a secondary server, reenter the primary server.  Default: 1812
	servicePassword	The password that the SESM web application uses to request service profiles from the RADIUS server. It must match the service password values used in the service profiles in the RADIUS database. It must also 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 of your SSGs.  Default: <code>servicecisco</code>
serviceGroup Password	The password that the SESM web application uses to request group profiles from the RADIUS server. It must match the service group password values used in the service group profiles in the RADIUS database.  Default: <code>groupcisco</code>	



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

Object	Attribute Name	Explanation
WebApp	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
	credentialMax Length	Controls the maximum length of user names and passwords. Default: 30
	serviceNotGivenURI defaultURI serviceSubscriptioURI serviceStartURI serviceLogonURI	These attributes are related to the captive portal solution. See <a href="#">Table 8-6 on page 8-20</a> for explanations of these attributes.
	Call addDimension	The addDimension call adds any arbitrary property to an incoming request. See the <a href="#">“Configuration-based Location and Brand Awareness” section on page 6-44</a> for more information.

## 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 Port-bundle Host Key with Identical SSG Configurations, page 6-25](#)
- [Using Port-bundle Host Key with Varying SSG Configurations, page 6-27](#)
- [Specifically Mapping SSGs to Subscriber Subnets, page 6-27](#)
- [Global and Subnet Attribute Elements, page 6-28](#)

### Using Port-bundle Host Key with Identical SSG Configurations

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

**Note**

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

When the port-bundle 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 port-bundle host key feature to associate SSGs, follow these procedures:

1. Enable and configure the port-bundle 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 device 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 device with the following command:
 

```
ssg radius-helper key
```
  - Port bundle length—The number of bits that SSG uses for port bundling when the port-bundle host key feature is enabled. This value must be 0 or 4. Configure this value on the SSG device 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 Port-Bundle Host Key**

When the port-bundle host key feature is 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 port-bundle 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 a host key is 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 Port-bundle Host Key with Varying SSG Configurations

If port-bundle 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 Port-bundle Host Key with One Noncomplying SSG

In this example, port-bundle 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 port-bundle host key is enabled—The port-bundle 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 SSG is using port-bundle 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, port-bundle 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>
```

## 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 6-5 on page 6-29](#).

- 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>255.255.0</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 ([Table 6-5](#)).

*argumentValue* is the value for *argumentName* ([Table 6-5](#)).

**Table 6-5** 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.
BUNDLE_LENGTH	The host key bundle length used on the SSG. Overrides the globally-set bundle length.  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 port-bundle host key mechanism.  This value must match the value used in the following command on the SSG host:  <pre>ssg port-map length</pre> To determine how SSG has configured the port bundle length, use the <b>show run</b> command on the SSG host.
IP	Explicitly sets the IP address for the SSG that services the specified <i>subnetAddress</i> .

*Table 6-5 Argument Names and Values for Subnet Entries (continued)*

<i>argumentName</i> Value	<i>argumentValue</i> Explanation
SESSION_LOCATION	The location associated with the specified subnet. Valid values are defined as arbitrary properties in the WebApp MBean. See the <a href="#">“Configuration-based Location and Brand Awareness”</a> section on page 6-44 for more information.
SESSION_BRAND	The brand of service associated with the specified subnet. Valid values are defined as arbitrary properties in the WebApp MBean. See the <a href="#">“Configuration-based Location and Brand Awareness”</a> section on page 6-44 for more information.

## Configuring RDP

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

- [RDP Modes, page 6-30](#)
- [RDP Attributes, page 6-30](#)

## RDP Modes

RDP can run in two modes:

- Non-proxy mode—In this mode, RDP uses the SPE 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 SPE 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 features. The logging feature 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.
- 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.

- **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 MBean**—The RDP MBean configures the listener for requests sent through SSG. It configures the SESM 3-key authentication feature.
- **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 6-6](#) explains the configurable attributes in these MBeans. For an example file, see the [“Sample RDP MBean Configuration File”](#) section on page F-13.

Table 6-6 Attributes in the RDP MBean Configuration File

MBean	Attribute Name	Explanation
Logger		See the Logger object in <a href="#">Table 6-4 on page 6-16</a> .
ManagementConsole		See the ManagementConsole object in <a href="#">Table 6-4 on page 6-16</a> .
RDPPacketFactory	<b>Note</b>	<p>RDP uses password values, described below, to identify the type of request it receives and determine how to handle the request. Each of the three password values <i>must</i> be unique.</p> <p>The only attributes in this MBean that administrators must 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"> <li>• On SSG, you set the values for these three passwords using IOS commands.</li> <li>• On RDP, you set the values for the three passwords as described here.</li> </ul> <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>To find the password attributes in this MBean, search the file for the following string:</p> <pre>&lt;arg&gt;PASSWORD:</pre> <p>No security implications exist for these attributes. It might be helpful to view the attributes as identifying keys, rather than passwords.</p> <p>The three password attributes are:</p> <ul style="list-style-type: none"> <li>• <b>ServiceRequest</b>—Requests containing this password are handled by the ServiceRequest packet handler. The ServiceRequest packet handler uses the SPE API to obtain a list of authorized services for a subscriber. This password must match:</li> <li>• <b>GroupRequest</b>—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.</li> <li>• <b>NextHopRequest</b>—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 SPE to the directory when the RDP is not in proxy mode. On the SSG side, set this password using the following command:</li> </ul> <pre>ssg next-hop download nextHopTableName password</pre> <p>See one of the following sections for more information about matching these password values to values configured elsewhere in an SESM deployment:</p> <ul style="list-style-type: none"> <li>• <a href="#">Communication Attributes for LDAP Mode, page 9-6</a></li> <li>• <a href="#">Communication Attributes for LDAP Mode with RDP in Proxy Mode, page 9-10</a></li> </ul> <p>See <a href="#">Appendix E, “RDP Packet Handlers,”</a> for more information about how RDP processes requests from SSG.</p>



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

MBean	Attribute Name	Explanation
RDP MBean	secret	<p>The useClientList attribute, which appears later in this MBean, affects how the secret attribute is used.</p> <ul style="list-style-type: none"> <li>If the useClientList attribute is false, the secret is the shared secret for communication between all of the SSGs and RDP. This value must match the value configured on the SSG devices, using the following command: <pre>radius-server key SharedSecret</pre> <p>The same shared secret value must be configured on all of the SSGs.</p> </li> <li>When the useClientList attribute is true, this secret attribute is ignored. Instead, you configure a specific shared secret for each client (each SSG).</li> </ul> <p>The installation program's displayed default is <code>cisco</code>.</p>
	localIPAddress	<p>Enter the IP address or host name of the RDP.</p> <p><b>Note</b> This value cannot be localhost (127.0.0.1)</p>
	localPort	<p>Enter the port on which the RDP will listen. The installed value is a Java system property:</p> <pre>application.portno</pre> <p>The installation program sets the value of <i>application.portno</i> in the RDP startup script to whatever you specified during installation. To change the value of <i>application.portno</i>, edit the start script.</p> <p>The installation program's displayed default is 1812.</p>
	minThreads	<p>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.</p> <p>Installed default: 10</p>
	maxThreads	<p>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.</p> <p>Installed default: 256</p>
	maxIdleTimeMs	<p>The number of milliseconds that a thread can remain idle before the system deallocates its resources.</p> <p>Installed default: 10000</p>
	threeKeyAuth	<p>Specifies whether to use the 2-key or 3-key method to authenticate a subscriber.</p> <ul style="list-style-type: none"> <li>True—Turns on 3-key authentication, which authenticates a subscriber using the user name and password, plus one additional attribute as specified in the authAttribute attribute.</li> <li>False—Turns off 3-key authentication. RDP authenticates using a user name and password.</li> </ul> <p>Installed default: false</p>

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

MBean	Attribute Name	Explanation
RDP (continued)	authAttribute	Specifies the RADIUS attribute number to use for the additional key when 3-key authentication is turned on. Any standard RADIUS attribute can be used. Typical values are: <ul style="list-style-type: none"> <li>• 30—CALLED_STATION_ID (APN)</li> <li>• 31—CALLING_STATION_ID (MSISDN)</li> <li>• 32—NAS_IDENTIFIER</li> </ul>
	useClientList	Turns the RDP restricted client feature on or off. Values are: <ul style="list-style-type: none"> <li>• True—The RDP accepts requests only from the clients specified in an addClient call later in this MBean. RDP clients are SSGs.</li> <li>• False—The RDP accepts requests from any client (any SSG).</li> </ul> You set the initial value of this attribute during RDP installation.
addClient	These are positional arguments.	The addClient call adds a client to the client list when the useClientList attribute is true. RDP clients are SSGs. You can add more clients by adding more addClient elements to the rdp.xml file. The addClient call has three positional arguments: <ol style="list-style-type: none"> <li>1. The first positional argument specifies a client name. This value is used in logs and traces and does not have to match any other configured value.</li> <li>2. The second positional argument specifies the client IP address.</li> <li>3. The third positional argument specifies the shared secret for communication between RDP and this client. It must match the shared secret configured on the SSG device using the following command: <pre>radius-server key SharedSecret</pre> </li> </ol>

Table 6-6 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"> <li>• NextHop</li> <li>• Proxy</li> </ul> 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 6-36](#)
- [CDAT Attributes, page 6-36](#)

## 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 6-7](#) explains the configurable attributes in this MBean. For an example file, see the [“Sample CDAT MBean Configuration File”](#) section on page F-16.

Table 6-7 Attributes in the CDAT MBean Configuration File

MBean Name	Attribute Name	Explanation
Logger		See the Logger object in <a href="#">Table 6-4 on page 6-16</a> .
ManagementConsole		See the ManagementConsole object in <a href="#">Table 6-4 on page 6-16</a> .
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 SPE

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

- [SPE Attributes, page 6-37](#)
- [Extending the Directory Schema and Loading Initial RBAC Objects, page 6-40](#)

Also see the “[LDAP Directory Configuration Requirements](#)” section on [page 5-4](#), which describes basic configuration requirements for the LDAP directory that must be met before you install the SPE component.

## SPE Attributes

The MBean configuration file for SPE is located in:

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

This file applies to SESM applications that incorporate the SPE APIs, which are:

- Any SESM portal deployed in LDAP mode
- RDP
- CDAT

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

The config.xml file for SPE 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 6-8](#) explains the configurable attributes in this MBean. For a sample file, see the “[Sample SPE MBean Configuration File](#)” section on page F-18.

**Table 6-8** Attributes in the Dess-Auth MBean Configuration File

Object Name	Attribute Name	Explanation
Directory MBean	poolSize	Number of active connections allowed to the LDAP server.
	URL	URL of the LDAP server.
	principal	Name used when connecting to the LDAP server.
	credentials	Credentials (such as password) used for connecting to the LDAP server.
	context	Default LDAP context. This is the organization and organizational unit that was created to hold the SESM data.
	DESSPrincipal	Name used to connect to the SESM organization and organization unit. This user must have permission to create objects in the SESM context.
	alwaysGetAllAttributes	If set to true, all the attributes of an LDAP entry are returned for every query.
	traceFileName	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 and writes them into the log file.
	stackTrace	If set to true, the application prints 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 6-8 Attributes in the Dess-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"> <li>• The startNWSP script uses 64 MB</li> <li>• The runrdp script uses 20 MB</li> </ul> <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><b>Note</b> 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 degrades 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 Loading Initial RBAC Objects

An SESM deployment running in LDAP 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 SPE 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 LDAP mode.



### Note

If the SESM components are distributed among different servers, which means that SPE 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 SPE installation, use either of the following procedures:

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

## Using an SESM Custom Installation to Update the Schema and Load RBAC Objects

To use the SESM custom installation process to extend the directory schema and load 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 **SPE**.
  - 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
```

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.
- 

## Loading Sample Data and Logging into CDAT for the First Time

Before any administrator can log into CDAT to create objects, some initial RBAC rules and roles must be loaded into the directory. Load these top level objects by loading the sample RBAC data files that are installed with SPE. You can also use your own data generating tool.

See the *Cisco Distributed Administration Tool Guide* for information about the initial RBAC objects, loading sample data, and logging into CDAT.

The sample data is located in the following directory:

```
dess-auth
  schema
```



### Note

The sample data uses common name (cn) as a component of distinguished name (dn). If your LDAP directory uses unique identifier (uid) rather than common name to allow access to the directory, you must edit the sample data files before loading them. Edit the DESSusecasedata.ldf and DESSadmin.ldf files, replacing all occurrences of cn with uid.

---

## Configuring Specific Features

This section describes how to configure the following features:

- [Automatic Connections, page 6-42](#)
- [Configuration-based Location and Brand Awareness, page 6-44](#)

## Automatic Connections

An automatically connected service is one that is connected immediately after the subscriber authenticates, without requiring the subscriber to explicitly select the service. This section describes two topics related to automatic connections:

- [Configuring Automatic Connections, page 6-42](#)
- [Subscriber Experiences with Automatic Connections, page 6-43](#)

## Configuring Automatic Connections

In general, if a service is marked as an auto connect service, the SSG performs the automatic connection after the subscriber authenticates. There is a special case with SESM in LDAP mode in which SESM is involved with automatic connection.

### Configuring a Service for Automatic Connection

A subscriber profile specifies services for automatic connection. The subscriber profile also controls whether or not the service is hidden or not. If an auto connect service is hidden, it does not appear in the service list displayed on a service connection page.

In RADIUS mode, to configure a service for automatic connection, use the Account-Info A attribute in the subscriber profile. See [Table D-5 on page D-9](#) for more information.

In LDAP mode, to configure a service for automatic connection:

- Subscribers can use the web portal's self-management features to select and deselect the auto connect feature for a service.
- Administrators can use CDAT to maintain subscriber profiles. See the *Cisco Distributed Administration Tool Guide* for information.

### Configuring SESM to Request Automatic Connections in LDAP Mode

In LDAP 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:

- 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 auto connection in the subscriber's profile.

The service information consumes memory on the SSG host.

- 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.

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:

```
<Set name="autoConnect" type="boolean">false</Set>
```

Change the value to `true` to enable automatic connections by the SESM web application.

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.

## Subscriber Experiences with Automatic Connections

This section describes the behavior of the SESM portal application regarding automatically connected services.

### Connection Status for Auto Connect Services

The status page in an SESM portal shows the status for all services, including automatically connected services. In NWSP, the selection page includes service status indicators for each service listed. Hidden services are not listed. See the [“Configuring a Service for Automatic Connection” section on page 6-42](#) for an explanation of a hidden service.

Immediately after logging in, the service status for auto connect services might display as not connected. This happens if the service indicators display before the connection is completed. Proxy and tunnel services, for example, can take a while to connect. If the subscriber refreshes the window or selects the status window, the automatically connected services display with a connected status.

### Pop-Up Window for Auto Connect Services

If the subscriber has a home URL set to an auto connect service, the pop-up window for the service might appear before the connection completes. If this occurs, the following message appears in the pop-up window:

```
Page cannot be displayed.
```

The URL is the correct one. If the subscriber waits a short time and resubmits the request using the URL already displayed in the window, the service pages appear.

### Changing the Auto Connect Property for a Service

In LDAP mode, a subscriber can use the SESM self-management features to select or deselect the auto connect property. These changes are recorded immediately in the LDAP directory, but the change is not effective immediately. Changes are not visible in SESM until the cache timeout period in RDP elapses.

For example, a subscriber might select the auto connect property for a service, log out of SESM, log back in, and notice that the service was not automatically connected. Caching in the RDP causes this delay.

Caching in RDP improves system performance. The deployer can turn off caching or reduce the cache period, but those actions impact performance.

### Disconnecting Auto Connect Services

A subscriber can disconnect an auto connected service at any time. The disconnected status persists as long as the subscriber remains authenticated. The SESM single sign-on option affects whether a subscriber remains authenticated across SESM sessions. If the subscriber has to reauthenticate after an SESM session expires, the SSG reconnects all auto connect services.

An SESM session might expire, for example, because the subscriber closed the browser or navigated away from the SESM pages. When an SESM session expires:

- With single sign-on, subscribers are not required to reauthenticate.
- Without single sign-on, subscribers are required to reauthenticate when they navigate back to the SESM portal application. As a result of the reauthentication, SSG reconnects the auto connect services.

We recommend running SESM portal applications with single sign-on turned on.

## Configuration-based Location and Brand Awareness

You can use various ways to determine a subscriber's location and brand. This section describes how to implement the configuration-based methods. See the “[Location Awareness](#)” section on page 3-10 and “[Brand Awareness](#)” section on page 3-10 sections for a summary of other ways to determine location and brand.

You can implement location and brand awareness by adding the following elements in the SESM portal application's configuration file.

- In the SSG MBean, an SSG subnet entry can have the following attributes:
  - SESSION\_LOCATION
  - SESSION\_BRAND

The subnet entry associates an SSG IP address or client subnet address with a specific location or brand value. See the “[Global and Subnet Attribute Elements](#)” section on page 6-28 for information about subnet entries.

- In the WebApp MBean, the addDimension call defines the SESSION\_LOCATION or SESSION\_BRAND values.
- In the WebApp MBean, the addDimension call can create and assign arbitrary properties to the location or brand values. The SESM portal can use these properties.

For the session or brand determination to be meaningful, a web developer must change the SESM portal application to use the values. For new arbitrary properties to be meaningful, the portal must be changed to take an action with them.

The nwsp.xml file includes a configuration example that:

- Uses the SESSION\_LOCATION attribute in an SSG subnet entry to associate a location to an SSG IP address.
- Uses the addDimension call to associate a different URL to specific locations.



### Note

The example is only a configuration example; the NWSP application does not use the derived location or the associated URL.

## Configuring a Customized SESM Application

The Cisco SESM is a collection of components for creating specialized Java 2 Platform, Enterprise Edition (J2EE) web server applications. J2EE provides a framework for using various Java-based components to develop multi-tiered applications. The multi-tiered application (as opposed to the 2-tiered client server application) provides many opportunities for isolating and controlling functional pieces of a large application. For more information about the J2EE development platform, see:

<http://java.sun.com/j2ee/>

## SESM Application Definition

A Cisco SESM application consists of the following:

- SESM servlets and classes—The SESM API defines the SESM classes, including the configurable MBeans, used to implement the application functionality.

- **ConfigAgent**—The ConfigAgent is a Cisco developed MBean that configures other MBeans. It configures MBeans that are registered with the JMX server by applying parameter values from .xml files. Because .xml files are easily maintained and changed by system administrators, applications that use ConfigAgent are highly configurable without recompiling. Chapter 4 in this guide explains all of the configurable parameters in all of the MBeans.
- **Java Server Pages (JSPs)**—JSPs offer a way to deliver dynamic content in web pages. Web developers at the deployment site can control their subscriber's SESM experience through the JSPs. The *Cisco Subscriber Edge Services Manager Web Developer Guide* provides instructions for defining and compiling JSPs.
- **Images**—Images are used by the JSPs and control the look and feel and branding aspects of an SESM application. The *Cisco Subscriber Edge Services Manager Web Developer Guide* provides instructions for changing images and incorporating them into the JSPs.

## SESM Application Names

The SESM application name that you use for a customized application is arbitrary, but it must match in all of the following locations:

- The name of the application-specific subdirectory under the installation directory. For example, the directory that holds all application specific information for the NWSP application is:

```
<installDir>nwsp
```

- Application parameter inside the application startup script. In the installed scripts, the application name is hardcoded on the line that calls the generic start script. For example, for the NWSP application on Windows NT, the call line is:

```
call "%SCRIPTDIR%start.cmd" nwsp %PORTNO%
```

- Name of the application's configuration file in the `jetty` subdirectory. For example, for the NWSP application, the configuration filename is:

```
nwsp.jetty.xml
```

An application name in the startup script tells the ConfigAgent which configuration file to open. The application name is passed to ConfigAgent by the application startup scripts. The application name might also be used in other ways. For example, you can configure the parameter that defines the Jetty Server log filename to incorporate the application name in the log filename.

## Creating Configuration Files and Startup Scripts

Application developers at your site might make changes to the delivered NWSP sample application, producing a customized application. Customized applications require their own set of configuration files, although the files might be very similar to those provided for the sample application.

To create the required configuration files and startup scripts for a customized SESM application that will run in a Jetty server, follow these steps:

- 
- Step 1** Create a configuration file for the new application in the container's config directory. You can copy the `nwsp.jetty.xml` file and appropriately rename it. For example:

```
jetty
  config
    newApplication.jetty.xml
```

**Step 2** Edit the new file, enabling and disabling features as described in the [“Configuring an SESM Portal Application” section on page 6-14](#).

**Step 3** Create a startup script for the new application by copying the `startNWSP` script and appropriately renaming the copy. For example:

```
jetty
  bin
    startNewApplication
```

**Step 4** Edit the new file, changing the application name and the port number parameters. See the [“Startup Script Explanation” section on page 7-3](#) for more information.

**Step 5** Copy the `nwsp` directory structure, and rename the `nwsp` objects appropriately. For example, copy:

```
nwsp
  config
    nwsp.xml
  docroot
  docs
```

**Step 6** See the *Cisco Subscriber Edge Services Manager Web Developer Guide* for information about customizing the JSPs, images, and other components. That guide also describes how to update the `docroot` folder, recompile affected components, and edit the `web.xml` file.

---