



Configuring a Jetty Container for SESM Applications

The SESM installation process performs all required configurations for running the SESM applications in Jetty containers. Use this chapter if you want to change or fine-tune the J2EE container configuration after installation.

This chapter contains the following topics:

- [J2EE Containers, page 4-1](#)
- [Jetty Container MBeans, page 4-2](#)

J2EE Containers

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 a NWSP application. This section includes the following topics:

- [One-to-One Relationship Between Containers and SESM Applications, page 4-1](#)
- [Using Containers Other Than Jetty, page 4-2](#)
- [Container Requirement for the Port-Bundle Host Key Feature, page 4-2](#)

One-to-One Relationship Between Containers and SESM Applications

SESM applications are designed and configured with the assumption that each application runs in its own container. If you are running two instances of the same application, or two different SESM portals, you are running two web servers.

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

To change the configuration of an SESM application, you might change attributes in:

- Application MBeans, which configure processes for the SESM application
- Container MBeans, which configure processes for the J2EE server

Using Containers Other Than Jetty

The installed configuration is specific to a Jetty container. If you choose to deploy a container other than Jetty, you must make changes to the container MBeans, such as changing class or object names, or adding more MBeans.

Container Requirement for the Port-Bundle Host Key Feature

Before deploying SESM applications in containers other than Jetty, determine if your solution requires the port-bundle host key feature on the Service Selection Gateway (SSG). For solutions that use SSG, we recommend enabling the port-bundle host key feature.



Note

The Jetty server is currently the only J2EE-compliant server that can support the port-bundle host key feature.

The port-bundle host key feature uses a software token (or key) that *uniquely* identifies each subscriber on the host SSG that is currently logged on to an SESM portal, even when multiple subscribers are using the same IP address. The port-bundle host key feature also provides an SSG IP address in the key.

The port-bundle host key feature provides the following advantages to SESM portal applications:

- It allows SESM portal applications to robustly handle overlapping IP addresses, nonroutable IP addresses, and dynamically assigned IP addresses.
- It eliminates the need to explicitly map subscriber subnets to SSGs.

When port-bundle host key is enabled on the SSG, the SSG preserves the port number of the incoming HTTP request. This remote port number becomes the key that uniquely identifies each subscriber. The key is included in the request that is forwarded to the SESM web application.

The SSG makes the port number available, but the J2EE server must access this information and pass it along to the SESM web application. To do this, the Jetty server uses the PortBundleHandler, an extension that allows access to the request handling part of the server API and thus get the remote port number.

The PortBundleHandler is added to the Jetty container in the Jetty MBean, as described in the [“SESMSocketListener MBean” section on page 4-7](#).

Jetty Container MBeans

A Jetty container uses the following MBeans:

- [Log MBean](#)
- [Debug MBean](#)
- [SESMSocketListener MBean](#)

To change attributes in these MBeans, you can use either of the following methods:

- Edit the container’s MBean configuration file:

```
jetty
  config
    nwsp.jetty.xml
    wap.jetty.xml
    pda.jetty.xml
    cdat.jetty.xml
```

- Make changes using the Agent View running on the application's management port. For example, the Agent View for the NWSP application provides access to both the application and the container MBeans.



Note Containers do not have their own management consoles.

Log MBean

The Log MBean enables the Jetty server debugging and logging mechanisms and configures the information that appears in the jetty log file. [Table 4-1](#) describes the attributes in the Log MBean.

Table 4-1 Jetty Container—Log MBean

Attribute Name	Explanation
logTimezone	Installed default: empty
logDateFormat	Controls the format of the date stamp in the log messages. Installed default: yyyyMMdd:HHmmss.SSS
logLabels	Controls whether or not the log messages include frame details. Installed default: false
logOneLine	Installed default: false
logStackSize	Controls whether or not the log messages include an indication of stack depth. Installed default: false
logStackTrace	Controls whether or not the log messages include trace information. Installed default: false
logTags	Installed default: true
logTimeStamps	Installed default: true
append	Indicates if messages overwrite existing contents (false) or are appended to the existing file (true). Installed default: true
retainDays	Indicates the number of days to keep an old log file before deleting it. Installed default: 31
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"> • <i>application.log</i>—A Java system property. The same system property is used for all log files, so that they are all created in the same directory. Table 9-1 on page 9-5 describes how the start script sets <i>application.log</i>. • <i>yyyy_mm_dd</i>—The year, month, and day that the file was created. • <i>.jetty.log</i>—A constant identifying the Jetty log files.

Debug MBean

The Debug MBean enables or disables the Jetty server debugging mechanism. [Table 4-2](#) describes the attributes in the DebugMBean.

Table 4-2 Jetty Container—Debug MBean

Attribute Name	Explanation
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
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

Server MBean

The Server MBean configures a request log, which records all incoming HTTP requests. [Table 4-3](#) describes the attributes in the Server MBean.

Table 4-3 Jetty Container—Server MBean

Attribute Name	Explanation
RequestLog	<p>Creates a new class with one argument, which specifies the name and location of the request log. The installed value is:</p> <pre>application.log/yyyy_mm_dd.request.log</pre> <p>Where:</p> <ul style="list-style-type: none"> <i>application.log</i>—Is a Java system property whose value is set in the generic startup script. The same system property is used for all log files, so that they are all created in the same directory. See Table 9-1 on page 9-5 for a description of how the start script sets <i>application.log</i>. <i>yyyy_mm_dd</i>—Is the year, month, and day that the file was created. The installation program uses the appropriate path name delimiter for the installation platform. <i>.request.log</i>—Is a constant identifying an HTTP request file.
retainDays	<p>Indicates the number of days to keep a log file before deleting it.</p> <p>Installed default: 90</p>
append	<p>Indicates whether or not to append messages to an existing file or to create a new file for each application instance.</p> <p>Installed default: true</p>

Table 4-3 Jetty Container—Server MBean (continued)

Attribute Name	Explanation
<Call addWebApplication>	<p>This call adds the SESM application to run on the web server. It uses five positional arguments:</p> <ol style="list-style-type: none"> 1. The first positional argument specifies the virtual host name for the web server application. 2. The second positional argument specifies the context path for locating the web server application. For example, / or /pathname/*. 3. The third positional argument identifies the location of the application. The value is: <i>application.home/docroot</i> Where: <i>application.home</i> is a Java system property. 4. The fourth positional argument identifies the location of the webdefault.xml file for this application. The value is: <i>jetty.home/config/webdefault.xml</i> Where: <i>jetty.home</i> is a Java system property 5. The fifth positional argument specifies whether or not web archive (WAR) files are used. Valid values are TRUE and FALSE. Set this value to FALSE, since SESM applications are not WAR files. <p>The first three arguments define the location of the web server application. <i>host/context/application</i></p> <p>The SESM application startup scripts derive the values for <i>application.home</i> and <i>jetty.home</i> from an expected (installed) directory structure. To change the value of <i>application.home</i> or <i>jetty.home</i>, edit the start script.</p>
<Call addHandler>	<p>This call adds the PortBundleHandler to the container. This handler is required for SESM solutions that use the port bundle host key feature on the SSG.</p>

SESMSocketListener MBean

The SESMSocketListener MBean configures the port that the Jetty server listens on for HTTP requests from subscribers. [Table 4-4](#) describes the attributes in the SESMSocketListener MBean.


Table 4-4 Jetty Container—SESMSocketListener MBean

Attribute Name	Explanation
port	<p>Sets the port number that the web server listens on. The installed value is a Java system property named:</p> <pre>application.portno</pre> <p>Note The startup script sets this system property. Unless you alter the startup script, the default value in the MBean configuration file is ignored during application startup.</p> <p>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 application port that you provided during the installation process.</p>
minThreads	<p>Sets the minimum number of threads that this listener maintains during periods of low load. This listener always has 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 the length of time 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 the length of time that the web server waits for a request 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>

SESMSSLListener MBean

The SESMSSLListener MBean configures the port that the Jetty server listens on for requests from subscribers on the Secure Sockets Layer (SSL). [Table 4-5](#) describes the attributes in the SESMSSLListener MBean.

Table 4-5 Jetty Container—SESMSSLListener MBean

Attribute Name	Explanation
port	<p>Sets the port that the secure socket layer (SSL) listener uses. The installed value is a Java system property named:</p> <pre>application.ssl.portno</pre> <p>Note The startup script sets this system property. Unless you alter the startup script, the default value in the MBean configuration file is ignored during application startup.</p> <p>The generic startup script derives a value for <code>application.ssl.portno</code> based on the value of <code>application.portno</code>, as follows:</p> <pre>application.ssl.portno = application.portno - 80 + 443</pre> <p>To change the value of <code>application.ssl.portno</code>, edit the generic startup script.</p>
MinThreads	<p>Sets the minimum number of threads that this listener maintains during periods of low load. The listener always has 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. Sample keystore files are included in the installation for each portal application. For example:</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> Caution 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 “Using HTTPS in SESM Portals” section on page A-1 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.