

JES Administration

This chapter describes how to use the framework console and the JES Management Panel to

- manage the bundle life cycle in the Java Embedded Server framework
- obtain information about installed bundles
- get and set system properties and OSGI environment variables

▼ Launching the Framework

1. Set the *JES_JAVA_HOME* environment variable to the pathname of the Java installation you wish to use.

On Solaris, `setenv JES_JAVA_HOME path_to_java_install_dir`. For example,

```
% setenv JES_JAVA_HOME /usr/bin/jdk1.2.2-006
```

On Windows NT, set `JES_JAVA_HOME=path_to_java_install_dir`. For example,

```
% set JES_JAVA_HOME=c:\pjava302
```

If you are using the Java 2 platform, we recommend version 1.2.2-006 or greater.

2. Use the `runjes` command to start the framework.

```
% cd jes_install_dir  
% bin/runjes
```

The JES welcome and framework prompt appear.

```
Java Embedded Server 2.0
Copyright 1998, 1999 and 2000 Sun Microsystems, Inc. All rights
reserved.
Use is subject to license terms.
Type 'h[elp]' for a list of commands.
>
```

Syntax for runjes and runjes.bat

```
runjes[.bat] [-cacheDir pathname] [-dryrun] [-j jvm_option]. . . [ commands ]
```

TABLE 1-1 Options to the runjes Command

Option	Description
<code>-cacheDir <i>pathname</i></code>	The pathname of the directory JES should use for its cache. The default is <code>/home/<i>user_name</i>/jescache</code> on Solaris and Linux and <code>%HOME%\jescache</code> on Windows NT. The value of variable <i>HOME</i> must not contain any quotes. The value of <i>pathname</i> is assigned to the property <code>com.sun.jes.framework.cache.dir</code> .
<code>-dryrun</code>	Show what would be done without actually doing it. You may want to use the <code>-dryrun</code> option to examine what the runjes utility does when it is called.
<code>-j <i>jvm_option</i></code>	Pass the <i>jvm_option</i> to the Java interpreter. By default, the following three system properties are defined. <code>com.sun.jes.framework.bundles.baseurl</code> is set to <code>file:<i>install_dir</i>/bundles</code> , where <i>install_dir</i> is the path to the JES installation directory. This allows you to use short names (instead of fully specifying the pathname) when referring to system bundles, for example, <code>install log servlet http</code> . <code>com.sun.jes.impl.keystore.store</code> is set to <code>file:<i>install_dir</i>/lib/tlscerts</code> and <code>com.sun.jes.impl.keystore.access</code> is set to <i>password</i> where <i>install_dir</i> is the path to the JES installation directory and <i>password</i> is the the encoded password needed to access the keystore. The keystore is an example keystore that is used in the SSL examples. You can override these properties to point to your own information.
<i>commands</i>	This can be the pathname or URL of a file from which to read JES commands, or a list of JES commands. If a command contains a blank, it must be quoted when used on the <code>runjes</code> command line. For example, <code>runjes "install log"</code> . If no commands are specified, then commands are read from stdin.

The Framework Cache

The framework maintains a cache in a directory on the local file system. This directory stores the actual bundle files that have been installed as well as the status of each bundle. When the framework instance is stopped, this cache directory is not removed. You can then start the framework again, and it restores itself with all bundles in the same state as they were just prior to shutdown, using the information in this cache directory.

The `runjes` command launches a framework instance as a new process. You should include the `cacheDir` option to explicitly specify which instance (that is, which cache) should be started. In this example, the framework that is dormant in `~/myCacheDir` is launched.

```
% runjes -cacheDir ~/myCacheDir
```

If there is no cache directory and you do not specify one, `runjes` creates one named `path_to_home/jescache` where `path_to_home` is the value of the Java system property `user.home`. You can change the default cache directory location by setting the `com.sun.jes.framework.cache.dir` system property.

You may find it useful to remove a cache directory, especially when developing bundles. You can remove the framework cache just as you would any directory in your file system.

Managing the Bundle Life Cycle

Once you have the framework running, you can begin using it to install, start, update, stop, and uninstall bundles, managing the bundle life cycle.

Bundle States

During its life cycle, a bundle may be in one of the following states.

TABLE 1-2 Bundle States

State	Description
INSTALLED	The bundle has been successfully installed.
RESOLVED	All Java classes and native code that this bundle requires have been made available.
STARTING	The bundle is being started

TABLE 1-2 Bundle States

State	Description
STOPPING	The bundles is being stopped
ACTIVE	The bundle has successfully started and is running.
UNINSTALLED	The bundle has been uninstalled.

Framework Commands

You can use the following framework commands

- on the `runjes` command line (see “Shortcuts to Starting Up” on page 15)
- in a file
- at the framework prompt.

TABLE 1-3 Framework Commands

Command Name	Description
<code>b[undles]</code>	List all installed bundles and their status.
<code>e[xportedpackages]</code>	List exported packages and the bundles that export and import them respectively.
<code>g[et] <i>property_name</i></code>	Get the value of the specified property.
<code>h[elp] [<i>command</i>]</code>	List all commands and their options.
<code>i[nstall]</code> <code><i>bundle_url</i> [, ...]</code>	Install the specified bundles.
<code>m[anifest]</code> <code><i>bundle_url</i> <i>bundle_id</i></code>	Display values of bundle manifest headers.
<code>r[un] <i>filename</i> <i>url</i></code>	Execute commands read from the specified filename or URL.
<code>se[t]</code> <code><i>property_name</i>=<i>property_value</i></code>	Set the value for the specified property.
<code>ser[vices] [<i>filter</i>]</code>	List all registered services whose properties match the given <i>filter</i> or all services if no filter is specified. The filter is the string representation of an LDAP search filter as defined in <i>RFC 1960: A String Representation of LDAP Search Filters</i> .
<code>sh[utdown]</code>	Shutdown the framework and exit.
<code>sta[rt]</code> <code><i>bundle_url</i> <i>bundle_id</i> [, ...]</code>	Install and start the specified bundles.

TABLE 1-3 Framework Commands

Command Name	Description
sto[p] <i>bundle_url</i> <i>bundle_id</i> [, ...]	Stop the specified bundles.
tty	Read commands from stdin. Use tty when you are starting the framework and you want to continue to work interactively.
un[install] <i>bundle_url</i> <i>bundle_id</i> [, ...]	Uninstall the specified bundles.
up[date] <i>bundle_url</i> <i>bundle_id</i> [<i>bundle_update_url</i>] [, ...]	Update the specified bundle. <i>bundle_url</i> takes precedence over the bundle's Bundle-UpdateLocation manifest header. The variables <i>bundle_url</i> and <i>bundle_update_url</i> must specify the full URL of a bundle or a path relative to that specified by the system property com.sun.jes.framework.bundles.baseurl.

▼ Installing and Starting the Core Service Bundles

1. From the framework prompt, install the log, servlet, and http service bundles.

```
> install log.jar, servlet.jar, http.jar
```

Since by default, `runjes` sets the system property `com.sun.jes.framework.bundles.baseurl` to `file:install_dir/bundles`, where `install_dir` is the path to the JES installation directory, you can use short names (instead of fully specifying the pathname) when referring to system bundles. You can also leave off the `.jar` suffix, if you prefer.

2. Get the IDs for the bundles you installed.

```
> bundles
ID  STATE      LOCATION
--  -
1   INSTALLED  file:/home/mcm/jes2.0/bundles/log.jar
2   INSTALLED  file:/home/mcm/jes2.0/bundles/servlet.jar
3   INSTALLED  file:/home/mcm/jes2.0/bundles/http.jar
```

Note that the state of the bundles is `INSTALLED`.

3. View the manifest for the HTTP bundle (optional).

```
> manifest 3
Bundle-Vendor: Sun Microsystems, Inc.
Bundle-Version: 0.1
Bundle-Activator: com.sun.jes.impl.http.HttpActivator
Bundle-DocURL: http://java.sun.com/products/embeddedserver
Created-By: 1.2.2 (Sun Microsystems Inc.)
Bundle-Name: http
Manifest-Version: 1.0
Bundle-ContactAddress: jes-comments@sun.com
Export-Package: org.osgi.service.http; specification-version=1.0,
org.osgi.service.log; specification-version=1.0,
com.sun.jes.service.http, com.sun.jes.service.ssl
Export-Service: org.osgi.service.http.HttpService
Bundle-Description: The HTTP Service
Import-Package: javax.servlet; specification-version=2.1.1,
javax.servlet.http; specification-version=2.1.1
```

Notice that the HTTP bundle imports the `javax.servlet` and `javax.servlet.http` packages. Though the HTTP bundle depends on the servlet bundle, you do not have to install or start the servlet bundle before the HTTP bundle. You can start bundles in groups (as demonstrated in the next step) and the framework will resolve dependencies automatically.

4. Start the service bundles.

```
> start 1,2,3
> bundles
ID  STATE      LOCATION
--  -
1   ACTIVE    file:/home/mcm/jes2.0/bundles/log.jar
2   ACTIVE    file:/home/mcm/jes2.0/bundles/servlet.jar
3   ACTIVE    file:/home/mcm/jes2.0/bundles/http.jar
```

Notice that the state of the bundles is now ACTIVE.

Examining Package Dependencies

- To obtain a list of package dependencies, use the `exportedpackages` command.

```
> exportedpackages
Package: javax.servlet.http (2.1.1)
  Exported by: 2 (file:/home/mcm/jes2.0/bundles/servlet.jar)
  Imported by: 3 (file:/home/mcm/jes2.0/bundles/http.jar)
Package: com.sun.jes.service.http (0.0.0)
  Exported by: 3 (file:/home/mcm/jes2.0/bundles/http.jar)
Package: org.osgi.service.log (1.0.0)
  Exported by: 1 (file:/home/mcm/jes2.0/bundles/log.jar)
  Imported by: 3 (file:/home/mcm/jes2.0/bundles/http.jar)
Package: com.sun.jes.service.ssl (0.0.0)
  Exported by: 3 (file:/home/mcm/jes2.0/bundles/http.jar)
Package: javax.servlet.jsp (2.1.1)
  Exported by: 2 (file:/home/mcm/jes2.0/bundles/servlet.jar)
Package: javax.servlet (2.1.1)
  Exported by: 2 (file:/home/mcm/jes2.0/bundles/servlet.jar)
  Imported by: 3 (file:/home/mcm/jes2.0/bundles/http.jar)
Package: org.osgi.service.http (1.0.0)
  Exported by: 3 (file:/home/mcm/jes2.0/bundles/http.jar)
```

Getting a List of Services

- To obtain a list of all registered services, use the `services` command.

```
> services
[com.sun.jes.service.http.auth.users.UserPasswordService]
  description=User and Password Management Service
[com.sun.jes.service.http.auth.basic.BasicSchemeHandler]
  description=The OSGI HTTP Basic Authentication Service
[org.osgi.service.http.HttpService]
  description=The OSGI HTTP Service
[com.sun.jes.service.http.HttpAdmin]
[org.osgi.service.log.LogService]
  description=The standard OSGi Log service
[org.osgi.service.log.LogReaderService]
  description=The standard OSGi LogReader service
```

- To obtain a list of services that match specific properties, use the `services` command with the `filter` option. The example returns the services that do not have the string “OSGI” in their descriptions.

```
> services (!(description=*OSGI*))
[com.sun.jes.service.http.auth.users.UserPasswordService]
    description=User and Password Management Service
[com.sun.jes.service.http.HttpAdmin]
```

Updating a Bundle

The `update` command replaces an existing bundle with a new one. The command requires the URL or ID of the bundle to update. You can also specify a URL for the bundle you want to use for the update. If you do not specify a URL, the framework uses the URL specified in the `Bundle-UpdateLocation` manifest header. If no `Bundle-UpdateLocation` is specified, the framework uses the location of the original bundle.

Stopping Bundles

The `stop` command stops a running bundle. The command requires the URL or ID of the bundle to stop. When you stop a bundle, the framework unregisters the bundle's services, releases any services used by the bundle, and sets the bundle's state to `RESOLVED`.

Uninstalling Bundles

The `uninstall` command stops a bundle (if it is running), releases any persistent resources the bundle was holding, and sets the bundle's status to `UNINSTALLED`. Exported packages remain, however, until the framework is shutdown.

Setting and Getting System Properties

You can use the `get` and `set` commands to view and modify system properties on the framework command line. Most system properties changes take effect immediately, but `com.sun.jes.framework.cachedir` does not take effect until the framework is restarted. For a complete list of JES system properties and how to use them, see Appendix A.

You can also use the `get` command to view the following OSGi framework environment properties.

TABLE 1-4 OSGi Framework Environment Properties

Property	Description
<code>org.osgi.framework.version</code>	The version of the framework
<code>org.osgi.framework.vendor</code>	The vendor of this framework implementation.
<code>org.osgi.framework.language</code>	The language being used. See ISO 639 for possible values.
<code>org.osgi.framework.os.name</code>	The name of the operating system of the hosting computer.
<code>org.osgi.framework.os.version</code>	The version number of the operating system of the hosting computer.
<code>org.osgi.framework.processor</code>	The name of the processor of the hosting computer.

Using the JES Management Panel

The JES Management Panel (JESMP) provides a graphical interface to a framework instance. Once you have a framework and core services running (see “Launching the Framework” on page 5 and “Installing and Starting the Core Service Bundles” on page 9), you can install and start the bundles required for the JESMP.

▼ Launching the JESMP

1. Install and start the JESMP bundle and the bundles it depends upon.

```
> start httpauth, tcatjspcruntime, httpusers, jesmp
> bundles
ID  STATE      LOCATION
-----
1   ACTIVE    file:/home/mcm/jes2.0/bundles/log.jar
2   ACTIVE    file:/home/mcm/jes2.0/bundles/servlet.jar
3   ACTIVE    file:/home/mcm/jes2.0/bundles/http.jar
4   ACTIVE    file:/home/mcm/jes2.0/bundles/httpauth.jar
5   ACTIVE    file:/home/mcm/jes2.0/bundles/tcatjspcruntime.jar
6   ACTIVE    file:/home/mcm/jes2.0/bundles/httpusers.jar
7   ACTIVE    file:/home/mcm/jes2.0/bundles/jesmp.jar
```

Notice that you do not have to type the `.jar` extension and that the `start` command was used to both install and start the bundles.

2. Open the JES Management Panel in your browser.

Open the JES Management Panel in your browser.

In the address window of your browser, type `http://host:port/admin`, where *host* is the host name specified in the `com.sun.jes.service.http.hostname` property (the default is the host where JES is running, or any host if that host is multi-homed), and *port* is the port number specified in the `com.sun.jes.service.http.port` property (default is 8080). For example, if neither property has been set, the JESMP can be accessed at this URL:
`http://localhost:8080/admin`.

You are prompted for a user name and password. By default, they are both `admin`.

Managing and Monitoring the Bundle Life Cycle

You can install, start, update, stop, and uninstall bundles, view the list of services a bundle has registered or is using, view package import and export information, read the JES log, and manage user accounts from the JESMP.

Bundles

The left pane of the *Bundles* tab lists all installed bundles. Click on a bundle name to view information about its state (INSTALLED, ACTIVE, RESOLVED), location, dependencies, and manifest headers and to perform operations such as starting, stopping, and updating the bundle.

You can also download and install a new bundle by typing the URL of the bundle in the first text box after the bundle list, then clicking *Install*.

You can browse to and install a bundle in a local file system by clicking the *Browse* button next to the second text box after the bundle list, then clicking *Install*.

Services

The left pane of the *Services* tab lists all registered services. Click on a service name to view information about its source bundle, client bundles, registration properties, and configuration properties. You can modify the configuration properties of a service as well. For example, you can change the `logSize` and `severityThreshold` properties for the `LogService`.

Packages

The left pane of the *Packages* tab lists all the packages currently in use. Click on a package name to view information about its exporter and importers.

View Log

The *View Log* tab lists the log messages as specified by the log system properties. To modify the number and type of messages that appear in the log, change the configuration properties listed for the `LogService` on the *Services* tab.

User Management

You can add and remove users from the *User Management* tab.

Shortcuts to Starting Up

Along with these utilities there are three scripts in the `jes_install_dir/bin` directory, `starthttp`, `startjesmp`, and `startall`, that automatically install and start the bundles required for the HTTP service, the JES Management Panel, or all JES services, respectively.

Using `runjes` and the `start*` Scripts

You can use either of the three start scripts, `starthttp`, `startjesmp`, and `startall`, with the `runjes` utility or with the `framework run` command.

Launching the Framework and Starting the HTTP Service

1. Launch an interactive framework session with `runjes`.

```
% cd install_dir
% bin/runjes tty
Java Embedded Server 2.0

Copyright 1998, 1999 and 2000 Sun Microsystems, Inc. All rights
reserved.
Use is subject to license terms.

Type 'h[elp]' for a list of commands.

>
```

2. Install and start the bundles required for the HTTP service.

```
> run install_dir/bin/starthttp
> bundles
ID  STATE  LOCATION
--  -
1   ACTIVE file:/home/mcm/jes2.0/bundles/servlet.jar
2   ACTIVE file:/home/mcm/jes2.0/bundles/http.jar
```

Launching the Framework and Starting the JES Management Panel

1. Launch a noninteractive framework session and install and start the bundles required for the JESMP.

```
% cd install_dir
% bin/runjes bin/startjesmp
% com.sun.jes.impl.http.ResourceServlet: init
com.sun.jes.impl.http.auth.users.UserAdminServlet: init
com.sun.jes.impl.http.auth.users.AddUserServlet: init
com.sun.jes.impl.http.auth.users.RemoveUserServlet: init
com.sun.jes.impl.http.auth.users.ErrorServlet: init
com.sun.jes.impl.http.auth.users.ChangePasswordServlet: init
com.sun.jes.impl.http.ResourceServlet: init
com.sun.jes.impl.jesmp.ui.MainPanel: init
com.sun.jes.impl.jesmp.ui.LRPanels: init
com.sun.jes.impl.jesmp.ui.BundleList: init
com.sun.jes.impl.jesmp.ui.BundleDetail: init
com.sun.jes.impl.jesmp.ui.ServiceList: init
com.sun.jes.impl.jesmp.ui.ServiceDetail: init
com.sun.jes.impl.jesmp.ui.PackageList: init
com.sun.jes.impl.jesmp.ui.PackageDetail: init
com.sun.jes.impl.jesmp.ui.LogDetail: init
com.sun.jes.impl.jesmp.ui.TabBar: init
```

2. Open the JES Management Panel in your browser.

In the address window of your browser, type `http://host:port/admin`, where *host* is the host name specified in the `com.sun.jes.service.http.hostname` property (the default is the host where JES is running, or any host if that host is multi-homed), and *port* is the port number specified in the `com.sun.jes.service.http.port` property (default is 8080). For example, if neither property has been set, the JESMP can be accessed at this URL:
`http://localhost:8080/admin`.

You are prompted for a user name and password. By default, they are both `admin`.

