Sun GlassFish Enterprise Server v3 Prelude Administration Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 820–4495–13
June 2009

090615@22510

Contents

Figures

Tables

Examples

Preface

The *Sun GlassFish Enterprise Server v3 Prelude Administration Guide* provides instructions for configuring and administering the Sun GlassFish Enterprise Server.

This preface contains information about and conventions for the entire Sun GlassFish™ Enterprise Server documentation set.

The following topics are addressed here:

Enterprise Server Documentation Set

The Enterprise Server documentation set describes deployment planning and system installation. The Uniform Resource Locator (URL) for Enterprise Server documentation is `http://docs.sun.com/coll/1343.7`. For an introduction to Enterprise Server, refer to the books in the order in which they are listed in the following table.

TABLE P–1 Books in the Enterprise Server Documentation Set

Book Title	Description
Release Notes	Provides late-breaking information about the software and the documentation. Includes a comprehensive, table-based summary of the supported hardware, operating system, Java™ Development Kit (JDK™), and database drivers.
Quick Start Guide	Explains how to get started with the Enterprise Server product.

TABLE P–1 Books in the Enterprise Server Documentation Set *(Continued)*

Book Title	Description
Installation Guide	Explains how to install the software and its components.
Application Deployment Guide	Explains how to assemble and deploy applications to the Enterprise Server and provides information about deployment descriptors.
Developer's Guide	Explains how to create and implement Java Platform, Enterprise Edition (Java EE platform) applications that are intended to run on the Enterprise Server. These applications follow the open Java standards model for Java EE components and APIs. This guide provides information about developer tools, security, and debugging.
Add-On Component Development Guide	Explains how to use published interfaces of Enterprise Server to develop add-on components for Enterprise Server. This document explains how to perform *only* those tasks that ensure that the add-on component is suitable for Enterprise Server.
RESTful Web Services Developer's Guide	Explains how to develop Representational State Transfer (RESTful) web services for Enterprise Server.
Getting Started With JRuby on Rails for Sun GlassFish Enterprise Server	Explains how to develop Ruby on Rails applications for deployment to Enterprise Server.
Getting Started With Project jMaki for Sun GlassFish Enterprise Server	Explains how to use the jMaki framework to develop Ajax-enabled web applications that are centered on JavaScript™ technology for deployment to Enterprise Server.
Roadmap to the Java EE 5 Tutorial	Explains which information in the *Java EE 5 Tutorial* is relevant to users of the v3 Prelude release of the Enterprise Server.
Java EE 5 Tutorial	Explains how to use Java EE 5 platform technologies and APIs to develop Java EE applications.
Java WSIT Tutorial	Explains how to develop web applications by using the Web Service Interoperability Technologies (WSIT). The tutorial focuses on developing web service endpoints and clients that can interoperate with Windows Communication Foundation (WCF) endpoints and clients.
Administration Guide	Explains how to configure, monitor, and manage Enterprise Server subsystems and components from the command line by using the `asadmin(1M)` utility. Instructions for performing these tasks from the Administration Console are provided in the Administration Console online help.
Administration Reference	Describes the format of the Enterprise Server configuration file, `domain.xml`.
Troubleshooting Guide	Describes common problems that you might encounter when using Enterprise Server and how to solve them.

TABLE P–1 Books in the Enterprise Server Documentation Set *(Continued)*

Book Title	Description
Reference Manual	Provides reference information in man page format for Enterprise Server administration commands, utility commands, and related concepts.

Related Documentation

A Javadoc™ tool reference for packages that are provided with the Enterprise Server is located at `https://glassfish.dev.java.net/nonav/api/v3-prelude/index.html`. Additionally, the following resources might be useful:

- The Java EE 5 Specifications (`http://java.sun.com/javaee/5/javatech.html`)
- The Java EE Blueprints (`http://java.sun.com/reference/blueprints/index.html`)

For information about creating enterprise applications in the NetBeans™ Integrated Development Environment (IDE), see `http://www.netbeans.org/kb/60/index.html`.

For information about the Java DB for use with the Enterprise Server, see `http://developers.sun.com/javadb/`.

Typographic Conventions

The following table describes the typographic changes that are used in this book.

TABLE P–2 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	A placeholder to be replaced with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized (note that some emphasized items appear bold online)	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file.

Symbol Conventions

The following table explains symbols that might be used in this book.

TABLE P–3 Symbol Conventions

Symbol	Description	Example	Meaning
`[ ]`	Contains optional arguments and command options.	`ls [-l]`	The `-l` option is not required.
`{ | }`	Contains a set of choices for a required command option.	`-d {y|n}`	The `-d` option requires that you use either the `y` argument or the `n` argument.
`${ }`	Indicates a variable reference.	`${com.sun.javaRoot}`	References the value of the `com.sun.javaRoot` variable.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
→	Indicates menu item selection in a graphical user interface.	File → New → Templates	From the File menu, choose New. From the New submenu, choose Templates.

Default Paths and File Names

The following table describes the default paths and file names that are used in this book.

TABLE P–4 Default Paths and File Names

Placeholder	Description	Default Value
as-install	Represents the base installation directory for Enterprise Server. In configuration files, *as-install* is represented as follows: `${com.sun.aas.installRoot}`	Installations on the Solaris™ operating system, Linux operating system, and Mac operating system: *user's-home-directory*`/glassfishv3-prelude/glassfish` Windows, all installations: *SystemDrive*`:\glassfishv3-prelude\glassfish`
domain-root-dir	Represents the directory in which a domain is created by default.	*as-install*`/domains/`

TABLE P–4 Default Paths and File Names *(Continued)*

Placeholder	Description	Default Value
domain-dir	Represents the directory in which a domain's configuration is stored. In configuration files, *domain-dir* is represented as follows: `${com.sun.aas.instanceRoot}`	*domain-root-dir*/*domain-name*

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Searching Sun Product Documentation

Besides searching Sun product documentation from the docs.sun.com[SM] web site, you can use a search engine by typing the following syntax in the search field:

search-term `site:docs.sun.com`

For example, to search for "broker," type the following:

```
broker site:docs.sun.com
```

To include other Sun web sites in your search (for example, java.sun.com, www.sun.com, and developers.sun.com), use `sun.com` in place of `docs.sun.com` in the search field.

Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Send Comments. In the online form, provide the full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the part number of this book is 820-4495.

◆ ◆ ◆ CHAPTER 1

Overview of Sun GlassFish Enterprise Server Administration

Sun GlassFish™ Enterprise Server v3 Prelude provides an environment for the development and deployment of Java applications and web services.

As an Enterprise Server administrator, your main responsibilities are to establish a secure Enterprise Server environment and to oversee the services, resources, and users that participate in that environment. Information and instructions on performing the associated tasks from the command line are provided in this document. Information and instructions for accomplishing the tasks by using the Administration Console are contained in the Administration Console online help.

The following topics are addressed here:

About Your Administrative Tasks

The command-line instructions in this guide are organized around the following administrative tasks:

- Administering domains, resources, and system properties

  See Chapter 2, "Basic Administration."
- Installing and updating add-on components by using the `pkg` command from the Update Center

  See Chapter 3, "Extending Enterprise Server."
- Administering Java Virtual Machine (JVM) options and profilers

  See Chapter 4, "Administering the Virtual Machine for the Java Platform."
- Starting and stopping Java DB and administering JDBC resources, connection pools, and drivers

See Chapter 5, “Administering Database Connectivity .”

- Administering passwords, audit modules, and JSSE certificates

  See Chapter 6, “Administering System Security.”
- Administering authentication realms and users

  See Chapter 7, “Administering User Security.”
- Administering HTTP listeners and virtual servers

  See Chapter 8, “Administering the HTTP Service.”
- Configuring logging and viewing log files

  See Chapter 9, “Administering Logging.”
- Configuring monitoring and viewing monitoring data

  See Chapter 10, “Monitoring the Enterprise Server.”

Although not documented in this guide, the following tasks are often performed by administrators:

- Assembling and deploying applications

  See *Sun GlassFish Enterprise Server v3 Prelude Application Deployment Guide*.
- Diagnosing and resolving problems
- See *Sun GlassFish Enterprise Server v3 Prelude Troubleshooting Guide*

About Your Administrative Tools

For the most part, you can perform the same tasks by using either the graphical Administration Console or the `asadmin` command-line utility, however, there are exceptions. Step-by-step instructions for using the Administration Console are provided in the Administration Console online help. Procedures for using the command-line utilities are provided in this guide and in the command-line help pages, which are similar to man pages.

Instructions for manually editing configuration files are provided when there is no way to accomplish the tasks by using the Administration Console or the `asadmin` utility.

Enterprise Server provides the following administration tools:

- “Administration Console” on page 25
- “Command-Line Utility for Administration (`asadmin`)” on page 25
- “Update Tool” on page 26
- “Command-Line Utility for Security (`keytool`)” on page 27
- “Java Monitoring and Management Console (JConsole)” on page 27

Note – Instructions written for the Enterprise Server tools use standard UNIX® forward slashes (/) for directory path separators in commands and file names. If you are running Enterprise Server on a Microsoft Windows system, use backslashes (\) instead. For example:

- UNIX: *as-install*`/bin/asadmin`
- Windows: *as-install*`\bin\asadmin`

Administration Console

The Administration Console is a browser-based utility that features an easy-to-navigate graphical interface that includes extensive online help for the administrative tasks.

You can display the help material for a page in the Administration Console by clicking the Help button on the page. The initial help page describes the functions and fields of the page itself. Associated task instructions can be accessed on additional pages by clicking a link in the See Also list.

To Start the Administration Console

To use the Administration Console, the domain administration server (DAS) must be running. Each domain has its own DAS, which has a unique port number. When Enterprise Server was installed, you chose a port number for the DAS, or used the default port of 4848. You also specified a user name and password if you did not choose anonymous login.

When specifying the URL for the Administration Console, use the port number for the domain to be administered. The format for starting the Administration Console in a web browser is `http://`*hostname*`:`*port*. For example:

```
http://kindness.sun.com:4848
```

If the Administration Console is running on the host where the Enterprise Server was installed, specify `localhost` for the host name. For example:

```
http://localhost:4848
```

For Microsoft Windows, an alternative way to start the Enterprise Server Administration Console is by using the Start menu.

Command-Line Utility for Administration (`asadmin`)

The `asadmin` utility is a command-line tool that invokes commands for identifying the operation or task that you want to perform. You can run `asadmin` commands either from a command prompt or from a script. Running `asadmin` commands from a script is helpful for automating repetitive tasks.

Note – Not all `asadmin` command options are supported for this release of Enterprise Server. If you specify an unsupported option, a syntax error does not occur. Instead, the command runs successfully and the unsupported option is silently ignored.

The following topics are addressed here:

Help for the `asadmin` Utility

The `asadmin`(1M) help page explains the basics of how the `asadmin` command works. To display the help page for the `asadmin` utility, type `asadmin` without a subcommand.

You can display a help page for an `asadmin` command by typing the command name followed by the `--help` option. For example:

```
asadmin create-jdbc-resource --help
```

Note – To display the help page for a remote command, Enterprise Server must be running.

Use the `asadmin list-commands` command to display the available commands. A collection of the `asadmin` help pages is available in HTML and PDF format in the *Sun GlassFish Enterprise Server v3 Prelude Reference Manual*.

To Start the `asadmin` Utility

To issue an `asadmin` command, go to the default *as-install*`/bin` directory and type the `asadmin` command and a subcommand. For example:

```
asadmin create-jdbc-resource
```

A list of the `asadmin` commands included in this release is contained in Appendix A, "The `asadmin` Utility Commands."

Update Tool

Enterprise Server provides a set of image packaging system (IPS) tools for updating software on a deployed Enterprise Server. Typical updates include new releases of Enterprise Server, and new or revised Enterprise Server add-on components.

- Update Tool is a graphical utility that you can either invoke in the Administration Console, or from the command line by using the `updatetool` command. A desktop Update Tool Notifier lets you know when updates are available and provides a direct means of applying the updates to your Enterprise Server.

  Instructions for using the graphical versions of the Update Tool are contained in the Administration Console online help and the standalone Update Tool online help.
- The `pkg` command is the command-line version of Update Tool.

  Instructions for using the `pkg` command are contained in Chapter 3, "Extending Enterprise Server."

Command-Line Utility for Security (`keytool`)

The `keytool` utility is used to set up and work with Java Security Socket Extension (JSSE) digital certificates. See "Administering JSSE Certificates " on page 96 for instructions on using `keytool`.

Java Monitoring and Management Console (JConsole)

Java SE provides tools to connect to an MBean Server and view the MBeans registered with the server. JConsole is one such popular JMX Connector Client and is available as part of the standard Java SE distribution. See "Connecting JConsole to Enterprise Server" on page 151 for instructions on implementing JConsole in the Enterprise Server environment.

About the Configuration Files

The following configuration files are associated with Enterprise Server administration:

- `asadminenv.conf`
- `domain.xml`
- `logging.properties`
- `resources.xml`
- `server.policy`

The bulk of the configuration information on Enterprise Server is stored in the `domain.xml` file. This file is the central repository for a given administrative domain and contains an XML representation of the Enterprise Server domain model. For details on the `domain.xml` file, see *Sun GlassFish Enterprise Server v3 Prelude Administration Reference*.

If you want to make configuration changes, you can use either the Administration Console or the `asadmin` utility. Either method is preferable to editing the configuration files directly, because direct editing is very prone to error and can have unintended results.

Configuration changes often require that you restart Enterprise Server for the changes to take effect. In other cases, changes are applied dynamically without restarting Enterprise Server.

Configuration Changes That Require Server Restart

When making any of the following configuration changes, you must restart the server for the changes to take effect:

- Creating or deleting a resource or entity
- Adding or deleting a component
- Changing JVM options
- Changing port numbers
- Changing log handler elements
- Configuring certificates
- Managing HTTP services
- Managing thread pools
- Modifying the following JDBC connection pool properties:
  - `datasource-classname`
  - `associate-with-thread`
  - `lazy-connection-association`
  - `lazy-connection-enlistment`
  - JDBC driver vendor-specific properties
- Modifying the following connector connection pool properties:
  - `resource-adapter-name`
  - `connection-definition-name`
  - `transaction-support`
  - `associate-with-thread`
  - `lazy-connection-association`
  - `lazy-connection-enlistment`
  - Vendor-specific properties

Dynamic Configuration Changes

With *dynamic configuration*, changes take effect while the server is running. To make the following configuration changes, you do not need to restart the server:

- Adding or removing JDBC, JMS, and connector resources and pools
- Adding file realm users
- Changing logging levels
- Changing monitoring levels

- Enabling and disabling resources and applications
- Deploying, undeploying, and redeploying applications

◆ ◆ ◆ CHAPTER 2

Basic Administration

This chapter provides procedures for performing basic administration tasks in the Sun GlassFish™ Enterprise Server v3 Prelude environment by using the `asadmin` command-line utility.

The following topics are addressed here:

Instructions for accomplishing these tasks by using the Administration Console are contained in the Administration Console online help.

Administering Domains

A domain provides a preconfigured runtime for user applications. This runtime includes a basic security structure where specific groupings of server instances (domains) can be administered by different administrators. Enterprise Server installer creates a default administrative domain named `domain1`, as well as an associated domain administration server (DAS) named `server`. The default administration port is 4848, but a different port can be specified during installation. The administration user name and password are also established if anonymous login was not chosen during installation.

A domain has its own configuration, log files, and application deployment areas that are independent of other domains. If the configuration is changed for a domain, the configurations for other domains are not affected.

The *domain administration server* (DAS) is a specially-designated instance that hosts the administrative applications. The DAS authenticates the administrator, accepts requests from administration tools, and communicates with server instances in the domain to carry out requests. The DAS is sometimes referred to as the *default server* because it is the only server instance created during Enterprise Server installation that can be used for deployment.

Each domain has its own DAS with a unique port number. The graphical Administration Console communicates with a specific DAS to administer the domain associated with the DAS. Each Administration Console session enables you to configure and manage the specific domain. If you create multiple domains, you must start a separate Administration Console session to manage each domain.

The following instructions are used to administer domains:

▼ To Create a Domain

After installing Enterprise Server and creating the default domain (`domain1`), you can create additional domains by using the local `create-domain` command. This command creates the configuration of a domain.

Any user who has access to the `asadmin` utility on a given system can create a domain and store the domain configuration in a folder of choice. By default, the domain configuration is created in the default directory for domains. You can override this location to store the configuration elsewhere.

1 Select a name for the domain that you are creating.

You can verify that a name is not already in use by listing the existing domains:

```
asadmin list-domains
```

2 Create a domain by using the `create-domain`(1) command.

Example 2–1 Creating a Domain

The following example command creates a domain named `domain1` . When you type the command, you might be prompted for login information.

```
asadmin create-domain --adminport 4848 domain1
```

Information similar to the following is displayed:

```
Enter admin user name[Enter to accept default]>
Using port 4848 for Admin.
Default port 8080 for HTTP Instance is in use. Using 1161
Using default port 7676 for JMS.
Using default port 3700 for IIOP.
Using default port 8181 for HTTP_SSL.
Using default port 3820 for IIOP_SSL.
Using default port 3920 for IIOP_MUTUALAUTH.
Default port 8686 for JMX_ADMIN is in use. Using 1162
Distinguished Name of the self-signed X.509 Server Certificate is:
[CN=moonbeam.gateway.2wire.net,OU=GlassFish,O=Sun Microsystems,L=Santa Clara,ST
California,C=US]
Domain domain1 created.
Command create-domain executed successfully.
```

To start the Administration Console in a browser, enter the URL in the following format:

`http://`*hostname*`:5000`

For this example, the domain's log files, configuration files, and deployed applications now reside in the following directory:

domain-root-dir`/mydomain`

See Also To see the full syntax and options of the command, type `asadmin create-domain --help` at the command line.

▼ To List Domains

The `list-domains` command provides a list of domains and their statuses. If the domain directory is not specified, the contents of the default *as-install*`/domains` directory is listed. If there is more than one domain, the domain name must be specified.

To list domains that were created in other directories, specify the `--domaindir` option.

- **List domains by using the `list-domains`(1) command.**

Example 2–2 Listing Domains

The following example command lists the domains in the default *as-install*`/domains` directory:

```
asadmin list-domains
```

Information similar to the following is displayed:

```
Name: domain1 Status: Running
Name: domain4 Status: Not Running
Name: domain6 Status: Not Running
Command list-domains executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin list-domain --help` at the command line.

▼ To Delete a Domain

Only the root user or the operating system user who is authorized to administer the domain can run this command.

Before You Begin A domain must be stopped before it can be deleted.

1. **Obtain the exact name of the domain that you are deleting.**

   To list the existing domains:

   ```
   asadmin list-domains
   ```

2. **If necessary, notify domain users that the domain is being deleted.**

3. **Ensure that the domain you want to delete is stopped.**

   For instructions, see "To Stop a Domain (or Server)" on page 35.

4. **Delete the domain by using the `delete-domain`(1) command.**

Example 2–3 Deleting a Domain

The following example command deletes a domain named `domain1` from the location specified:

```
asadmin delete-domain --domaindir ..\domains domain1
```

Information similar to the following is displayed:

```
Domain domain1 deleted.
Command delete-domain executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin delete-domain --help` at the command line.

▼ To Start a Domain (or Server)

When you start a domain, the domain administration server (DAS) is started. After startup, the DAS runs constantly, listening for and accepting requests.

If the domain directory is not specified, the domain in the default *as-install*`/domains` directory is started. If there are two or more domains, the `domain_name` operand must be specified. Each domain must be started separately.

Note – For Microsoft Windows, you can use an alternate method to start a domain. From the Windows Start menu, select Programs → Sun Microsystems →Enterprise Server → Start Admin Server.

This command is supported in local mode only.

- **Start a domain by using the `start-domain`(1) command.**

Example 2–4 Starting a Domain

The following example command starts `domain1`:

```
asadmin start-domain domain1
```

Information similar to the following is displayed:

```
Name of the domain started: [domain1] and its location:
[C:\prelude\v3_prelude_release\distributions\web\target\glassfish
domains\domain1].
Admin port for the domain: [4848].
```

See Also To see the full syntax and options of the command, type `asadmin start-domain --help` at the command line.

▼ To Stop a Domain (or Server)

Stopping a domain shuts down its domain administration server (DAS). When stopping a domain, the DAS stops accepting new connections and then waits for all outstanding connections to complete. This shutdown process takes a few seconds. While the domain is stopped, the Administration Console and most of the `asadmin` commands cannot be used.

Note – For Microsoft Windows, you can use an alternate method to stop a domain. From the Start menu, select Programs → Sun Microsystems → Enterprise Server → Stop Admin Server.

1. **If necessary, notify users that you are going to stop the domain.**
2. **Stop the domain by using the `stop-domain(1)` command.**

Example 2–5 Stopping a Domain (or Server)

The following example command stops `domain1`:

```
asadmin stop-domain domain1
```

Information similar to the following is displayed:

```
Waiting for the domain to stop ..........
Command stop-domain executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin stop-domain --help` at the command line.

▼ To Log In to a Domain (or Server)

The `login` command enables you to authenticate yourself (log in to) a specific domain, after which you need not specify the administration password through the `--user` option or the `---passwordfile` option for subsequent operations on the domain.

The `login` command can only be used to specify the administration password. For other passwords that remote commands require, use the `--passwordfile` option, or specify the passwords at the command prompt. You are always prompted for the admin user name and password.

There is no logout command. If you want to login to another domain, invoke `asadmin login` with new values for `--host` and `--port`.

1. **Determine the name of the domain that you are logging in to.**

   To list the existing domains:

   ```
   asadmin list-domains
   ```

2. **Log in to the domain by using the `login(1)`command.**

Example 2–6 Logging In To a Domain Located on Another Machine

The following example command logs you in to the running domain on port 4848 of host `moonbeam`:

```
asadmin login --host moonbeam --port 4848
```

Information similar to the following is displayed:

```
Enter admin user name[Enter to accept default]>admin
Enter admin password>adminadmin
GlassFish v3 Prelude (build Jennifer-private)
Login information relevant to admin user name [admin] for host [localhost] and
dmin port [4848] stored at [C:\Documents and Settings\Jennifer\.asadminpass] su
cessfully.
Make sure that this file remains protected. Information stored in this file will
be used by asadmin commands to manage associated domain.
Command login executed successfully.
```

Example 2–7 Logging In to a Domain on the Default Port of Localhost

The following example command logs you in to the default domain of `localhost`:

```
asadmin login
```

Information similar to the following is displayed:

```
Enter admin user name[Enter to accept default]>
GlassFish v3 Prelude (build Jennifer-private)
Admin login information for host [localhost] and port [4848] is being
overwrittn with credentials provided. This is because the --savelogin option
was used during create-domain command.
Login information relevant to admin user name [anonymous] for host [localhost]
nd admin port [4848] stored at [C:\Documents and Settings\Jennifer\.asadminpass
successfully.
Make sure that this file remains protected. Information stored in this file will
be used by asadmin commands to manage associated domain.
Command login executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin login --help` at the command line. For additional information on passwords, see "Administering Passwords" on page 92.

▼ To Display Domain Uptime

The `uptime` command displays the length of time that the domain administration server (DAS) has been running since it was last started.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Display uptime by using the `uptime(1)` command.**

Example 2–8 Displaying the DAS Uptime

The following example command displays the length of time that the DAS has been running:

```
asadmin uptime
```

Information similar to the following is displayed:

```
Uptime: 31 minutes, 54 secons
Command uptime executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin uptime --help` at the command line.

▼ To Switch a Domain to Another Supported Java Version

Enterprise Server v3 Prelude requires Version 5 Java SE platform as the underlying virtual machine for the Java™ platform (Java Virtual Machine or JVM™ machine).

Note – Do not downgrade to an earlier Java version after a domain has been created with a newer JVM machine. If you must downgrade your JVM machine, downgrade it only for individual domains.

---

1. **If you have not already done so, download the desired Java SDK (not the JRE) and install it on your system.**

   The Java SDK can be downloaded from `http://java.sun.com/j2se`.

2. **Start the domain for which you are changing the JDK.**

   as-install`/bin/asadmin start-domain` *domain-name*

For a valid JVM installation, the order in which locations are checked is as follows:

a. `domain.xml` (`java-home` **inside** `java-config`)

b. `asenv.conf` (**setting** `AS_JAVA="path to java home"`)

c. `JAVA_HOME` **environmental variable**

d. `system.getProperty ("java.home") + "../"`

e. `system.getProperty ("java.home")`

3 **If necessary, change the JVM machine attributes for the domain.**

In particular, you might need to change the `JAVA_HOME` environment variable. For example, to change the `JAVA_HOME` variable, type:

as-install`/bin/asadmin set "server.java-config.java-home=path-to-java-home"`

Administering System Properties

Shared server instances will often need to override attributes defined in their referenced configuration. Any configuration attribute can be overridden through a system property of the corresponding name.

The following topics are addressed here:

▼ To Create System Properties

The remote `create-system-properties` command enables you to add or update one or more system properties of the domain or configuration.

1 **Ensure that the server is running.**

Remote commands require a running server.

2 **Create system properties by using the `create-system-properties`(1) command.**

Example 2–9 Creating a System Property

The following example command creates a system property associated with `http-listener-port=1088` on `localhost`:

```
asadmin create-system-properties http-listener-port=1088
```

Information similar to the following is displayed:

```
Command create-system-properties executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin create-system-properties --help` at the command line.

▼ To List System Properties

The remote `list-system-properties` command enables you to list the system properties that apply to a domain or configuration.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **List system properties by using the `list-system-properties(1)` command.**

   The existing system properties are displayed, including predefined properties such as `HTTP_LISTENER_PORT` and `HTTP_SSL_LISTENER_PORT`.

Example 2–10 Listing System Properties

The following example command lists the system properties on host `localhost`:

```
asadmin list-system-properties
```

Information similar to the following is displayed:

```
http-listener-port=1088
Command list-system-properties executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin list-system-properties --help` at the command line.

▼ To Delete a System Property

The remote `delete-system-property` command enables you to delete system properties. Any configuration attribute can be overwritten through a system property of the corresponding name.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Obtain the exact name of the system property that you are deleting.**

   To list the existing system properties:

   ```
   asadmin list-system-properties
   ```

3. **Delete the system property by using the `delete-system-property`(1) command.**

4. **If necessary, notify users that the system property has been deleted.**

Example 2–11 Deleting a System Property

The following example command deletes a system property named `http-listener-port` from `localhost`:

```
asadmin delete-system-property http-listener-port
```

Information similar to the following is displayed:

```
Command delete-system-property executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin delete-system-property --help` at the command line.

Additional Administrative Tasks

The following topics are addressed here:

Listing Various System Elements

Various list commands can be useful in administering your system.

▼ To List Applications

The remote `list-applications` command enables you to list the deployed Java applications. If the `--type` option is not specified, all applications are listed.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **List applications by using the `list-applications`(1) command.**

Example 2–12 Listing Applications

The following example command lists the web applications on `localhost`:

```
asadmin list-applications --type web
```

Information similar to the following is displayed:

```
hellojsp <web>
Command list-applications executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin list-applications --help` at the command line.

▼ To List Commands

The remote `list-commands` command enables you to list the deployed `asadmin` commands. You can specify that only remote commands or only local commands are listed. By default, this command displays a list of local commands followed by a list of remote commands.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **List commands by using the `list-commands`(1) command.**

Example 2–13 Listing Commands

The following example command lists only local commands:

```
asadmin list-commands --localonly
```

Information similar to the following is displayed:

```
create-domain
delete-domain
list-commands
list-domains
login
monitor
start-database
start-domain
stop-domain
stop-database
version
Command list-commands executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin list-commands --help` at the command line.

▼ To List Containers

The remote `list-containers` command enables you to list application containers.

1. **Ensure that the server is running.**
   Remote commands require a running server.
2. **List containers by using the `list-containers(1)` command.**

Example 2–14 Listing Containers

The following example command lists the containers on `localhost`:

```
asadmin list-containers
```

Information similar to the following is displayed:

```
List all known application containers
Container : connectors
    properties=(ContractProvider=connectors)
Container : jpa
    properties=(ContractProvider=jpa)
```

```
Container : web
    properties=(ContractProvider=web)
Container : jruby
    properties=(ContractProvider=jruby)
Container : security
    properties=(ContractProvider=security)
Command list-containers executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin list-containers --help` at the command line.

▼ To List Modules

The remote `list-modules` command enables you to list the modules that are accessible to the Enterprise Server module subsystem. The status of each module is included. Possible statuses include NEW and READY.

1 Ensure that the server is running.

Remote commands require a running server.

2 List modules by using the `list-modules(1)` command.

Example 2–15 Listing Modules

The following example command lists the accessible modules:

```
asadmin list-modules
```

Information similar to the following is displayed (partial output):

```
List Of Modules
Module : org.glassfish.web.jstl-connector:10.0.0.b28
    properties=(visibility=public,State=READY,Sticky=true)
    Module Characteristics : List of Jars implementing the module
        Jar : file:/C:/prelude/v3_prelude_release/distributions/web/target/glass
fish/modules/web/jstl-connector.jar
    Module Characteristics : List of imported modules
    Module Characteristics : Provides to following services
Module : org.glassfish.admingui.console-common:10.0.0.b28
    properties=(visibility=public,State=NEW,Sticky=true)
Module : org.glassfish.admin.launcher:10.0.0.b28
    properties=(visibility=public,State=NEW,Sticky=true)
Module : org.glassfish.external.commons-codec-repackaged:10.0.0.b28
    properties=(visibility=public,State=NEW,Sticky=true)
Module : com.sun.enterprise.tiger-types-osgi:0.3.32.prelude-b28
    properties=(visibility=public,State=READY,Sticky=true)
```

```
    Module Characteristics : List of imported modules
    Module Characteristics : Provides to following services
    Module Characteristics : List of Jars implementing the module
        Jar : file:/C:/prelude/v3_prelude_release/distributions/web/target/glass
fish/modules/tiger-types-osgi.jar.
.
.
.
Command list-modules executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin list-modules --help` at the command line.

▼ To Add Resources

The remote `add-resources` command enables you to create the resources named in the specified XML file.

The XML file must reside in the *as-install*`/domains/domain1/config` directory. If you specify a relative path or simply provide the name of the XML file, this command will prepend *as-install*`/domains/domain1/config` to this operand.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Add resources from an XML file by using the `add-resources`(1) command.**

Example 2–16 Adding Resources

The following example command creates resources using the contents of the `resource.xml` file on `localhost`:

```
asadmin add-resources c:\tmp\resource.xml
```

Information similar to the following is displayed:

```
Command : JDBC resource jdbc1 created successfully.
Command : JDBC connection pool poolA created successfully.
Command add-resources executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin add-resources --help` at the command line.

▼ To Display the Enterprise Server Version

The remote `version` command enables you to display information on the Enterprise Server version for a particular server. If the command cannot communicate with the server by using the specified login (user/password) and target (host/port) information, then the local version is displayed along with a warning message.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Display the version by using the `version(1)` command.**

Example 2–17 Displaying Version Information

The following example command displays the version of Enterprise Server on `localhost`:

```
asadmin version
```

Information similar to the following is displayed:

```
Version = GlassFish v3 Prelude
Command version executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin version --help` at the command line.

◆ ◆ ◆ CHAPTER 3

Extending Enterprise Server

This chapter provides procedures for performing tasks associated with extending and updating the deployed Sun GlassFish™ Enterprise Server v3 Prelude by using the `pkg` command. A set of reference pages that contain details on using the `pkg` command is included with Enterprise Server.

The following topics are addressed here:

Instructions for accomplishing these tasks by using the Administration Console are contained in the Administration Console online help for Update Tool. The standalone Update Tool also contains online help.

For additional information on Update Tool, see `http://wikis.sun.com/display/IpsBestPractices/Screenshots`

About Add-on Components

Enterprise Server is designed to provide its functionality in a modular form so that you can choose to include the functionality that you need and leave out the functionality that is not needed. Over time, new add-on components will be developed and existing components will be modified.

You can install new components or updated components by using the graphical Update Tool, which can be accessed in the Enterprise Server Administration Console. If you are interested in exploring the standalone version, run the `updatetool` command in the *as-install*`/bin` directory. Extensive online help is provided for both graphical versions.

From the command line, you can use the pkg(5) command to extend Enterprise Server. The pkg command, located in the *as-install*/bin directory, can perform most of the tasks offered by the graphical Update Tool. However, the main reasons for using the pkg command are:

- As a basis for creating update scripts
- As a method for working with systems that do not have a monitor, graphic card, or keyboard (headless systems)

Adding Components

This section provides instructions for using the pkg command to install Enterprise Server add-on components on your deployed Enterprise Server.

▼ To Install an Add-on Component

The pkg command enables you to install an add-on component on your system. If multiple versions of a package are available, the latest one is applied unless you specify otherwise.

Note – If the pkg component, the updatetool component, or any other valid component that you try to invoke from the command line is not yet installed on your deployed Enterprise Server, you will receive a query asking if you want to install the component. Answer Y to install the component.

Before You Begin Enterprise Server v3 Prelude must be fully deployed before you can install additional components. If you need installation instructions, see Chapter 1, "Installing Sun GlassFish v3 Preview Enterprise Server," in *Sun GlassFish Enterprise Server v3 Prelude Installation Guide*.

1 **List your installed components:**

```
pkg list
```

Information similar to the following is displayed:

```
NAME (AUTHORITY)                        VERSION         STATE      UFIX
glassfishv3-common                      0-1             installed  ----
glassfishv3-ejb                         0-1             installed  u---
glassfishv3-nucleus                     0-1             installed  ----
glassfishv3-web                         0-1             installed  ----
grails                                  1.0-1.0         installed  ----
jersey                                  0.7-0.1         installed  u---
jmaki                                   1.8.0-1.0       installed  ----
jruby                                   1.1.1-1.0       installed  ----
metro                                   1.2-1           installed  u---
```

```
pkg                                     0.1.4-6.564     installed  u---
python2.4-minimal                       2.4.4-6.564     installed  u---
updatetool (glassfish.org)              2.0-6.564       installed  u---
wxpython2.8-minimal                     2.8.7.1-6.564   installed  u---
```

2 List all packages that are available:

```
pkg list -a
```

Information similar to the following is displayed from the repository:

```
NAME (AUTHORITY)                        VERSION         STATE      UFIX
glassfishv3-common                      0-1             known      u---
glassfishv3-common                      0-1             known      u---
glassfishv3-common                      0-1             installed  ----
glassfishv3-ejb                         0-1             known      u---
glassfishv3-ejb                         0-1             known      u---
glassfishv3-ejb                         0-1             known      u---
glassfishv3-ejb                         0-1             installed  u---
glassfishv3-ejb                         0-1             known      ----
glassfishv3-nucleus                     0-1             known      u---
glassfishv3-nucleus                     0-1             known      u---
glassfishv3-nucleus                     0-1             installed  ----
glassfishv3-web                         0-1             known      u---
glassfishv3-web                         0-1             known      u---
glassfishv3-web                         0-1             installed  ----
grails                                  1.0-1.0         installed  ----
javadb                                  0-1             known      u---
javadb                                  0-1             known      u---
javadb                                  0-1             known      ----
jersey                                  0.7-0.1         installed  u---
jersey                                  0.7-0.2         known      u---
jersey                                  0.7-0.3         known      u---
jersey                                  0.8-0.1         known      ----
jmaki                                   1.8.0-1.0       installed  ----
jruby                                   1.1.1-1.0       installed  ----
metro                                   1.2-1           installed  u---
metro                                   1.2-2           known      u---
metro                                   1.2-3           known      ----
pkg                                     0.1.4-6.564     installed  u---
pkg                                     0.1.5-8.724     known      ----
python2.4-minimal                       2.4.4-6.564     installed  u---
python2.4-minimal                       2.4.4-8.724     known      ----
updatetool                              2.0-6.564       installed  u---
updatetool                              2.0-8.724       known      ----
wxpython2.8-minimal                     2.8.7.1-6.564   installed  u---
wxpython2.8-minimal                     2.8.7.1-8.724   known      ----
```

3 Navigate to the *as-install* directory.

4 **Install a package from the available packages list.**

Use the following syntax: `pkg install` *package-name*. For example:

```
pkg install javadb
```

The most recent version of the component is installed and information similar to the following is displayed:

```
DOWNLOAD                                  PKGS       FILES     XFER (MB)
javadb                                    0/1      61/200     2.10/7.26

PHASE                                     ACTIONS
Install Phase                             222/222
```

5 **To apply your changes, restart Enterprise Server.**

a. **Stop Enterprise Server.**

For instructions, see "To Stop a Domain (or Server)" on page 35.

b. **Start Enterprise Server.**

For instructions, see "To Start a Domain (or Server)" on page 35.

See Also Full syntax and options for using the `pkg` command are described in the pages in the *as-install*`/pkg/man/` directory. These reference pages are not displayed by using the `man` command. Instead, use a command such as `more` or `cat`.

Updating Installed Components

This section provides the following instructions for updating Enterprise Server components after they have been installed:

▼ To Update an Installed Component

When you install an updated version of a component, only those files that have been modified are downloaded and installed. Files that have been removed in the updated package are removed during the update process.

1 **Obtain a list of only the installed packages that have available updates:**

```
pkg list -u
```

Information similar to the following is displayed:

```
NAME (AUTHORITY)                          VERSION         STATE      UFIX
glassfishv3-ejb                           0-1             installed  u---
jersey                                    0.7-0.1         installed  u---
metro                                     1.2-1           installed  u---
pkg                                       0.1.4-6.564     installed  u---
python2.4-minimal                         2.4.4-6.564     installed  u---
updatetool (glassfish.org)                2.0-6.564       installed  u---
wxpython2.8-minimal                       2.8.7.1-6.564   installed  u---
```

2 Install a new version of a package.

Use the following syntax: `pkg install` *package-name*. For example:

```
pkg install metro
```

Information similar to the following is displayed:

```
DOWNLOAD                                PKGS       FILES     XFER (MB)
Completed                               1/1         5/5     0.49/0.49

PHASE                                   ACTIONS
Removal Phase                              2/2
Update Phase                               7/7
Install Phase                              2/2
```

3 To apply your changes, restart Enterprise Server.

a. Stop Enterprise Server.

For instructions, see "To Stop a Domain (or Server)" on page 35.

b. Start Enterprise Server.

For instructions, see "To Start a Domain (or Server)" on page 35.

See Also Full syntax and options for using the `pkg` command are described in the pages in the *as-install* `/pkg/man/` directory. These reference pages are not displayed by using the `man` command. Instead, use a command such as `more` or `cat`.

▼ To Update All Installed Components in an Image

Enterprise Server enables you to maintain multiple installation images on a single system. When you update an installation image, all the components that are present in that image are updated to new versions, if new versions are available. When you install updated versions of components, only those files that have been modified are downloaded and installed. Files that have been removed in the updated package are removed during the update process.

1. **Install all packages for the image.**

   Use the following syntax: `pkg install` *image-name*. For example:

   ```
   pkg image-update
   ```

   Information similar to the following is displayed:

   ```
   DOWNLOAD                                  PKGS       FILES     XFER (MB)
   Completed                                 6/6     729/729   21.59/21.59

   PHASE                                     ACTIONS

   Removal Phase                             887/887
   Update Phase                              253/253
   Install Phase                             584/584
   ```

2. **To apply your changes, restart Enterprise Server.**

   a. **Stop Enterprise Server.**

      For instructions, see "To Stop a Domain (or Server)" on page 35.

   b. **Start Enterprise Server.**

      For instructions, see "To Start a Domain (or Server)" on page 35.

See Also Full syntax and options for using the `pkg` command are described in the pages in the *as-install* `/pkg/man/` directory. These reference pages are not displayed by using the `man` command. Instead, use a command such as `more` or `cat`.

Removing Installed Components

If you are discontinuing use of a component and want to remove it from your system, you can do this by using the `uninstall` command. If you need to revert to a prior version of a component, you will need to uninstall the current version and install the prior version by specifying the version number.

- "To Uninstall an Installed Component" on page 52
- "To Uninstall and Revert to an Older Version of a Component" on page 53

▼ To Uninstall an Installed Component

Before You Begin Verify that there are no dependencies on the installed component before removing it.

1. **Obtain a list of all your installed components:**

   ```
   pkg list
   ```

Information similar to the following is displayed:

```
NAME (AUTHORITY)                          VERSION         STATE      UFIX
glassfishv3-common                        0-1             installed  ----
glassfishv3-ejb                           0-1             installed  u---
glassfishv3-nucleus                       0-1             installed  ----
glassfishv3-web                           0-1             installed  ----
grails                                    1.0-1.0         installed  ----
jersey                                    0.7-0.1         installed  u---
jmaki                                     1.8.0-1.0       installed  ----
jruby                                     1.1.1-1.0       installed  ----
metro                                     1.2-1           installed  u---
pkg                                       0.1.4-6.564     installed  u---
python2.4-minimal                         2.4.4-6.564     installed  u---
updatetool (glassfish.org)                2.0-6.564       installed  u---
wxpython2.8-minimal                       2.8.7.1-6.564   installed  u---
```

2 **Uninstall the component that you want removed from your system.**

Use the following syntax: `pkg uninstall` *package-name*. For example:

```
pkg uninstall jruby
```

3 **To apply your changes, restart Enterprise Server.**

a. **Stop Enterprise Server.**

For instructions, see "To Stop a Domain (or Server)" on page 35.

b. **Start Enterprise Server.**

For instructions, see "To Start a Domain (or Server)" on page 35.

See Also Full syntax and options for using the `pkg` command are described in the pages in the *as-install* `/pkg/man/` directory. These reference pages are not displayed by using the `man` command. Instead, use a command such as `more` or `cat`.

▼ To Uninstall and Revert to an Older Version of a Component

If there is a malfunction in an installed component, you might want to revert to an older version of that component. The way to restore an older version of a component is to first uninstall the current version of the component, then install the specific older version that you want to reinstate.

Before You Begin Be sure to verify that the older version of the component is in the repository before you uninstall your current version.

1. **Verify that the older version of the component is still available:**

   ```
   pkg list -a
   ```

2. **Obtain a list of your installed components:**

   ```
   pkg list
   ```

3. **Uninstall the currently-installed component that you want to replace.**

   Use the following syntax: `pkg uninstall` *package-name*. For example:

   ```
   pkg uninstall jersey
   ```

4. **Install the older version of the component.**

   Use the following syntax: `pkg install` *package-name* –`version` *version-number*. For example:

   ```
   pkg install jersey -version 0.7-0.2
   ```

5. **Verify that the older version is installed:**

   ```
   pkg list
   ```

See Also Full syntax and options for using the `pkg` command are contained in the pages in the *as-install* `/pkg/man/` directory. These reference pages are not displayed by using the `man` command. Instead, use a command such as `more` or `cat`.

◆ ◆ ◆ CHAPTER 4

Administering the Virtual Machine for the Java Platform

This chapter provides procedures for administering the Virtual Machine for the Java™ platform (Java Virtual Machine) or JVM™ machine) in the Sun GlassFish™ Enterprise Server v3 Prelude environment by using the `asadmin` command-line utility.

The following topics are addressed here:

Instructions for accomplishing these tasks by using the Administration Console are contained in the Administration Console online help.

Administering JVM Options

The Java Virtual Machine is an interpretive computing engine responsible for running the byte codes in a compiled Java program. The virtual machine translates the Java byte codes into the native instructions of the host machine. Enterprise Server, being a Java process, requires a virtual machine to run and support the Java applications running on it. JVM settings are part of an Enterprise Server configuration.

The following instructions are used to administer the JVM options:

▼ To Create JVM Options

The remote `create-jvm-options` command enables you to create JVM options in the Java configuration or the profiler elements of the `domain.xml` file. If JVM options are created for a profiler, these options are used to record the settings that initiate the profiler.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Create JVM options by using the `create-jvm-options`(1) command.**

   To create more than one JVM option, use a colon (:) to separate the options. If the JVM option itself contains a colon (:), use the backslash (\) to offset the colon delimiter.

3. **To apply your changes, restart Enterprise Server.**

   a. **Stop Enterprise Server.**

      For instructions, see "To Stop a Domain (or Server)" on page 35.

   b. **Start Enterprise Server.**

      For instructions, see "To Start a Domain (or Server)" on page 35.

Example 4–1 Creating JVM Options

```
asadmin create-jvm-options -Dunixlocation=/root/
example:-Dvariable=\$HOME:-Dwindowslocation
=d\\:\\\sun\\\appserver:-Doption1=-value1
```

Information similar to the following is displayed:

```
created 4 option(s)
Command create-jvm-options executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin create-jvm-options --help` at the command line.

▼ To List JVM Options

The remote `list-jvm-options` command enables you to list the existing JVM options.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **List JVM options by using the `list-jvm-options`(1) command.**

Example 4–2 Listing JVM Options

The following example command lists all JVM options:

```
asadmin list-jvm-options
```

Information similar to the following is displayed:

```
-Djava.security.auth.login.config=${com.sun.aas.instanceRoot}/config/login.conf
-XX: LogVMOutput
-XX: UnlockDiagnosticVMOptions
-Dcom.sun.enterprise.config.config_environment_factory_class=com.sun.enterprise.
config.serverbeans.AppserverConfigEnvironmentFactory
-Djavax.net.ssl.keyStore=${com.sun.aas.instanceRoot}/config/keystore.jks
-XX:NewRatio=2
-Djava.security.policy=${com.sun.aas.instanceRoot}/config/server.policy
-Djdbc.drivers=org.apache.derby.jdbc.ClientDriver
-Djavax.net.ssl.trustStore=${com.sun.aas.instanceRoot}/config/cacerts.jks
-client
-Djava.ext.dirs=${com.sun.aas.javaRoot}/lib/ext${path.separator}${com.sun.aas.ja
vaRoot}/jre/lib/ext${path.separator}${com.sun.aas.instanceRoot}/lib/ext${path.se
parator}${com.sun.aas.derbyRoot}/lib
-Xmx512m
-XX:LogFile=${com.sun.aas.instanceRoot}/logs/jvm.log
-Djava.endorsed.dirs=${com.sun.aas.installRoot}/lib/endorsed

Command list-jvm-options executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin list-jvm-options --help` at the command line.

▼ To Delete JVM Options

The remote `delete-jvm-options` command enables you to delete JVM options from the Java configuration or profiler elements of the `domain.xml` file.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Obtain the exact name of the JVM option that you are deleting.**

   To list the existing JVM options:

   ```
   asadmin list-jvm-options
   ```

3. **If necessary, notify users that the JVM option is being deleted.**

4. **Delete JVM options by using the `delete-jvm-options`(1) command.**

   To remove more than one JVM option, use a colon (:) to separate the options. If the JVM option itself contains a colon, use the backslash (\) to offset the colon delimiter.

5. **To apply your changes, restart Enterprise Server.**

   a. **Stop Enterprise Server.**

      For instructions, see "To Stop a Domain (or Server)" on page 35.

   b. **Start Enterprise Server.**

      For instructions, see "To Start a Domain (or Server)" on page 35.

Example 4–3 Deleting a JVM Option

The following example command removes a single JVM option:

```
asadmin delete-jvm-options -Dopt1=A
```

Information similar to the following is displayed:

```
deleted 1 option(s)
Command delete-jvm-options executed successfully.
```

Example 4–4 Deleting Multiple JVM Options

The following example command removes multiple JVM options:

```
asadmin delete-jvm-options -Doption1=-value1:-Dvariable=\$HOME
```

Information similar to the following is displayed:

```
deleted 2 option(s)
Command delete-jvm-options executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin delete-jvm-options --help` at the command line.

▼ To Generate a JVM Report

The remote `generate-jvm-report` command enables you to generate a JVM report showing the threads (dump of a stack trace), classes, memory, and loggers for a specified domain administration server (DAS). You can generate the following types of reports: summary (default), class, thread, log.

1. **Ensure that the server is running.**
   Remote commands require a running server.
2. **Generate the report by using the `generate-jvm-report(1)` command.**

Example 4–5 Generating a JVM Report

The following example command displays summary information about the threads, classes, and memory:

```
generate-jvm-report --type summary
```

Information similar to the following is displayed (partial output):

```
Operating System Information:
Name of the Operating System: Windows XP
Binary Architecture name of the Operating System: x86, Version: 5.1
Number of processors available on the Operating System: 2
System load on the available processors for the last minute: NOT_AVAILABLE.
(Sum of running and queued runnable entities per minute).
.
,
.
user.home = C:\Documents and Settings\Jennifer
user.language = en
user.name = Jennifer
user.timezone = America/New_York
user.variant =
variable = \$HOME
web.home = C:\prelude\v3_prelude_release\distributions\web\target\glassfish\modu
les\web
Command generate-jvm-report executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin generate-jvm-report --help` at the command line or see `generate-jvm-report(1)` in the reference manual.

Administering Profilers

Profilers generate information used to analyze server performance. If JVM options are created for a profiler, the options are used to record the settings needed to activate a particular profiler. You can use the `create-jvm-options` command to create JVM options in the Java configuration or profiler elements of the `domain.xml` file.

A server instance is associated with a particular profile by the profiler element in the Java configuration.

The following instructions are used to administer profiles:

- "To Create a Profiler" on page 60
- "To Delete a Profiler" on page 60

To Create a Profiler

The remote `create-profiler` command enables you to create a profiler element in the Java configuration.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Create a profiler by using the `create-profiler`(1) command.**

3. **To apply your changes, restart Enterprise Server.**

   a. **Stop Enterprise Server.**

      For instructions, see "To Stop a Domain (or Server)" on page 35.

   b. **Start Enterprise Server.**

      For instructions, see "To Start a Domain (or Server)" on page 35.

Example 4–6 Creating a Profiler

The following example command creates a profiler named `sample_profiler`:

```
asadmin create-profiler --classpath=/home/appserver/ --nativelibrarypath=/u/home/lib --enabled=false
--property=defaultuser=admin:password=adminadmin sample_profiler
```

Information similar to the following is displayed:

```
Command create-profiler executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin create-profiler --help` at the command line.

To Delete a Profiler

The remote `delete-profiler` command enables you to delete a profiler element from the Java configuration.

1. **Ensure that the server is running.**
   Remote commands require a running server.
2. **If necessary, notify users that the profiler is being deleted.**
3. **Delete the profiler by using the `delete-profiler`(1) command.**
4. **To apply your changes, restart Enterprise Server.**
   a. **Stop Enterprise Server.**
      For instructions, see "To Stop a Domain (or Server)" on page 35.
   b. **Start Enterprise Server.**
      For instructions, see "To Start a Domain (or Server)" on page 35.

Example 4–7 Deleting a Profiler

The following example command deletes the profiler named `sample_profiler`:

```
delete-profiler sample_profiler
```

Information similar to the following is displayed:

```
Command delete-profiler executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin delete-profiler --help` at the command line.

◆ ◆ ◆ CHAPTER 5

Administering Database Connectivity

This chapter provides procedures for performing database connectivity tasks in the Sun GlassFish™ Enterprise Server v3 Prelude environment by using the `asadmin` command-line utility.

The following topics are addressed here:

Instructions for accomplishing these tasks by using the Admin Console are contained in the Admin Console online help.

Setting Up the Database

Most applications use relational databases to store, organize, and retrieve data. Applications access relational databases through the Java™ Database Connectivity (JDBC) API.

The following topics are addressed here:

▼ To Install the Database and Database Driver

1 Install a supported database product.

To see the current list of database products supported by Enterprise Server, refer to the *Sun GlassFish Enterprise Server v3 Prelude Release Notes*.

2 **Install a supported JDBC driver for the database product.**

For a list of drivers supported by Enterprise Server, see "Configuration Specifics for JDBC Drivers" on page 73.

3 **Make the JAR file for the JDBC driver accessible to the domain administration server (DAS).**

See "Integrating the JDBC Driver" on page 73.

4 **Create the database.**

The application provider usually delivers scripts for creating and populating the database.

▼ To Start the Database

Enterprise Server includes an implementation of Java DB, however, you can use any JDBC-compliant database. The database is not started automatically when you start Enterprise Server, so if you have applications that require a database, you need to start Java DB manually by using the local `start-database` command.

- **Start the database by using the `start-database(1)` command.**

When the database server starts, or a client connects to it successfully, the following files are created at the location that is specified by the `--dbhome` option:

- The `derby.log` file contains the database server process log along with its standard output and standard error information.
- The database files contain your schema (for example, database tables).

Example 5–1 Starting a Database

The following example command starts Java DB on port 5001 of `localhost`:

```
asadmin start-database --dbhost=localhost --dbport=5001
```

Information similar to the followingt is displayed (partial output):

```
Database started in Network Server mode on host localhost and port 5001.
--------- Derby Network Server Information --------
Version: CSS10020/10.2.2.1 - (538595)  Build: 538595  DRDA Product Id: CSS10020
-- listing properties --
derby.drda.traceDirectory=C:\prelude\v3_prelude_release\distrib...
derby.drda.maxThreads=0
derby.drda.keepAlive=true
derby.drda.minThreads=0
derby.drda.portNumber=5001
derby.drda.logConnections=false
```

```
derby.drda.timeSlice=0
derby.drda.startNetworkServer=false
derby.drda.host=localhost
derby.drda.traceAll=false
.
.
.
Starting database in the background.
Log redirected to C:\prelude\v3_prelude_release\distributions\web\target\glassfi
sh\databases\derby.log.
Command start-database executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin start-database --help` at the command line.

▼ To Stop the Database

The local `stop-database` command enables you to stop Java DB on a specified port. A single host can have multiple database server processes running on different ports.

1. **If necessary, notify users that the database is being stopped.**
2. **Stop the database by using the `stop-database(1)` command.**

Example 5–2 Stopping a Database

The following example command stops Java DB on port 5001 of `localhost`:

```
asadmin stop-database --dbhost=localhost --dbport=5001
```

Information similar to the following is displayed:

```
onnection obtained for host: localhost, port number 5001.
Apache Derby Network Server - 10.2.2.1 - (538595) shutdown at 2008-10-17 23:34:2
7.218 GMT
Command stop-database executed successfully.
```

Troubleshooting For a laptop that roams between networks, you might have trouble shutting down the database. If you start Java DB and then change your IP address, you will not be able to stop Java DB unless you add a specific `--dbhost` argument. For example, if you run `asadmin start-database --dbhost = 0.0.0.0`, and then disconnect Ethernet and switch to wifi, you should run a command similar to the following to stop the database:

```
asadmin stop-database --dbhost localhost
```

See Also To see the full syntax and options of the command, type `asadmin stop-database --help` at the command line.

Java DB Utility Scripts

The Java DB configuration that is available for use with Enterprise Server includes scripts that can help you use Java DB. The following scripts are available in the *as-install*`/javadb/frameworks/NetworkServer/bin` directory:

`startNetworkServer,startNetworkServer.bat`
Script to start the network server

`stopNetworkServer,stopNetworkServer.bat`
Script to stop the network server

`ij,ij.bat`
Interactive JDBC scripting tool

`dblook,dblook.bat`
Script to view all or part of the DDL for a database

`sysinfo, sysinfo.bat`
Script to display versioning information about the Java DB environment

`NetworkServerControl,NetworkServerControl.bat`
Script to execute commands on the `NetworkServerControl` API

▼ To Configure Your Environment to Run Java DB Utility Scripts

1. **Ensure that the `JAVA_HOME` environment variable specifies the directory where the JDK is installed.**
2. **Set the `DERBY_HOME` environment variable to point to the *as-install*`/javadb` directory.**

See Also For more information about these utilities, see the following documentation:

- *Derby Tools and Utilities Guide* (`http://db.apache.org/derby/docs/10.1/tools/`)
- *Derby Server and Administration Guide* (`http://db.apache.org/derby/docs/10.1/adminguide/`)

Configuring Access to the Database

After establishing the database, you are ready to set up access for Enterprise Server applications. Before an application can access a database, the application must get a connection. At runtime, the following sequence occurs when an application connects to a database:

1. The application gets the JDBC resource (data source) associated with the database by making a call through the JNDI API.

   Using the JNDI name of the resource, the naming and directory service locates the JDBC resource. Each JDBC resource specifies a connection pool.

2. Using the JDBC resource, the application gets a database connection.

   Enterprise Server retrieves a physical connection from the connection pool that corresponds to the database. The pool defines connection attributes such as the database name (URL), user name, and password.

3. After the database connection is established, the application can read, modify, and add data to the database.

   The application accesses the database by making calls to the JDBC API. The JDBC driver translates the application's JDBC calls into the protocol of the database server.

4. When the application is finished accessing the database, the application closes the connection and returns the connection to the connection pool.

The following topics are addressed here:

- "Administering JDBC Connection Pools" on page 67
- "Administering JDBC Resources" on page 71
- "Integrating the JDBC Driver" on page 73

Administering JDBC Connection Pools

A *JDBC connection pool* is a group of reusable connections for a particular database. Because creating each new physical connection is time consuming, Enterprise Server maintains a pool of available connections. When an application requests a connection, it obtains one from the pool. When an application closes a connection, the connection is returned to the pool.

A JDBC resource is created by specifying the connection pool with which the resource is associated. Multiple JDBC resources can specify a single connection pool. The properties of connection pools can vary with different database vendors. Some common properties are the database name (URL), the user name, and the password.

The following tasks and information are used to administer JDBC connection pools:

- "To Create a JDBC Connection Pool" on page 68
- "To List JDBC Connection Pools" on page 69

- "To Contact (Ping) a Connection Pool" on page 69
- "To Delete a JDBC Connection Pool" on page 70

To Create a JDBC Connection Pool

The remote `create-jdbc-connection-pool` command enables you to register a new JDBC connection pool with the specified JDBC connection pool name. A JDBC connection pool or a connector connection pool can be created with authentication. You can either use a command option to specify user, password, or other connection information using the `asadmin` utility, or specify the connection information in the XML descriptor file.

When you are building the connection pool, certain data specific to the JDBC driver and the database vendor will be required. You can find some of the following specifics in"Configuration Specifics for JDBC Drivers" on page 73:

- Database vendor name
- Resource type, such as `javax.sql.DataSource` (local transactions only) `javax.sql.XADataSource` (global transactions)
- Data source class name
- Required properties, such as the database name (URL), user name, and password

Creating a JDBC connection pool is a dynamic event and does not require server restart.

Before You Begin Before creating the connection pool, you must first install and integrate the database and its associated JDBC driver. For instructions, see "Setting Up the Database" on page 63.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Create the JDBC connection pool by using the `create-jdbc-connection-pool(1)` command.**

Example 5–3 Creating a JDBC Connection Pool

The following example command creates a JDBC connection pool named `sample_derby_pool` on `localhost`:

```
asadmin create-jdbc-connection-pool
--datasourceclassname org.apache.derby.jdbc.ClientDataSource
--restype javax.sql.XADataSource
--property portNumber=1527:password=APP:user=APP:serverName=
localhost:databaseName=sun-appserv-samples:connectionAttribut
es=\;create\\=true sample_derby_pool
```

Information similar to the following is displayed:

```
Command create-jdbc-connection-pool executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin create-jdbc-connection-pool --help` at the command line.

To List JDBC Connection Pools

The remote `list-jdbc-connection-pools` command enables you to list all existing JDBC connection pools.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **List the JDBC connection pools by using the `list-jdbc-connection-pools(1)` command.**

Example 5–4 Listing JDBC Connection Pools

The following example command lists the JDBC connection pools that are on `localhost`:

```
asadmin list-jdbc-connection-pools
```

Information similar to the following is displayed:

```
sample_derby_pool2
poolA
__TimerPool
DerbyPool
sample_derby_pool
Command list-jdbc-connection-pools executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin list-jdbc-connection-pools --help` at the command line.

To Contact (Ping) a Connection Pool

The remote `ping-connection-pool` command tests if a connection pool is usable. For example, if you create a new JDBC connection pool for an application that is expected to be deployed later, you can test the JDBC pool with this command before the application is deployed.

Before You Begin Before you can contact a connection pool, the connection pool must be created with authentication, and the server or database must be running.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Ping a connection pool by using the `ping-connection-pool`(1) command.**

Example 5–5 Contacting a Connection Pool

The following example command tests to see if the `DerbyPool` connection pool is usable:

```
asadmin ping-connection-pool DerbyPool
```

Information similar to the following is displayed if the connection pool is usable:

```
Command ping-connection-pool executed successfully
```

See Also To see the full syntax and options of the command, type `asadmin ping-connection-pool --help` at the command line.

▼ To Delete a JDBC Connection Pool

The remote `delete-jdbc-connection-pool` command enables you to delete an existing JDBC connection pool. Deleting a JDBC connection pool is a dynamic event and does not require server restart.

Before You Begin Before deleting a JDBC connection pool, all associations to the resource must be removed.

1. **Ensure that the server is running.**
   Remote commands require a running server.
2. **Obtain the exact name of the JDBC connection pool that you are deleting.**
   To list the existing JDBC connection pools:
   ```
   asadmin list-jdbc-connection-pools
   ```
3. **If necessary, notify users that the JDBC connection pool is being deleted.**
4. **Delete the connection pool by using the `delete-jdbc-connection-pool`(1) command.**

Example 5–6 Deleting a JDBC Connection Pool

The following example command deletes the JDBC connection pool named `DerbyPool`:

```
asadmin delete-jdbc-connection-pool jdbc/DerbyPool
```

Information similar to the following is displayed if the connection pool is usable:

```
Command delete-jdbc-connection-pool executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin delete-jdbc-connection-pool --help` at the command line.

Administering JDBC Resources

A *JDBC resource*, also known as a data source, provides an application with a means of connecting to a database. Typically, you create a JDBC resource for each database that is accessed by the applications deployed in a domain. Multiple JDBC resources can be specified for a database.

A JDBC resource is created by specifying the connection pool with which the resource will be associated . Unse a unique Java Naming and Directory Interface (JNDI) name to identify the resource. For example, the JNDI name for the resource of a payroll database might be `java:comp/env/jdbc/payrolldb`.

The following tasks and information are used to administer JDBC resources:

- "To Create a JDBC Resource" on page 71
- "To List JDBC Resources" on page 72
- "To Delete a JDBC Resource" on page 72

▼ To Create a JDBC Resource

The remote `create-jdbc-resource` command enables you to create a JDBC resource. Creating a JDBC resource is a dynamic event and does not require server restart.

Because all JNDI names are in the `java:comp/env` subcontext, when specifying the JNDI name of a JDBC resource in the Administration Console, use only the `jdbc/`*name* format is used. For example, a payroll database might be specified as `jdbc/payrolldb`.

Before You Begin Before creating a JDBC resource, you must first create a JDBC connection pool. For instructions, see "To Create a JDBC Connection Pool" on page 68.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Create a JDBC resource by using the `create-jdbc-resource`(1) command.**

3. **If necessary, notify users that the new resource has been created.**

Example 5–7 Creating a JDBC Resource

The following example command creates a JDBC resource named `DerbyPool`:

```
asadmin create-jdbc-resource --connectionpoolid DerbyPool jdbc/DerbyPool
```

Information similar to the following is displayed if the connection pool is usable:

```
Command create-jdbc-resource executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin create-jdbc-resource --help` at the command line.

▼ To List JDBC Resources

The remote `list-jdbc-resources` command enables you to list the existing JDBC resources.

1 Ensure that the server is running.

Remote commands require a running server.

2 List JDBC resources by using the `list-jdbc-resources`(1) command.

Example 5–8 Listing JDBC Resources

The following example command lists JDBC resources for `localhost`:

```
asadmin list-jdbc-resources
```

Information similar to the following is displayed:

```
jdbc/__TimerPool
jdbc/DerbyPool
jdbc/__default
jdbc1
Command list-jdbc-resources executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin list-jdbc-resources --help` at the command line.

▼ To Delete a JDBC Resource

This remote command enables you to delete an existing JDBC resource. Deleting a JDBC resource is a dynamic event and does not require server restart.

Before You Begin Before deleting a JDBC resource, all associations with this resource must be removed.

1. **Ensure that the server is running.**
   Remote commands require a running server.
2. **Obtain the exact name of the JDBC resource that you are deleting.**
   To list the existing JDBC resources:
   ```
   asadmin list-jdbc-resources
   ```
3. **If necessary, notify users that the JDBC resource is being deleted.**
4. **Delete a JDBC resource by using the `delete-jdbc-resource`(1) command.**

Example 5–9 Deleting a JDBC Resource

The following example command deletes a JDBC resource named `DerbyPool`:

```
asadmin delete-jdbc-resource jdbc/DerbyPool
```

Information similar to the following is displayed:

```
Command delete-jdbc-resource executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin delete-jdbc-resource --help` at the command line.

Integrating the JDBC Driver

After setting up the connection pool and resources, integrate the JDBC driver in either of the following ways:

- Make the driver accessible to the common class loader, and restart the domain.
  Copy the driver's JAR and ZIP files into the *domain-dir*`/lib` directory, or copy its class files into the *domain-dir*`/lib/ext` directory.
- Identify the fully-qualified path name for the driver's JAR file.

Configuration Specifics for JDBC Drivers

Enterprise Server is designed to support connectivity to any database management system by using a corresponding JDBC driver.

- "JDBC Drivers, Full Support " on page 74
- "JDBC Drivers, Limited Support " on page 78

JDBC Drivers, Full Support

The following JDBC driver and database combinations have been tested and are supported for container-managed persistence:

To see the most current list of supported JDBC drivers, refer to the *Sun GlassFish Enterprise Server v3 Prelude Release Notes*.

Sun GlassFish JDBC Driver for DB2 Databases

The JAR files for this driver are `smbase.jar`, `smdb2.jar`, and `smutil.jar`. Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.
- **Database Vendor:** DB2
- **DataSource Classname:** `com.sun.sql.jdbcx.db2.DB2DataSource`
- **Properties:**
  - **`serverName`** – Specify the host name or IP address of the database server.
  - **`portNumber`** – Specify the port number of the database server.
  - **`databaseName`** – Set as appropriate.
  - **`user`** – Set as appropriate.
  - **`password`** – Set as appropriate.
- **URL:** `jdbc:sun:db2://`*serverName*`:`*portNumber*`;databaseName=`*databaseName*

Sun GlassFish JDBC Driver for Oracle 8.1.7 and 9.x Databases

The JAR files for this driver are `smbase.jar`, `smoracle.jar`, and `smutil.jar`. Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.
- **Database Vendor:** Oracle
- **DataSource Classname:** `com.sun.sql.jdbcx.oracle.OracleDataSource`

- **Properties:**
  - **`serverName`** – Specify the host name or IP address of the database server.
  - **`portNumber`** – Specify the port number of the database server.
  - **`SID`** – Set as appropriate.
  - **`user`** – Set as appropriate.
  - **`password`** – Set as appropriate.
- **URL:** `jdbc:sun:oracle://`*serverName*`[:`*portNumber*`][;SID=`*databaseName*`]`

Sun GlassFish JDBC Driver for Microsoft SQL Server Databases

The JAR files for this driver are `smbase.jar`, `smsqlserver.jar`, and `smutil.jar`. Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.
- **Database Vendor:** Microsoft SQL Server
- **DataSource Classname:** `com.sun.sql.jdbcx.sqlserver.SQLServerDataSource`
- **Properties:**
  - **`serverName`** – Specify the host name or IP address and the port of the database server.
  - **`portNumber`** – Specify the port number of the database server.
  - **`user`** – Set as appropriate.
  - **`password`** – Set as appropriate.
  - **`selectMethod`** – Set to `cursor`.
- **URL:** `jdbc:sun:sqlserver://`*serverName*`[:`*portNumber*`]`

Sun GlassFish JDBC Driver for Sybase Databases

The JAR files for this driver are `smbase.jar`, `smsybase.jar`, and `smutil.jar`. Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.
- **Database Vendor:** Sybase
- **DataSource Classname:** `com.sun.sql.jdbcx.sybase.SybaseDataSource`
- **Properties:**
  - **`serverName`** – Specify the host name or IP address of the database server.
  - **`portNumber`** – Specify the port number of the database server.
  - **`databaseName`** – Set as appropriate. This is optional.
  - **`user`** – Set as appropriate.
  - **`password`** – Set as appropriate.

- **URL:** `jdbc:sun:sybase://`*serverName*`[:`*portNumber*`]`

IBM DB2 Type 2 Driver

The JAR files for the DB2 driver are `db2jcc.jar`, `db2jcc_license_cu.jar`, and `db2java.zip`. Set your environment variables . For example:

```
LD_LIBRARY_PATH=/usr/db2user/sqllib/lib:${j2ee.home}/lib
DB2DIR=/opt/IBM/db2/V8.2
DB2INSTANCE=db2user
INSTHOME=/usr/db2user
VWSPATH=/usr/db2user/sqllib
THREADS_FLAG=native
```

Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.
- **Database Vendor:** DB2
- **DataSource Classname:** `com.ibm.db2.jcc.DB2SimpleDataSource`
- **Properties:**
  - **`databaseName`** - Set as appropriate.
  - **`user`** – Set as appropriate.
  - **`password`** – Set as appropriate.
  - **`driverType`** – Set to `2`.
  - **`deferPrepares`** – Set to `false`.

Java DB/Derby Type 4 Driver

The JAR file for the Java DB/Derby driver is `derbyclient.jar`. (Java DB is based upon Apache Derby.) Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.
- **Database Vendor:** Java DB/Derby
- **DataSource Classname:** Specify one of the following:

  ```
  org.apache.derby.jdbc.ClientDataSource
  org.apache.derby.jdbc.ClientXADataSource
  ```

- **Properties:**
  - **`serverName`** – Specify the host name or IP address of the database server.
  - **`portNumber`** – Specify the port number of the database server if it is different from the default.

- **`databaseName`** – Specify the name of the database.
- **`user`** - Specify the database user.

  This is only necessary if Derby is configured to use authentication. Derby does *not* use authentication by default. When the user is provided, it is the name of the schema where the tables reside.
- **`password`** – Specify the database password.

  This is only necessary if Java DB/Derby is configured to use authentication.

- **URL:** `jdbc:derby://`*serverName*`:`*portNumber*`/`*databaseName*`;create=true`

  Include the `;create=true` part only if you want the database to be created if it does not exist.

JConnect Type 4 Driver for Sybase ASE 12.5 Databases

The JAR file for the Sybase driver is `jconn2.jar`. Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.
- **Database Vendor:** Sybase
- **DataSource Classname:** Specify one of the following:

  ```
  com.sybase.jdbc2.jdbc.SybDataSource
  com.sybase.jdbc2.jdbc.SybXADataSource
  ```

- **Properties:**
  - **`serverName`** – Specify the host name or IP address of the database server.
  - **`portNumber`** – Specify the port number of the database server.
  - **`databaseName`** – Set as appropriate. Do not specify the complete URL, only the database name.
  - **`user`** – Set as appropriate.
  - **`password`** – Set as appropriate.
  - **`BE_AS_JDBC_COMPLIANT_AS_POSSIBLE`** – Set to `true`.
  - **`FAKE_METADATA`** – Set to `true`.

MM MySQL Type 4 Driver (Non-XA)

The JAR file for the MySQL driver is `mysql-connector-java-`*version*`-bin-g.jar`, for example, `mysql-connector-java-3.1.12-bin-g.jar`. Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.

- **Database Vendor:** Microsoft SQL Server
- **DataSource Classname:** Specify one of the following:

  ```
  com.mysql.jdbc.jdbc2.optional.MysqlDataSource
  ```

- **Properties:**
  - **`serverName`** – Specify the host name or IP address of the database server.
  - **`portNumber`** – Specify the port number of the database server.
  - **`databaseName`** – Set as appropriate.
  - **`user`** – Set as appropriate.
  - **`password`** – Set as appropriate.
  - **`URL`** – If you are using global transactions, you can set this property instead of `serverName`, `port`, and `databaseName`.

    The MM MySQL Type 4 driver doesn't provide a method to set the required `relaxAutoCommit` property, so you must set it indirectly by setting the **`URL`** property:

    ```
    jdbc:mysql://host:port/database?relaxAutoCommit="true"
    ```

JDBC Drivers, Limited Support

The following JDBC drivers can also be used with Enterprise Server, but have not been fully tested. Although Sun offers no product support for these drivers, Sun does offer limited support for the use of these drivers with Enterprise Server:

Note – An Oracle database user running the `capture-schema` command needs ANALYZE ANY TABLE privileges if that user does not own the schema. These privileges are granted to the user by the database administrator. For information about `capture-schema`, see *Sun GlassFish Enterprise Server v3 Prelude Reference Manual*.

CloudScape 5.1 Type 4 Driver

The JAR files for the CloudScape driver are `db2j.jar`, `db2jtools.jar`, `db2jcview.jar`, `jh.jar`, `db2jcc.jar`, and `db2jnet.jar`. Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.
- **Database Vendor:** Cloudscape
- **DataSource Classname:** `com.ibm.db2.jcc.DB2DataSource`
- **Properties:**
  - **`user`** – Set as appropriate.
  - **`password`** – Set as appropriate.
  - **`databaseName`** – Set as appropriate.

IBM Informix Type 4 Driver

Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.
- **Database Vendor:** Informix
- **DataSource Classname:** Specify one of the following:

  ```
  com.informix.jdbcx.IfxDataSource
  com.informix.jdbcx.IfxXADataSource
  ```

- **Properties:**
  - **`serverName`** – Specify the Informix database server name.
  - **`portNumber`** – Specify the port number of the database server.
  - **`databaseName`** – Set as appropriate. This is optional.
  - **`user`** – Set as appropriate.
  - **`password`** – Set as appropriate.
  - **`IfxIFXHost`** – Specify the host name or IP address of the database server.

Inet Oraxo JDBC Driver for Oracle 8.1.7 and 9.x Databases

The JAR file for the Inet Oracle driver is `Oranxo.jar`. Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.
- **Database Vendor:** Oracle
- **DataSource Classname:** `com.inet.ora.OraDataSource`
- **Properties:**

- **`serverName`** – Specify the host name or IP address of the database server.
- **`portNumber`** – Specify the port number of the database server.
- **`user`** – Specify the database user.
- **`password`** – Specify the database password.
- **`serviceName`** – Specify the URL of the database. The syntax is as follows:

  ```
  jdbc:inetora:server:port:dbname
  ```

  For example:

  ```
  jdbc:inetora:localhost:1521:payrolldb
  ```

  In this example,`localhost` is the name of the host running the Oracle server, `1521` is the Oracle server's port number, and `payrolldb` is the SID of the database. For more information about the syntax of the database URL, see the Oracle documentation.
- **`streamstolob`** - If the size of BLOB or CLOB data types exceeds 4 KB and this driver is used for CMP, this property must be set to `true`.
- **`xa-driver-does-not-support-non-tx-operations`** - Set to the value `true`. Only needed if both non-XA and XA connections are retrieved from the same connection pool. Might degrade performance.

  As an alternative to setting this property, you can create two connection pools, one for non-XA connections and one for XA connections.

Inet Merlia JDBC Driver for Microsoft SQL Server Databases

The JAR file for the Inet Microsoft SQL Server driver is `Merlia.jar`. Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.
- **Database Vendor:** Microsoft SQL Server
- **DataSource Classname:** `com.inet.tds.TdsDataSource`
- **Properties:**
  - **`serverName`** – Specify the host name or IP address and the port of the database server.
  - **`portNumber`** – Specify the port number of the database server.
  - **`user`** – Set as appropriate.
  - **`password`** – Set as appropriate.

Inet Sybelux JDBC Driver for Sybase Databases

The JAR file for the Inet Sybase driver is `Sybelux.jar`. Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.
- **Database Vendor:** Sybase
- **DataSource Classname:** `com.inet.syb.SybDataSource`
- **Properties:**
  - **`serverName`** – Specify the host name or IP address of the database server.
  - **`portNumber`** – Specify the port number of the database server.
  - **`databaseName`** – Set as appropriate. Do not specify the complete URL, only the database name.
  - **`user`** – Set as appropriate.
  - **`password`** – Set as appropriate.

MM MySQL Type 4 Driver (XA Only)

The JAR file for the MySQL™ driver is `mysql-connector-java-`*version*`-bin-g.jar`, for example, `mysql-connector-java-3.1.12-bin-g.jar`. Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.
- **Database Vendor:** Microsoft SQL Server
- **DataSource Classname:**

  `com.mysql.jdbc.jdbc2.optional.MysqlXADataSource`
- **Properties:**
  - **`serverName`** – Specify the host name or IP address of the database server.
  - **`portNumber`** – Specify the port number of the database server.
  - **`databaseName`** – Set as appropriate.
  - **`user`** – Set as appropriate.
  - **`password`** – Set as appropriate.
  - **`URL`** – If you are using global transactions, you can set this property instead of `serverName`, `port`, and `databaseName`.

    The MM MySQL Type 4 driver doesn't provide a method to set the required `relaxAutoCommit` property, so you must set it indirectly by setting the `URL` property:

```
jdbc:mysql://host:port/database?relaxAutoCommit="true"
```

OCI Oracle Type 2 Driver for Oracle 8.1.7 and 9.x Databases

The JAR file for the OCI Oracle driver is `ojdbc14.jar`. Make sure that the shared library is available through `LD_LIBRARY_PATH` and that the `ORACLE_HOME` property is set. Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.
- **Database Vendor:** Oracle
- **DataSource Classname:** Specify one of the following:

  ```
  oracle.jdbc.pool.OracleDataSource
  oracle.jdbc.xa.client.OracleXADataSource
  ```

- **Properties:**
  - **`user`** – Set as appropriate.
  - **`password`** – Set as appropriate.
  - **`URL`** – Specify the complete database URL using the following syntax:

    ```
    jdbc:oracle:oci:[user/password]@host[:port]/service
    ```

    For example:

    ```
    jdbc:oracle:oci:@localhost:1521:customer_db
    ```

  - **`xa-driver-does-not-support-non-tx-operations`** - Set to the value `true`. Only needed if both non-XA and XA connections are retrieved from the same connection pool. Might degrade performance.

    As an alternative to setting this property, you can create two connection pools, one for non-XA connections and one for XA connections.

Oracle Thin Type 4 Driver for Oracle 8.1.7 and 9.x Databases

The JAR file for the Oracle driver is `ojdbc14.jar`.

Note – When using this driver, keep in mind that you cannot insert more than 2000 bytes of data into a column. To circumvent this problem, use the OCI driver (JDBC type 2).

Configure the connection pool using the following settings:

- **Name:** Use this name when you configure the JDBC resource later.
- **Resource Type:** Specify the appropriate value.

- **Database Vendor:** Oracle
- **DataSource Classname:** Specify one of the following:

  ```
  oracle.jdbc.pool.OracleDataSource
  oracle.jdbc.xa.client.OracleXADataSource
  ```

- **Properties:**
  - **`user`** – Set as appropriate.
  - **`password`** – Set as appropriate.
  - **`URL`** – Specify the complete database URL using the following syntax:

    ```
    jdbc:oracle:thin:[user/password]@host[:port]/service
    ```

    For example:

    ```
    jdbc:oracle:thin:@localhost:1521:customer_db
    ```

  - **`xa-driver-does-not-support-non-tx-operations`** - Set to the value `true`. Optional: only needed if both non-XA and XA connections are retrieved from the same connection pool. Might degrade performance.

    As an alternative to setting this property, you can create two connection pools, one for non-XA connections and one for XA connections.

Note – For the Oracle thin driver, the `XAResource.recover` method repeatedly returns the same set of in-doubt Xids regardless of the input flag. According to the XA specifications, the Transaction Manager initially calls this method with TMSTARTSCAN and then with TMNOFLAGS repeatedly until no Xids are returned. The `XAResource.commit` method also has some issues.

To disable this Enterprise Server workaround, the `oracle-xa-recovery-workaround` property value must be set to `false`.

◆ ◆ ◆ CHAPTER 6

Administering System Security

This chapter provides instructions for administering system security in the Sun GlassFish™ Enterprise Server v3 Prelude environment by using the `asadmin` and `keytool` command-line utilities.

The following topics are addressed here:

Instructions for accomplishing many of these tasks by using the Administration Console are contained in the Administration Console online help.

Additional information on security is contained in Chapter 7, "Administering User Security."

About Enterprise Server Security

Security is about protecting data, that is, how to prevent unauthorized access or damage to data that is in storage or in transit. The Enterprise Server is built on the Java security model, which uses a sandbox where applications can run safely, without potential risk to systems or users. *System security* affects all the applications in the Enterprise Server environment.

System security features include the following:

Authentication

Authentication is the way in which an entity (a user, an application, or a component) determines that another entity is who it claims to be. An entity uses security *credentials* to authenticate itself. The credentials might be a user name and password, a digital certificate, or something else. Usually, servers or applications require clients to authenticate themselves. Additionally, clients might require servers to authenticate themselves. When authentication is bidirectional, it is called *mutual authentication*.

When an entity tries to access a protected resource, Enterprise Server uses the authentication mechanism configured for that resource to determine whether to grant access. For example, a user can enter a user name and password in a web browser, and if the application verifies those credentials, the user is authenticated. The user is associated with this authenticated security identity for the remainder of the session.

Authentication Types

Within its deployment descriptors, an application specifies the type of authentication that it uses. Enterprise Server supports the following types of authentication:

BASIC	Uses the server's built-in login dialog box. The communication protocol is HTTP (SSL optional). There is no user-credentialed encryption unless using SSL.
FORM	The application provides its own custom login and error pages. The communication protocol is HTTP (SSL optional). There is no user-credentialed encryption unless using SSL.
CLIENT-CERT	The server authenticates the client using a public key certificate. The communication protocol is HTTPS (HTTP over SSL). User-credentialed encryption is SSL.

Passwords

Passwords are your first line of defense against unauthorized access to the components and data of Enterprise Server. For information on how to use passwords for Enterprise Server, see "Administering Passwords" on page 92.

Administration Password

The administration password, also known as the admin password, is used to invoke the Administration Console and the `asadmin` utility. This password is usually set during installation, but it can be changed. For instructions, see "To Change the Administration Password" on page 93.

Encoded Passwords

Files that contain encoded passwords need to be protected using file system permissions. These files include the following:

- *domain-dir*`/master-password`

  This file contains the encoded master password and should be protected with file system permissions 600.
- Any password file created to pass as an argument by using the `--passwordfile` argument to the `asadmin` utility should be protected with file system permissions 600.

For instructions, see "To Set a Password From a File" on page 93.

Single Sign-on

With *single sign-on*, a user who logs in to one application becomes implicitly logged in to other applications that require the same authentication information. Single sign-on is based on groups. All web applications whose deployment descriptor defines the same group and uses the same authentication method (BASIC, FORM, or CLIENT-CERT) share single sign-on.

On Enterprise Server, single sign-on is enabled by default for virtual servers, allowing multiple applications in one virtual server to share the user authentication state.

Authorization

Authorization, also known as access control, is the means by which users are granted permission to access data or perform operations. After a user is authenticated, the user's level of authorization determines what operations the owner can perform. A user's authorization is based on the user's role.

Roles

A *role* defines which applications and what parts of each application users can access and what those users or groups can do with the applications. For example, in a personnel application, all employees might be able to see phone numbers and email addresses, but only managers have access to salary information. This application would define at least two roles: `employee` and `manager`. Only users in the `manager` role are allowed to view salary information.

A role is different from a group in that a role defines a function in an application, while a group is a set of users who are related in some way. For example, the personnel application specify groups such as `full-time`, `part-time`, and `on-leave`. Users in these groups are all employees (the `employee` role). In addition, each user has its own designation that defines an additional level of employment.

Roles are defined in the deployment descriptor for the application. The application developer or deployer maps roles to one or more groups in the deployment descriptor for each application. When the application is being packaged and deployed, the application specifies mappings between users, groups, and roles, as illustrated in the following figure.

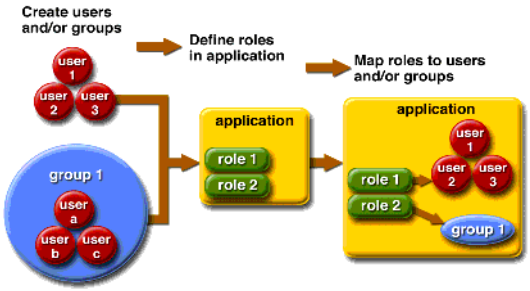

FIGURE 6–1 Role Mapping

Java Authorization Contract for Containers

Java Authorization Contract for Containers (JACC) is the part of the Java EE specification that defines an interface for pluggable authorization providers. This enables you to set up third-party plug-in modules to perform authorization. By default, the Enterprise Server provides a simple, file-based authorization engine that complies with the JACC specification. You can also specify additional third-party JACC providers.

JACC providers use the Java Authentication and Authorization Service (JAAS) APIs. JAAS enables services to authenticate and enforce access controls upon users. JAAS implements a Java technology version of the standard Pluggable Authentication Module (PAM) framework.

Auditing

Auditing is the means used to capture security-related events for the purpose of evaluating the effectiveness of security measures. Enterprise Server uses audit modules to capture audit trails of all authentication and authorization decisions. Enterprise Server provides a default audit module, as well as the ability to customize the audit modules.

For administration instructions, see "Administering Audit Modules" on page 94.

Firewalls

A *firewall* controls the flow of data between two or more networks, and manages the links between the networks. A firewall can consist of both hardware and software elements. The following guidelines pertain primarily to Enterprise Server:

- In general, firewalls should be configured so that clients can access the necessary TCP/IP ports.

  For example, if the HTTP listener is operating on port 8080, configure the firewall to allow HTTP requests on port 8080 only. Likewise, if HTTPS requests are set up for port 8181, you must configure the firewalls to allow HTTPS requests on port 8181.
- In double firewall architecture, you must configure the outer firewall to allow for HTTP and HTTPS transactions. You must configure the inner firewall to allow the HTTP server plug-in to communicate with Enterprise Server behind the firewall.

Certificates and SSL

The following topics are addressed here:

For administration instructions, see "Administering JSSE Certificates " on page 96.

Certificates

Certificates, also called digital certificates, are electronic files that uniquely identify people and resources on the Internet. Certificates also enable secure, confidential communication between two entities. There are different kinds of certificates:

- *Personal certificates* are used by individuals.
- *Server certificates* are used to establish secure sessions between the server and clients through secure sockets layer (SSL) technology.

Certificates are based on *public key cryptography*, which uses pairs of digital keys (very long numbers) to encrypt, or encode, information so the information can be read only by its intended recipient. The recipient then decrypts (decodes) the information to read it. A *key pair* contains a public key and a private key. The owner distributes the public key and makes it available to anyone. But the owner never distributes the private key, which is always kept secret. Because the keys are mathematically related, data encrypted with one key can only be decrypted with the other key in the pair.

Certificates are issued by a trusted third party called a *Certification Authority* (CA). The CA is analogous to a passport office: it validates the certificate holder's identity and signs the certificate so that it cannot be forged or tampered with. After a CA has signed a certificate, the holder can present it as proof of identity and to establish encrypted, confidential communications. Most importantly, a certificate binds the owner's public key to the owner's identity.

In addition to the public key, a certificate typically includes information such as the following:

- The name of the holder and other identification, such as the URL of the web server using the certificate, or an individual's email address
- The name of the CA that issued the certificate
- An expiration date

Certificates are governed by the technical specifications of the X.509 format. To verify the identity of a user in the `certificate` realm, the authentication service verifies an X.509 certificate, using the common name field of the X.509 certificate as the principal name.

Certificate Chains

A *certificate chain* is a series of certificates issued by successive CA certificates, eventually ending in a root CA certificate.

Web browsers are preconfigured with a set of root CA certificates that the browser automatically trusts. Any certificates from elsewhere must come with a certificate chain to verify their validity.

Certificate Files

During Enterprise Server installation, a certificate is generated in Java Secure Socket Extension (JSSE) format suitable for internal testing. By default, Enterprise Server stores its certificate information in certificate databases in the *domain-dir*`/config` directory:

Keystore file	The `key3.db` file contains Enterprise Server certificate, including its private key. The keystore file is protected with a password. Each keystore entry has a unique alias. After installation, the Enterprise Server keystore has a single entry with an alias of `s1as`.
Truststore file	The `cert8.db` file contains the Enterprise Server trusted certificates, including public keys for other entities. For a trusted certificate, the server has confirmed that the public key in the certificate belongs to the certificate's owner. Trusted certificates generally include those of CAs.

By default, Enterprise Server is configured with a keystore and truststore that will work with the example applications and for development purposes.

Secure Sockets Layer

Secure Sockets Layer (SSL) is the most popular standard for securing Internet communications and transactions. Secure web applications use HTTPS (HTTP over SSL). The HTTPS protocol uses certificates to ensure confidential and secure communications between server and clients. In an SSL connection, both the client and the server encrypt data before sending it. Data is decrypted upon receipt.

The newest version of the SSL standard is called Transport Layer Security (TLS). The Enterprise Server supports the SSL 3.0 and the TLS 1.0 encryption protocols.

To use SSL, Enterprise Server must have a certificate for each external interface or IP address that accepts secure connections. The HTTPS service of most web servers will not run unless a certificate has been installed.

Ciphers

A *cipher* is a cryptographic algorithm used for encryption or decryption. SSL and TLS protocols support a variety of ciphers used to authenticate the server and client to each other, transmit certificates, and establish session keys.

Some ciphers are stronger and more secure than others. Clients and servers can support different cipher suites. During a secure connection, the client and the server agree to use the strongest cipher that they both have enabled for communication, so it is usually sufficient to enable all ciphers.

Application Security

Unlike system security, which affects all the applications on Enterprise Server, *Application security* affects a particular application. There are basically two types of application security: programmatic and declarative.

- In declarative security, Enterprise Server container handles security through an application's deployment descriptors. You can control declarative security by editing deployment descriptors directly. Because deployment descriptors can change after an application is developed, declarative security allows for more flexibility.
- In programmatic security, application code handles security chores. Generally, programmatic security is discouraged since security configurations are hard coded in the application instead of being managed through the Java EE containers. Programmatic security is controlled by the application developer.

Information on application security is contained in the Chapter 4, "Securing Applications," in *Sun GlassFish Enterprise Server v3 Prelude Developer's Guide*.

Tools for Managing System Security

Enterprise Server provides the following tools for managing system security:

Administration Console	The Administration Console is a browser-based utility used to configure security for the entire server. Tasks include managing certificates, users, groups, and realms, and performing other system-wide security tasks. For a general introduction to the Administration Console, see "Administration Console" on page 25.
The `asadmin` utility	The `asadmin` command-line utility performs many of the same tasks as the Administration Console. You might be able to do some things with the `asadmin` utility that you cannot do with the Administration Console. For a general introduction to `asadmin`, see "Command-Line Utility for Administration (`asadmin`)" on page 25.
The `keytool` utility	The `keytool` Java 2 Platform, Standard Edition (J2SE) command-line utility is used for managing digital certificates and key pairs. For more information, see "Administering JSSE Certificates " on page 96.
The `policytool` utility	The `policytool` J2SE graphical utility is used for managing system-wide Java security policies. As an administrator, you rarely use `policytool`.

For more information on using `keytool`, `policytool`, and other Java security tools, see *Java 2 SDK Tools and Utilities* at `http://java.sun.com/j2se/1.4.2/docs/tooldocs/tools.html#security`.

Administering Passwords

There are multiple ways to administer passwords. You can rely on administrators to keep passwords secret and change the passwords regularly. You can set up files for storing passwords so that `asadmin` commands can access these files rather than having users type the commands.

The following topics are addressed here:

▼ To Change the Administration Password

The remote `change-admin-password` command allows you to change the administration password. You are prompted for the old and new admin passwords, with confirmation.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Change the admin password by using the `change-admin-password`(1) command.**

3. **Enter the old and new admin password as prompted.**

Example 6–1 Changing the Admin Password

The following example changes the admin password for user anonymous from `adminadmin` to `newadmin`:

```
asadmin change-admin-password --user anonymous
```

Information similar to the following is displayed:

```
Enter admin password>adminadmin
Enter new admin password>newadmin
Enter new admin password again>newadmin
Command change-admin-password executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin change-admin-password --help` at the command line.

▼ To Set a Password From a File

Instead of typing the password at the command line, you can access the password for a command from a file such as `passwords.txt`. The `--passwordfile` option of the `asadmin` utility takes the name of the file that contains the passwords. The entry for a password in the file must have the `AS_ADMIN_` prefix followed by the password name in uppercase letters.

The following other types of passwords can be specified:

```
AS_ADMIN_MASTERPASSWORD
AS_ADMIN_USERPASSWORD
AS_ADMIN_ALIASPASSWORD
```

1. **Edit the password file.**

   For example, to specify the password for the domain administration server (DAS), add an entry similar to the following to the password file, where `adminadmin` is the administrator password:

   ```
   AS_ADMIN_PASSWORD=adminadmin
   ```

2. **Save the password file.**

3. **Specify the password file in an `asadmin` command.**

   For example:

   ```
   asadmin delete-jdbc-resource --user admin --password passwords.file jdbc/DerbyPool
   ```

Troubleshooting If `AS_ADMIN_PASSWORD` has been exported to the global environment, specifying the `--passwordfile` option will produce a warning about using the `--passwordfile` option. To prevent this warning situation from happening, unset `AS_ADMIN_PASSWORD`.

Administering Audit Modules

The following topics are addressed here:

▼ To Create an Audit Module

The remote `create-audit-module` command enables you to create an audit module for the add-on component that implements the audit capabilities.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Create an audit module by using the `create-audit-module`(1) command.**

Example 6–2 Creating an Audit Module

The following example command creates an audit module named `sampleAuditModule` on `localhost`:

```
asadmin create-audit-module
--classname com.sun.appserv.auditmodule --property defaultuser=
admin:Password=admin sampleAuditModule
```

Information similar to the following is displayed:

```
Creation of AuditModule sampleAuditModule completed successfully

Command create-audit-module executed successfully.
```

▼ To List Audit Modules

The remote `list-audit-modules` command enables you to list the audit modules on one of the following targets:

- Server instance, `server` (the default)
- Specified server instance
- Specified configuration

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **List the audit modules by using the `list-audit-modules`(1) command.**

Example 6–3 Listing Audit Modules

The following example command lists the audit modules on `localhost`:

```
asadmin list-audit-modules
```

Information similar to the following is displayed:

```
audit-module : default
audit-module : sampleAuditModule
Command list-audit-modules executed successfully.
```

▼ To Delete an Audit Module

The remote `delete-audit-module` command allows you to delete an existing audit module.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2 **Delete an audit module by using the `delete-audit-module`(1) command.**

Example 6–4 Deleting an Audit Module

The following example command deletes `sampleAuditModule` from `localhost`:

```
asadmin delete-audit-module sampleAuditModule
```

Information similar to the following is displayed:

```
Deletion of Audit Module sampleAuditModule completed successfully

Command delete-audit-module executed successfully.
```

Administering JSSE Certificates

The J2SE SDK ships with the `keytool` utility, which enables you to set up and work with Java Secure Socket Extension (JSSE) digital certificates. You can administer public/private key pairs and associated certificates, and cache the public keys (in the form of certificates) of their communicating peers.

The following topics are addressed here:

- "To Generate a Certificate by Using `keytool`" on page 96
- "To Sign a Certificate by Using `keytool`" on page 98
- "To Delete a Certificate by Using `keytool`" on page 100

▼ To Generate a Certificate by Using `keytool`

By default, the `keytool` utility creates a keystore file in the directory where the utility is run.

Before You Begin To run the `keytool` utility, your shell environment must be configured so that the J2SE `/bin` directory is in the path, otherwise the full path to the utility must be present on the command line.

1 **Change to the directory that contains the keystore and truststore files.**

Always generate the certificate in the directory containing the keystore and truststore files. The default is *domain-dir*`/config`.

2 **Generate the certificate in the keystore file, `keystore.jks`, using the following command format:**

```
keytool -genkey -alias keyAlias-keyalg RSA
 -keypass changeit
 -storepass changeit
keystore keystore.jks
```

Use any unique name as your *keyAlias*. If you have changed the keystore or private key password from the default (`changeit`), substitute the new password for `changeit`. The default key password alias is `s1as`.

A prompt appears that asks for your name, organization, and other information.

3 **Export the generated certificate to the `server.cer` file (or `client.cer` if you prefer), using the following command format:**

```
keytool -export -alias keyAlias-storepass changeit
 -file server.cer
 -keystore keystore.jks
```

4 **If a certificate signed by a certificate authority is required, see "To Sign a Certificate by Using `keytool`" on page 98.**

5 **Create the `cacerts.jks` truststore file and add the certificate to the truststore, using the following command format:**

```
keytool -import -v -trustcacerts
-alias keyAlias
 -file server.cer
-keystore cacerts.jks
 -keypass changeit
```

If you have changed the keystore or private key password from the default (`changeit`), substitute the new password.

Information about the certificate is displayed and a prompt appears asking if you want to trust the certificate.

6 **Type `yes`, then press Enter.**

Information similar to the following is displayed:

```
Certificate was added to keystore
[Saving cacerts.jks]
```

7 **To apply your changes, restart Enterprise Server.**

a. **Stop Enterprise Server.**

For instructions, see "To Stop a Domain (or Server)" on page 35.

b. Start Enterprise Server.

For instructions, see "To Start a Domain (or Server)" on page 35.

Example 6–5 Creating a Self-Signed Certificate in a JKS Keystore by Using an RSA Key Algorithm

RSA is public-key encryption technology developed by RSA Data Security, Inc.

```
keytool  -genkey -noprompt -trustcacerts -keyalg RSA -alias ${cert.alias}
-dname  ${dn.name} -keypass ${key.pass} -keystore ${keystore.file}
-storepass ${keystore.pass}
```

Example 6–6 Creating a Self-Signed Certificate in a JKS Keystore by Using a Default Key Algorithm

```
keytool -genkey -noprompt -trustcacerts -alias ${cert.alias} -dname
${dn.name} -keypass ${key.pass} -keystore ${keystore.file} -storepass
${keystore.pass}
```

Example 6–7 Displaying Available Certificates From a JKS Keystore

```
keytool -list -v  -keystore ${keystore.file} -storepass ${keystore.pass}
```

Example 6–8 Displaying Certificate information From a JKS Keystore

```
keytool -list -v  -alias ${cert.alias} -keystore ${keystore.file}
-storepass ${keystore.pass}
```

See Also For more information on `keytool`, see the `keytool` documentation at `http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html`.

▼ To Sign a Certificate by Using `keytool`

After creating a certificate, the owner must sign the certificate to prevent forgery. E-commerce sites, or those for which authentication of identity is important, can purchase a certificate from a well-known Certificate Authority (CA).

Note – If authentication is not a concern, for example if private secure communications are all that is required, you can save the time and expense involved in obtaining a CA certificate by using a self-signed certificate.

---

1 Follow the instructions on the CA's web site for generating certificate key pairs.

2. **Download the generated certificate key pair.**

   Save the certificate in the directory containing the keystore and truststore files. The default is *domain-dir*`/config`.

3. **In your shell, change to the directory containing the certificate.**

4. **Import the certificate into the local keystore and, if necessary, the local truststore using the following command format:**

```
keytool -import -v -trustcacerts
-alias keyAlias
 -file server.cer
-keystore cacerts.jks
 -keypass changeit
-storepass changeit
```

   If the keystore or private key password is not the default password, then substitute the new password for the default (`changeit`).

5. **To apply your changes, restart Enterprise Server.**

   a. **Stop Enterprise Server.**

      For instructions, see "To Stop a Domain (or Server)" on page 35.

   b. **Start Enterprise Server.**

      For instructions, see "To Start a Domain (or Server)" on page 35.

Example 6–9 Importing an RFC/Text-Formatted Certificate Into a JKS Keystore

Certificates are often stored using the printable encoding format defined by the Internet Request for Comments (RFC) 1421 standard instead of their binary encoding. This certificate format, also known as Base 64 encoding, facilitates exporting certificates to other applications by email or through some other mechanism.

```
keytool -import -noprompt -trustcacerts -alias ${cert.alias} -file
${cert.file} -keystore ${keystore.file} -storepass ${keystore.pass}
```

Example 6–10 Exporting a Certificate From a JKS Keystore in PKCS7 Format

The reply format defined by the Public Key Cryptography Standards #7, Cryptographic Message Syntax Standard, includes the supporting certificate chain in addition to the issued certificate.

```
keytool -export -noprompt -alias ${cert.alias} -file ${cert.file}
-keystore ${keystore.file} -storepass ${keystore.pass}
```

Example 6–11 Exporting a Certificate From a JKS Keystore in RFC/Text Format

```
keytool -export -noprompt -rfc -alias ${cert.alias} -file
${cert.file} -keystore ${keystore.file} -storepass ${keystore.pass}
```

See Also For more information on `keytool`, see the `keytool` documentation at `http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html`

▼ To Delete a Certificate by Using `keytool`

The `keytool -delete` command enables you to delete an existing certificate.

- **Delete a certificate using the following command format:**

```
keytool -delete
 -alias keyAlias
 -keystore keystore-name
 -storepass password
```

Example 6–12 Deleting a Certificate From a JKS Keystore

```
keytool -delete -noprompt -alias ${cert.alias}  -keystore ${keystore.file}
-storepass ${keystore.pass}
```

See Also For more information on `keytool`, see the `keytool` documentation at `http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html`.

◆ ◆ ◆ CHAPTER 7

Administering User Security

This chapter provides instructions for administering user security in the Sun GlassFish™ Enterprise Server v3 Prelude environment by using the `asadmin` command-line utility.

Enterprise Server enforces its authentication and authorization policies upon realms, users, and groups. This chapter assumes that you are familiar with security features such as authentication, authorization, and certificates. If you are not, see Chapter 6, "Administering System Security."

The following topics are addressed here:

Instructions for accomplishing these tasks by using the Administration Console are contained in the Administration Console online help.

Administering Authentication Realms

An *authentication realm*, also called a *security policy domain* or *security domain*, is a scope over which the Enterprise Server defines and enforces a common security policy. Enterprise Server is preconfigured with the `file`, `certificate`, and `admin-realm` realms. In addition, you can set up `ldap`, `jdbc`, `solaris`, or custom realms. An application can specify which realm to use in its deployment descriptor. If the application does not specify a realm, Enterprise Server uses its default realm (`file`).

`file` realm	Enterprise Server stores user credentials locally in a file named `keyfile`. The `file` realm is the initial default realm.
`admin-realm` realm	The `admin-realm` is also a `file` realm and stores administrator user credentials locally in a file named `admin-keyfile`.
`certificate` realm	Enterprise Server stores user credentials in a certificate database. When using the `certificate` realm, the server uses certificates with the HTTPS protocol to authenticate web clients.

`ldap` realm	Enterprise Server gets user credentials from a Lightweight Directory Access Protocol (LDAP) server such as the Directory Server. LDAP is a protocol for enabling anyone to locate organizations, individuals, and other resources such as files and devices in a network, whether on the public Internet or on a corporate intranet. Consult your LDAP server documentation for information on managing users and groups in the `ldap` realm.
`jdbc` realm	Enterprise Server gets user credentials from a database. The server uses the database information and the enabled `jdbc` realm option in the configuration file.
Custom realm	You can create other repositories for user credentials, such as a relational database or third-party components. For more information on custom realms, see the Administration Console online help.

The Enterprise Server authentication service can govern users in multiple realms.

The following tasks and information are used to administer authentication realms:

- "To Create an Authentication Realm" on page 102
- "To List Authentication Realms" on page 103
- "To Delete an Authentication Realm" on page 103
- "To Configure a JDBC Realm" on page 104

▼ To Create an Authentication Realm

The remote `create-auth-realm` command enables you to create an authentication realm.

1 **Ensure that the server is running.**

Remote commands require a running server.

2 **Create a realm by using the `create-auth-realm`(1) command.**

Example 7–1 Creating a Realm

The following example command creates a realm named `db`:

```
asadmin create-auth-realm --classname com.iplanet.ias.security.auth.realm.DB.Database
--property defaultuser=admin:Password=admin db
```

Information similar to the following is displayed:

```
Command create-auth-realm executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin create-auth-realm --help` at the command line.

▼ To List Authentication Realms

The remote `list-auth-realms` command enables you to list the existing authentication realms.

1. **Ensure that the server is running.**
   Remote commands require a running server.
2. **List realms by using the `list-auth-realms`(1) command.**

Example 7–2 Listing Realms

The following example command lists the authentication realms on `localhost`:

```
asadmin list-auth-realms
```

Information similar to the following is displayed:

```
db
certificate
file
admin-realm
Command list-auth-realms executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin list-auth-realms --help` at the command line.

▼ To Delete an Authentication Realm

The remote `delete-auth-realm` command enables you to delete an existing authentication realm.

1. **Ensure that the server is running.**
   Remote commands require a running server.

2 **Obtain the exact name of the realm that you are deleting.**

To list the existing realms:

```
asadmin list-auth-realms
```

3 **If necessary, notify users that the realm is being deleted.**

4 **Delete the realm by using the `delete-auth-realm`(1) command.**

5 **To apply your changes, restart Enterprise Server.**

a. **Stop Enterprise Server.**

For instructions, see "To Stop a Domain (or Server)" on page 35.

b. **Start Enterprise Server.**

For instructions, see "To Start a Domain (or Server)" on page 35.

Example 7–3 Deleting a Realm

The following example command deletes an authentication realm named `db` `localhost`:

```
asadmin delete-auth-realm db
```

Information similar to the following is displayed:

```
Command delete-auth-realm executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin` `delete-auth-realm` `--help` at the command line.

▼ To Configure a `JDBC` Realm

Enterprise Server enables you to specify a user's credentials (user name and password) in the `jdbc` realm instead of in the connection pool. Using the `jdbc` realm instead of the connection pool prevents other applications from browsing the database tables for user credentials.

Note – By default, storage of passwords as clear text is not supported in the `jdbc` realm. Under normal circumstances, passwords should not be stored as clear text.

---

1 **Create the database tables in which to store user credentials for the realm.**

How you create the database tables depends on the database that you are using.

2. **Add user credentials to the database tables that you created.**

   How you add user credentials to the database tables depends on the database that you are using.

3. **Create a `jdbc` realm.**

   For instructions, see "To Create an Authentication Realm" on page 102.

4. **Modify the deployment descriptor to specify the `jdbc` realm.**

   Modify the deployment descriptor that is associated with your application.

   For more information about how to specify a realm, see "How to Configure a Realm" in *Sun GlassFish Enterprise Server v3 Prelude Developer's Guide*.

5. **Assign security roles to users in the realm.**

   To assign a security role to a user, add a `security-role-mapping` element to the deployment descriptor that you modified.

Example 7–4 Assigning a Security Role

The following example shows a `security-role-mapping` element that assigns the security role `Employee` to user `Calvin`:

```
<security-role-mapping>
    <role-name>Employee</role-name>
    <principal-name>Calvin</principal-name>
  </security-role-mapping>
```

Administering File Users

A *user* is an individual (or application program) identity that is defined in Enterprise Server. A user who has been authenticated is sometimes called a *principal*.

As the administrator, you are responsible for integrating users into the Enterprise Server environment so that their credentials are securely established and they are provided with access to the applications and services that they are entitled to use.

The following tasks are used to manage users:

- "To Create a File User" on page 106
- "To List File Users" on page 106
- "To List File Groups" on page 107
- "To Update a File User" on page 108
- "To Delete a File User" on page 108

▼ To Create a File User

The remote `create-file-user` command enables you to create a new user by adding a new entry to the `keyfile`. The entry includes the user name, password, and any groups for the user. Multiple groups can be specified by separating the groups with colons (:).

Creating a new `file` realm user is a dynamic event and does not require server restart.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **If the user will belong to a particular group, list the current file groups:**

   ```
   asadmin list-file-groups –user admin –passwordfile passwords.txt
   ```

3. **Create a file user by using the `create-file-user`(1) command.**

Example 7–5 Creating a User

The following example command create user `Jennifer` on the default realm `file` (no groups are specified):

```
asadmin create-file-user --user admin
--passwordfile=c:\tmp\asadminpassword.txt Jennifer
```

Information similar to the following is displayed:

```
Command create-file-user executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin create-file-user --help` at the command line.

▼ To List File Users

The remote `list-file-users` command enables you to list the users that are in the `keyfile`.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **List users by using the `list-file-users`(1) command.**

Example 7–6 Listing File Users

The following example command lists file users on the default `file` realm file:

```
asadmin list-file-users
```

Information similar to the following is displayed:

```
Jennifer
Command list-file-users executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin list-file-users --help` at the command line.

▼ To List File Groups

A *group* is a category of users classified by common traits, such as job title or customer profile. For example, users of an e-commerce application might belong to the `customer` group, and the big spenders might also belong to the `preferred` group. Categorizing users into groups makes it easier to control the access of large numbers of users. A group is defined for an entire server and realm. A user can be associated with multiple groups of users.

A group is different from a role in that a role defines a function in an application, while a group is a set of users who are related in some way. For example, in the personnel application there might be groups such as `full-time`, `part-time`, and `on-leave`. Users in these groups are all employees (the `employee` role). In addition, each user has its own designation that defines an additional level of employment.

The remote `list-file-groups` command lists groups for a file user, or all file groups if the `--name` option is not specified.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **List file groups by using the `list-file-groups`(1)command.**

Example 7–7 Listing Groups for a User

The following example command lists the groups for user `joesmith`:

```
asadmin list-file-groups --name joesmith
```

Information similar to the following is displayed:

```
staff
manager
Command list-file-groups executed successfully
```

▼ To Update a File User

The remote `update-file-user` command enables you to modify the information in the `keyfile` for a specified user.

1. **Ensure that the server is running.**
   Remote commands require a running server.
2. **Update the user information by using the `update-file-user`(1) command.**
3. **To apply your changes, restart Enterprise Server.**
   a. **Stop Enterprise Server.**
      For instructions, see "To Stop a Domain (or Server)" on page 35.
   b. **Start Enterprise Server.**
      For instructions, see "To Start a Domain (or Server)" on page 35.

Example 7–8 Updating a User

The following command updates the groups for user `Jennifer`:

```
asadmin update-file-user --passwordfile
c:\tmp\asadminpassword.txt --groups
staff:manager:engineer Jennifer
```

Information similar to the following is displayed:

```
Command update-file-user executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin update-file-user --help` at the command line.

▼ To Delete a File User

The remote `delete-file-user` command enables you to remove a user entry from the `keyfile` by specifying the user name.

1. **Ensure that the server is running.**
   Remote commands require a running server.
2. **Obtain the exact name of the file user that you are deleting.**
   To list the existing file users:
   ```
   asadmin list-file-users
   ```
3. **Delete the user by using the `delete-file-user`(1) command.**

Example 7–9 Deleting a User

The following example command deletes user `Jennifer` from the default `file` realm:

```
asadmin delete-file-user Jennifer
```

Information similar to the following is displayed:

```
Command delete-file-user executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin delete-file-user --help` at the command line.

◆ ◆ ◆ CHAPTER 8

Administering the HTTP Service

The HTTP service provides functionality for deploying web applications and for making deployed web applications accessible by HTTP clients. HTTP services are provided by two kinds of related objects: HTTP listeners and virtual servers. This chapter provides procedures for administering the HTTP service in the Sun GlassFish™ Enterprise Server v3 Prelude environment by using the `asadmin` command-line utility.

The following topics are addressed here:

Instructions for accomplishing these tasks by using the Administration Console are contained in the Administration Console online help.

Administering HTTP Listeners

An *HTTP listener* is a listen socket that has an Internet Protocol (IP) address, a port number, a server name, and a default virtual server. Each virtual server provides connections between the server and clients through one or more HTTP listeners. Each HTTP listener must have a unique combination of port number and IP address. For example, an HTTP listener can listen on all configured IP addresses on a given port for a host by specifying the IP address 0.0.0.0. Alternatively, the HTTP listener can specify a unique IP address for each listener while using the same port.

Because an HTTP listener is a combination of IP address and port number, you can have multiple HTTP listeners with the same IP address and different port numbers, or with different IP addresses and the same port number (if your host was configured to respond to these addresses). The host running the Enterprise Server typically has access to only one IP address. HTTP listeners typically use the 0.0.0.0 IP address and different port numbers, with each port number serving a different purpose. However, if the host does have access to more than one IP address, each address can serve a different purpose.

By default, when Enterprise Server starts, the following HTTP listeners are started automatically:

- HTTP listeners associated with the virtual server named `server`:
  - The listener named `http-listener-1` does not have security enabled.
  - The listener named `http-listener-2` has security enabled
- An HTTP listener named `admin-listener`, associated with the virtual server named `__asadmin`. This listener does not have security enabled.

To access a web application deployed on the Enterprise Server, use the URL `http://localhost:8080/` (or `https://localhost:8181/` for a secure application), along with the context root specified for the web application.

To access the Administration Console, use the URL `https://localhost:4848/` or `http://localhost:4848/asadmin/` (console default context root).

The following table describes the Enterprise Server default ports for the listeners that use ports.

TABLE 8–1 Default Ports for Listeners

Listener	Default Port	Description
Administrative server	4848	A domain's administrative server is accessed by the Administration Console and the `asadmin` utility. For the Administration Console, specify the port number in the URL of the browser. When running an `asadmin` command remotely, specify the port number by using the `--port` option.
HTTP	8080	The web server listens for HTTP requests on a port. To access deployed web applications and services, clients connect to this port.
HTTPS	8181	Web applications configured for secure communications listen on a separate port.

The following tasks are used to administer HTTP listeners:

- "To Create an HTTP Listener" on page 112
- "To List HTTP Listeners" on page 113
- "To Delete an HTTP Listener" on page 114
- "To Configure an HTTP Listener for SSL" on page 115
- "To Delete SSL From an HTTP Listener" on page 115

▼ To Create an HTTP Listener

The remote `create-http-listener` command enables you to create an HTTP listener.

1 Ensure that the server is running.

Remote commands require a running server.

2. **Create an HTTP listener by using the `create-http-listener`(1) command.**

3. **To apply your changes, restart Enterprise Server.**

   a. **Stop Enterprise Server.**
      For instructions, see "To Stop a Domain (or Server)" on page 35.

   b. **Start Enterprise Server.**
      For instructions, see "To Start a Domain (or Server)" on page 35.

Example 8–1 Creating an HTTP Listener

The following example command creates an HTTP listener named `sampleListener` that uses a non-default number of acceptor threads. Security is not enabled at runtime.

```
asadmin create-http-listener --listeneraddress 0.0.0.0
--listenerport 7272 --defaultvs server --servername host1.sun.com
--acceptorthreads 100 --securityenabled=false
--enabled=false sampleListener
```

Information similar to the following is displayed:

```
Command create-http-listener executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin help create-http-listener` at the command line.

▼ To List HTTP Listeners

The remote `list-http-listeners` command enables you to list the existing HTTP listeners.

1. **Ensure that the server is running.**
   Remote commands require a running server.

2. **List HTTP listeners by using the `list-http-listeners`(1) command.**

Example 8–2 Listing HTTP Listeners

The following example command lists the HTTP listeners on `localhost`:

```
asadmin list-http-listeners
```

Information similar to the following is displayed:

```
sampleListener
admin-listener
http-listener-2
http-listener-1
Command list-http-listeners executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin list-http-listeners -help` at the command line.

▼ To Delete an HTTP Listener

The remote `delete-http-listener` command enables you to delete an existing HTTP listener. This disables secure communications for the listener.

1. **Ensure that the server is running.**
   Remote commands require a running server.
2. **Obtain the exact name of the HTTP listener that you are deleting.**
   To list the existing HTTP listeners:
   ```
   asadmin list-http-listeners
   ```
3. **If necessary, notify users that the HTTP listener is being deleted.**
4. **Delete an HTTP listener by using the `delete-http-listener`(1) command.**
5. **To apply your changes, restart Enterprise Server.**
   a. **Stop Enterprise Server.**
      For instructions, see "To Stop a Domain (or Server)" on page 35.
   b. **Start Enterprise Server.**
      For instructions, see "To Start a Domain (or Server)" on page 35.

Example 8–3 Deleting an HTTP Listener

The following example command deletes the HTTP listener named `sampleListener` from `localhost`:

```
delete-http-listener sampleListener
```

Information similar to the following is displayed:

```
Command delete-http-listener executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin delete-http-listener -help` at the command line.

▼ To Configure an HTTP Listener for SSL

The remote `create-ssl` command enables you to create and configure an SSL element in the specified listener. This enables secure communication for the listener.

1. **Ensure that the server is running.**
   Remote commands require a running server.
2. **Configure an HTTP listener by using the `create-ssl(1)` command.**
3. **To apply your changes, restart Enterprise Server.**
   a. **Stop Enterprise Server.**
      For instructions, see "To Stop a Domain (or Server)" on page 35.
   b. **Start Enterprise Server.**
      For instructions, see "To Start a Domain (or Server)" on page 35.

Example 8–4 Configuring an HTTP Listener for SSL

The following example command enables the HTTP listener named `http-listener-1` for SSL:

```
asadmin create-ssl --type http-listener --certname sampleCert http-listener-1
```

Information similar to the following is displayed:

```
Command create-ssl executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin create-ssl --help` at the command line.

▼ To Delete SSL From an HTTP Listener

The remote `delete-ssl` command enables you to delete the SSL element in the specified listener. This disables secure communications for the listener.

1. **Ensure that the server is running.**
   Remote commands require a running server.

2. **Delete SSL from an HTTP listener by using the `delete-ssl(1)` command.**

3. **To apply your changes, restart Enterprise Server.**

   a. **Stop Enterprise Server.**
      For instructions, see "To Stop a Domain (or Server)" on page 35.

   b. **Start Enterprise Server.**
      For instructions, see "To Start a Domain (or Server)" on page 35.

Example 8–5 Deleting SSL From an HTTP Listener

The following example command disables SSL for the HTTP listener named `http-listener-1`:

```
asadmin delete-ssl --type http-listener http-listener-1
```

Information similar to the following is displayed:

```
Command delete-http-listener executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin delete-ssl --help` at the command line.

Administering Virtual Servers

A *virtual server*, sometimes called a virtual host, is an object that allows the same physical server to host multiple Internet domain names. All virtual servers hosted on the same physical server share the IP address of that physical server. A virtual server associates a domain name for a server (such as `www.aaa.com`) with the particular server on which Enterprise Server is running. Each virtual server must be registered with the DNS server for your network.

Note – Do not confuse an Internet domain with the administrative domain of Enterprise Server.

For example, assume that you want to host the following domains on your physical server: `www.aaa.com`, `www.bbb.com`, and `www.ccc.com`. Assume that these domains are respectively associated with web modules `web1`, `web2`, and `web3`. This means that the following URLs are handled by your physical server:

```
http://www.aaa.com:8080/web1
http://www.bbb.com:8080/web2
http://www.ccc.com:8080/web3
```

The first URL is mapped to virtual server `www.aaa.com`, the second URL is mapped to virtual server `www.bbb.com`, and the third is mapped to virtual server `www.ccc.com`. For this mapping to work, `www.aaa.com`, `www.bbb.com`, and `www.ccc.com` must all resolve to your physical server's IP address and each virtual server must be registered with the DNS server for your network.

By default, when Enterprise Server starts, the following virtual servers are started automatically:

- A virtual server named `server`, which hosts all user-defined web modules.

  For development, testing, and deployment of web services in a non-production environment, `server` is often the only virtual server required.
- A virtual server named `__asadmin`, which hosts all administration-related web modules (specifically, the Administration Console). This server is restricted, which means that you cannot deploy web modules to this virtual server.

The following tasks are used to administer virtual servers:

- "To Create a Virtual Server" on page 117
- "To List Virtual Servers" on page 118
- "To Delete a Virtual Server" on page 119

▼ To Create a Virtual Server

The remote `create-virtual-server` command enables you to create the named virtual server.

Before You Begin A virtual server must specify an existing HTTP listener. Because the virtual server cannot specify an HTTP listener that is already being used by another virtual server, create at least one HTTP listener before creating a new virtual server.

1. **Ensure that the server is running.**

   Remote commands require a running server.

2. **Create a virtual server by using the `create-virtual-server`(1) command.**

3. **To apply your changes, restart Enterprise Server.**

   a. **Stop Enterprise Server.**

      For instructions, see "To Stop a Domain (or Server)" on page 35.

   b. **Start Enterprise Server.**

      For instructions, see "To Start a Domain (or Server)" on page 35.

Example 8–6 Creating a Virtual Server

The following example command creates a virtual server named `sampleServer` for hosts `pigeon` and `localhost`:

```
asadmin create-virtual-server --hosts pigeon,localhost sampleServer
```

Information similar to the following is displayed:

```
Command create-virtual-server executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin create-virutal-server --help` at the command line.

▼ To List Virtual Servers

The remote `list-virtual-servers` command enables you to list the existing virtual servers.

1. **Ensure that the server is running.**
   Remote commands require a running server.
2. **List virtual servers by using the `list-virtual-servers(1)` command.**

Example 8–7 Listing Virtual Servers

The following example command lists the virtual servers for `localhost`:

```
asaadmin list-virtual-servers
```

Information similar to the following is displayed:

```
sampleListener
admin-listener
http-listener-2
http-listener-1

Command list-http-listeners executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin list-virutal-servers --help` at the command line.

▼ To Delete a Virtual Server

The remote `delete-virtual-server` command enables you to delete an existing virtual server.

1. **Ensure that the server is running.**
   Remote commands require a running server.
2. **Obtain the exact name of the virtual server that you are deleting.**
   To list the existing virtual servers:
   ```
   asadmin list-virtual-servers
   ```
3. **If necessary, notify users that the virtual server is being deleted.**
4. **Delete a virtual server by using the `delete-virtual-server`(1) command.**
5. **To apply your changes, restart Enterprise Server.**
   a. **Stop Enterprise Server.**
      For instructions, see "To Stop a Domain (or Server)" on page 35.
   b. **Start Enterprise Server.**
      For instructions, see "To Start a Domain (or Server)" on page 35.

Example 8–8 Deleting a Virtual Server

The following example command deletes the virtual server named `sampleServer` from `localhost`:

```
asadmin delete-virtual-server sampleServer
```

Information similar to the following is displayed:

```
Command delete-virtual-server executed successfully.
```

See Also To see the full syntax and options of the command, type `asadmin delete-virutal-server --help` at the command line.

◆ ◆ ◆ CHAPTER 9

Administering Logging

This chapter provides instructions on how to configure logging and how to view log information in the Sun GlassFish™ Enterprise Server v3 Prelude environment.

The following topics are addressed here:

About Logging

Logging is the process by which Enterprise Server captures data about events that occur during Enterprise Server operation. This data is recorded in a log file and is usually the first source of information when Enterprise Server problems occur.

Although application components can use the Apache Commons Logging Library to record messages, the platform standard JSR 047 API is recommended for better log configuration.

The following topics are addressed here:

Log File

Enterprise Server log records are captured in the `server.log` file, typically located in *domain-dir*`/logs`. You can change the default name or location of the log file by following instructions in "To Change the Log File Name or Directory" on page 128.

Enterprise Server log records follow a uniform format:

[#|*yyyy-mm-dd*T*hh*:*mm*:*ss*.*SSS-Z*|*Log Level*|*ProductName-Version*|*LoggerName*|*Key Value Pairs*|*Message*|#]

- [# and #] mark the beginning and end of the record.
- The vertical bar (|) separates the fields of the record.
- *yyyy-mm-ddThh:mm:ss.SSSS-Z* specifies the date and time that the record was created. For example: 2006-10-21T13:25:53.852-0400
- *Log Level* specifies the desired log level. You can select any of the following values: SEVERE, WARNING, INFO, CONFIG, FINE, FINER, and FINEST. The default is INFO.
- *ProductName-Version* refers to the current version of the Enterprise Server. For example: glassfish
- *LoggerName* is a hierarchical logger namespace that identifies the source of the log module. For example: javax.enterprise.system.core
- *Key Value Pairs* refers to pairs of key names and values, typically a thread ID. For example: _ThreadID=14;
- *Message* is the text of the log message. For all Enterprise Server SEVERE and WARNING messages and for many INFO messages, the message begins with a message ID that consists of a module code and a numerical value. For example: CORE5004

An example log record might look like this:

```
[#|2006-10-21T13:25:53.852-0400|INFO|GlassFish10.0|javax.enterprise.
system.core|_ThreadID=13;|CORE5004: Resource Deployed:
[cr:jms/DurableConnectionFactory].|#]
```

Configuration File

You can configure logging by editing the logging.properties file. The default logging.properties file is located in the same directory as the domain.xml file, typically *domain-dir*/config. You can choose a different file name by using the java.util.logging.config.file system property to specify a file name. For example:

```
java -Djava.util.logging.config.file=myfile
```

Logger Namespaces

A logger is provided for each Enterprise Server module. The following table lists the names of the modules and the namespace for each logger.

TABLE 9–1 Logger Namespaces for Enterprise Server Modules

Module Name	Namespace
Admin	`javax.enterprise.system.tools.admin`
Classloader	`javax.enterprise.system.core.classloading`
Configuration	`javax.enterprise.system.core.config`
Deployment	`javax.enterprise.system.tools.deployment`
Persistence	`oracle.toplink.essentials`, `javax.enterprise.resource.jdo`, `javax.enterprise.system.container.cmp`
Root	`javax.enterprise`
Security	`javax.enterprise.system.core.security`
Util	`javax.enterprise.system.util`
Verifier	`javax.enterprise.system.tools.verifier`

Web container	`javax.enterprise.system.container.web`
	`org.apache.catalina`
	`org.apache.coyote`
	`org.apache.jasper`

Configuring Logging

This section explains how to configure logging for Enterprise Server by setting properties in the `logging.properties` file. Log level configuration changes are dynamic and do not require that you restart the server, but other configuration changes do require you to restart the server.

The following topics are addressed here:

- "Configuring Log Handler Elements" on page 123
- "Setting Log Levels" on page 126
- "To Change the Log File Name or Directory" on page 128

Configuring Log Handler Elements

The following tasks apply to configuring log handler elements:

- "To Set Handler-Specific Properties" on page 124
- "To Add a Log Handler" on page 125

▼ To Set Handler-Specific Properties

You can modify logging behavior by setting properties in the `logging.properties` file as follows:

1. **In a text editor, find the property that you want to modify and make your changes.**

   The following properties are available:

   File Description
   : Contains the name and location of the log file. By default, the name is `server.log` and the location is the `logs` directory. These can be changed.

   ```
   com.sun.enterprise.server.logging.FileandSyslogHandler.file=logs/server.log
   ```

   File Rotation Time Limit
   : Rotates the log file based on time in minutes. If defined, time limit takes precedence over size limit. If set to 0 (the default), there is no rotation based on time.

   ```
   com.sun.enterprise.server.logging.FileandSyslogHandler.rotationTimelimitInMinutes=0
   ```

   File Rotation Size Limit
   : Rotates the log file based on limit in bytes. If set to 0 (the default), there is no rotation based on file size. 500000 is the minimum.

   ```
   com.sun.enterprise.server.logging.FileandSyslogHandler.rotationLimitInBytes=0
   ```

   Formatter
   : Uses `UniformLogFormatter` to format the records in the log file.

   ```
   com.sun.enterprise.server.logging.FileandSyslogHandler.formatter=com.
   sun.enterprise.server.logging.UniformLogFormatter
   ```

2. **Save the file.**

3. **To apply your changes, restart Enterprise Server.**

   a. **Stop Enterprise Server.**

      For instructions, see "To Stop a Domain (or Server)" on page 35.

   b. **Start Enterprise Server.**

      For instructions, see "To Start a Domain (or Server)" on page 35.

Example 9–1 Changing a Handler-Specific Logging Property

The following example changes the `rotationTimelimitInMinutes` from 0 (the default) to 90 minutes:

Before:

```
com.sun.enterprise.server.logging.FileandSyslogHandler.rotationTimelimitInMinutes=0
```

After: :

```
com.sun.enterprise.server.logging.FileandSyslogHandler.rotationTimelimitInMinutes=90
```

▼ To Add a Log Handler

A comma-separated list of log handlers is installed during startup of the Java™ Virtual Machine (JVM™) host. The default log handler that is provided in the `logging. properties` file, `ConsoleHandler`, is configured as follows:

```
handlers= java.util.logging.ConsoleHandler
```

In Enterprise Server, the best approach to developing a handler is to define a Hundred-Kilobyte Kernel (HK2) component which implements the handler contract. Enterprise Server registers this handler automatically because it is an HK2 component, and there is no task required of the administrator.

To implement a new handler that is not developed as an HK2 component, you need to add the new handler to the `logging.properties` file after the developer has put the new handler JAR file into the `/lib` directory.

1. **In a text editor, add the new file handler to the line that begins with `handlers=`.**
2. **Save the file.**
3. **To apply your changes, restart Enterprise Server.**
   a. **Stop Enterprise Server.**
      For instructions, see "To Stop a Domain (or Server)" on page 35.
   b. **Start Enterprise Server.**
      For instructions, see "To Start a Domain (or Server)" on page 35.

Example 9–2 Adding a New Log Handler

The following example adds the new logger `com.example.logging.MyHandler`:

Before:

```
handlers= java.util.logging.ConsoleHandler
```

After:

```
handlers= java.util.logging.ConsoleHandler, com.example.logging.MyHandler
```

Setting Log Levels

The *log level* describes the type of material that is contained in the message. The following values apply: `SEVERE`, `WARNING`, `INFO`, `CONFIG`, `FINE`, `FINER`, and `FINEST`. These log levels are hierarchically inclusive, which means that if you set a particular log level, such as INFO, the messages that have log levels above that level (`SEVERE` and `WARNING`) are also included. If you set the log level to the lowest level, `FINEST`, your output will include all the messages in the file. The default setting is `INFO`.

There are two levels of log settings available: global and logger-specific. If you have chosen a logger-specific setting that is different from the global setting, the logger-specific setting takes precedence.

Because setting log levels is a dynamic operation, you do not need to restart Enterprise Server for changes to take effect.

The following topics are addressed here:

- "To Set the Global Log Level" on page 126
- "To Set Logger-Specific Properties" on page 127

To Set the Global Log Level

The *global log level* specifies which kinds of events are logged across all loggers. The default level for message output to the console is `INFO` (which also includes `SEVERE` and `WARNING` messages).

The `ConsoleHandler` has a separate log level setting that limits the messages that are displayed. For example:

```
java.util.logging.ConsoleHandler.level = INFO
java.util.logging.ConsoleHandler.formatter =
com.sun.enterprise.server.logging.UniformLogFormatter
```

1. **In a text editor, find the `ConsoleHandler` log level line and make your changes.**
2. **Save the file.**

Example 9–3 Changing the Global Log Level for All Loggers

If you set the log level at the root level, you are setting the level of all loggers. The following example sets the log level for all loggers to `INFO`:

```
.level= INFO
```

To Set Logger-Specific Properties

Logger-specific properties provide extra control for the logger of each module. By default, these properties are set to log level `FINE` in the `logging. properties` file, however, the lines in the file are commented out. The lines for the loggers might look like this (the modules are indicated in bold):

```
#javax.enterprise.system.tools.level=FINE
#javax.enterprise.system.container.ejb.level=FINE
#javax.enterprise.system.core.security.level=FINE
#javax.enterprise.system.tools.admin.level=FINE
#javax.enterprise.level=FINE
#javax.enterprise.system.container.web.level=FINE
```

For any given module logger, such as the EJB logger, the global log level can be overridden by the module-specific setting.

1. **In a text editor, uncomment the line that applies to the logger that you want to modify.**
   Remove the #. The logger is activated using the default level INFO.

2. **Set the property to the desired log level.**
   Your choices are `SEVERE`, `WARNING`, `INFO`, `CONFIG`, `FINE`, `FINER`, and `FINEST`.

3. **Save the file.**
   The log levels are automatically updated.

Example 9–4 Changing the Log Level for the Console Handler

The following example changes the log level for `ConsoleHandler` from INFO to FINE

Before:

```
java.util.logging.ConsoleHandler.level = INFO
```

After:

```
java.util.logging.ConsoleHandler.level = FINE
```

Example 9–5 Setting a Log Level for a Specific Logger

The following example sets the log level for the EJB logger to SEVERE (overriding the global level INFO)

Before:

```
#javax.enterprise.system.container.ejb.level=FINE
```

After: :

```
javax.enterprise.system.container.ejb.level=SEVERE
```

▼ To Change the Log File Name or Directory

By default, the name of the log output file is `server.log` and it is located in *domain-dir*`/logs`. You can change the name or location of the file by editing the `logging.properties` file as follows:

1. **In a text editor, find the following line:**

   ```
   com.sun.enterprise.server.logging.FileandSyslogHandler.file=logs/server.log
   ```

   In this line, `logs` indicates the directory where the file is located, and `server.log` is the file name.

2. **Make your changes and save the file.**

3. **To apply your changes, restart Enterprise Server.**

   a. **Stop Enterprise Server.**

      For instructions, see "To Stop a Domain (or Server)" on page 35.

   b. **Start Enterprise Server.**

      For instructions, see "To Start a Domain (or Server)" on page 35.

Example 9–6 Changing the Name and Location of the Log File

The following example changes the log file name from `server.log` to `myLogging.log` and changes the location from the `logs` directory to the `dpin` directory:

Before:

```
com.sun.enterprise.server.logging.FileandSyslogHandler.file=logs/server.log
```

After:

```
com.sun.enterprise.server.logging.FileandSyslogHandler.file=dpin/myLogging.log
```

Viewing Log Information

By default, all log information is captured in the `server.log` file, typically located in *domain-dir*`/logs`. To view information that has been collected for a module, you can open the `server.log` file in a text editor and search for the module that you are interested in.

You can also use the command-line file viewing commands, such as `tail`, `grep`, or `more`.

◆ ◆ ◆ CHAPTER 10

Monitoring the Enterprise Server

This chapter explains how to monitor the Sun GlassFish™ Enterprise Server v3 Prelude components and services by using the `asadmin` command-line utility. Instructions for connecting JConsole are also provided.

The following topics are addressed here:

Instructions for monitoring by using the Administration Console are contained in the Administration Console online help.

About Monitoring

Monitoring is the process of reviewing the statistics of a system to improve performance or solve problems. By monitoring the state of various components and services deployed in the Enterprise Server, you can identify performance bottlenecks, predict failures, perform root cause analysis, and ensure that everything is functioning as expected. Data gathered by monitoring can also be useful in performance tuning and capacity planning.

The `asadmin` utility provides multiple ways of viewing monitoring data: by using the `monitor` command to view common data, and by using the `list` and `get` commands to view more comprehensive data.

The following topics are addressed here:

How the Monitoring Tree Structure Works

A *monitorable object* is a component, subcomponent, or service that can be monitored. Enterprise Server uses a tree structure to track monitorable objects. Because the tree is dynamic, the tree changes as components of the Enterprise Server instance are added or removed.

In the tree, a monitorable object can have child objects (nodes) that represent exactly what can be monitored for that object. All child objects are addressed using the dot (.) character as a separator. These constructed names are referred to as *dotted names*.

The following are monitorable objects for the instance `server`:

```
server.applications
server.http-service
server.jvm
server.web
```

Dotted names can also address specific attributes in monitorable objects. For example, the `jvm` type has a `memory` attribute called `maxheapsize-count`. The following dotted name addresses the attribute:

```
server.jvm.memory.maxheapsize-count
```

Although an object is monitorable, it is not necessarily being monitored. For instructions on activating monitoring, see "Configuring Monitoring" on page 134.

Tree Structure of Monitorable Objects

Each monitorable object has a hierarchical tree structure. In the tree, a replaceable such as **statistics* represents the name of the attribute that you want to show statistics for.

This section shows the node tree hierarchies of the following monitorable objects:

- "Applications Tree Hierarchy" on page 132
- "HTTP Service Tree Hierarchy" on page 133
- "JVM Tree Hierarchy" on page 133
- "Web Tree Hierarchy" on page 133

Applications Tree Hierarchy

The applications tree contains the following nodes:

```
server.applications
        application-name
            virtual-server
                  request
                     *statistic
```

An example dotted name might be `server.applications.hello.server.request.maxtime-count`. For available attributes, see "Applications Statistics" on page 142.

HTTP Service Tree Hierarchy

The `http-service` tree contains the following nodes:

```
server.http-service
       virtual-server
           request
               *statistic
       _asadmin
           request
               *statistic
```

An example dotted name might be `server.http-service.virtual-server1.request.requestcount-count`. For available attributes, see "HTTP Service Statistics" on page 143.

JVM Tree Hierarchy

The `jvm` tree contains the following nodes:

```
server.jvm
           class-loading-system
           compilation-system
           garbage-collectors
           memory
           operating-system
           runtime
```

An example dotted name might be `server.jvm.memory.maxheapsize-count`. For available attributes, see "Java Virtual Machine (JVM) Statistics" on page 146.

Web Tree Hierarchy

The `web` tree contains the following nodes:

```
server.web
           jsp
              *statistic
           servlet
              *statistic
           session
              *statistic
           request
              *statistic
```

An example dotted name might be `server.web.servlet.activeservletsloadedcount-count`. For available attributes, see "Web Module Common Statistics" on page 137.

Tools for Monitoring Enterprise Server

The following `asadmin` commands are provided for monitoring the services and components of Enterprise Server:

- The `set` command is used to set the monitoring levels for monitorable objects, essentially turning monitoring on or off for Enterprise Server components and services. For instructions, see "Configuring Monitoring" on page 134.
- The `monitor --type` command is used to display basic data for a particular type of monitorable object. For instructions, see "Viewing Common Monitoring Data" on page 136.
- The `list --monitor` command is used to display available monitorable objects. For guidelines and instructions, see "Guidelines for Using the `list` and `get` Commands for Monitoring" on page 138.
- The `get` command is used to display comprehensive data, such as the attributes and values for a dotted name. The `get` command used with a wildcard parameter displays all available attributes for any monitorable object. For instructions on viewing comprehensive data, see "To View Comprehensive Monitoring Data" on page 139.

Configuring Monitoring

By default, monitoring is *not* enabled for Enterprise Server components and services. When you set the monitoring level, you are enabling monitoring for that monitorable object. You can choose to leave monitoring off for objects that do not need to be monitored.

The following tasks are addressed here:

▼ To Enable Monitoring

The `set(1)` command enables monitoring for Enterprise Server components and services. By setting the monitoring level to HIGH or LOW, you start monitoring for that service or component. For the Prelude release of Enterprise Server, there is no difference between the HIGH and LOW settings.

1 Determine which services and components are currently enabled for monitoring.

```
asadmin get server.monitoring-service.module-monitoring-levels.*
```

The following example output shows that monitoring for the HTTP service is not enabled (OFF), but other objects are enabled:

```
configs.config.server-config.monitoring-service.module-monitoring-levels.web-container=HIGH
       configs.config.server-config.monitoring-service.module-monitoring-levels.http-service=OFF
           configs.config.server-config.monitoring-service.module-monitoring-levels.jvm=HIGH
```

2. **Enable monitoring for each service or component that you want to monitor.**

   The following example command enables monitoring for the HTTP service by setting the monitoring level to HIGH:

   ```
   asadmin set server.monitoring-service.module-monitoring-levels.http-service=HIGH
   ```

See Also To see the full syntax and options of the command, type `asadmin set --help` at the command line.

▼ To Disable Monitoring

The `set(1)` command enables you to disable monitoring for Enterprise Server services and components.

1. **Determine which services and components currently are enabled for monitoring.**

   ```
   asadmin get server.monitoring-service.module-monitoring-levels.*
   ```

   The following example output shows that monitoring is enabled for `web-container`, `http-service`, and `jvm`:

```
configs.config.server-config.monitoring-service.module-monitoring-levels.web-container=HIGH
       configs.config.server-config.monitoring-service.module-monitoring-levels.http-service=HIGH
              configs.config.server-config.monitoring-service.module-monitoring-levels.jvm=HIGH
```

2. **Disable monitoring for a service or component that you do not want to monitor.**

   The following example command disables monitoring for the HTTP service by setting the monitoring level to OFF:

   ```
   asadmin set server.monitoring-service.module-monitoring-levels.http-service=OFF
   ```

See Also To see the full syntax and options of the command, type `asadmin set --help` at the command line.

Viewing Common Monitoring Data

The `asadmin monitor` command enables you to display basic data on commonly-monitored objects.

▼ To View Common Monitoring Data

Use the `--type` option of the `monitor` command to specify the object for which you want to display data. If you use the `monitor` command without specifying a type, an error message is displayed.

Output from the command is displayed continuously in a tabular format; the `--interval` option can be used to display output at a particular interval (the default is 30 seconds).

Before You Begin A monitorable object must be configured for monitoring before you can display data on the object. See "To Enable Monitoring " on page 134.

1. **Determine which type of monitorable object you want to monitor.**

   Your choices for v3 Prelude are `jvm`, `httplistener`, and `webmodule`.

2. **Request the monitoring data by using the `monitor`(1) command.**

Example 10–1 Viewing Common Monitoring Data

The following example command requests common data for type `jvm` on instance `server`:

```
asadmin monitor --type jvm server
```

Information similar to the following is displayed:

```
UpTime(ms)                          Heap and NonHeap Memory(bytes)
current                  min         max         low         high        count

9437266                  8585216     619642880   0           0           93093888
9467250                  8585216     619642880   0           0           93093888
```

See Also To see the full syntax and options of the command, type `asadmin monitor --help` at the command line.

Common Monitoring Statistics

Common monitoring statistics are described in the following sections:

HTTP Listener Common Statistics

The statistics available for the `httplistener` type are shown in the following table.

TABLE 10–1 HTTP Listener Common Monitoring Statistics

Statistic	Description
`ec`	Error count. Cumulative value of the error count
`mt`	Maximum time. Longest response time for a request; not a cumulative value, but the largest response time from among the response times
`pt`	Processing time. Cumulative value of the times taken to process each request, with processing time being the average of request processing times over request
`rc`	Request count. Cumulative number of requests processed so far

JVM Common Statistics

The statistics available for the `jvm` type are shown in the following table.

TABLE 10–2 JVM Common Monitoring Statistics

Statistic	Description
`count`	Amount of memory (in bytes) that is guaranteed to be available for use by the JVM machine
`high`	Retained for compatibility with other releases
`low`	Retained for compatibility with other releases
`min`	Initial amount of memory (in bytes) that the JVM machine requests from the operating system for memory management during startup
`UpTime`	Number of milliseconds that the JVM machine has been running since it was last started

Web Module Common Statistics

The statistics available for the `webmodule` type are shown in the following table.

TABLE 10–3 Web Module Common Monitoring Statistics

Statistic	Description
`ajlc`	Number of active JavaServer Pages™ (JSP™) technology pages that are loaded
`asc`	Current active sessions
`aslc`	Number of active servlets that are loaded
`ast`	Total active sessions
`mjlc`	Maximum number of JSP pages that are loaded
`mslc`	Maximum number of servlets that are loaded
`rst`	Total rejected sessions
`st`	Total sessions
`tjlc`	Total number of JSP pages that are loaded
`tslc`	Total number of servlets that are loaded

Viewing Comprehensive Monitoring Data

By applying the `list` and `get` commands against the tree structure using dotted names, you can display more comprehensive monitoring data.

The following topics are addressed here:

Guidelines for Using the `list` and `get` Commands for Monitoring

The underlying assumptions for using the `list` and `get` commands with dotted names are:

- Any `list` command that specifies a dotted name followed by a wildcard of the form `.*` lists a hierarchical tree of child nodes from the specified node. For example, the following command lists all children of the `applications` node, their subsequent child nodes, and so on:

  ```
  list --monitor=true server.applications.*
  ```

- Any `list` command that specifies a dotted name preceded or followed by a wildcard of the form **dottedname* or *dotted*name* or *dottedname** lists all nodes and their children that match the regular expression created by the specified matching pattern.

- A `get` command followed by a `.*` or a `*` gets the set of attributes and their values that belong to the node specified.

The following tables show the command, dotted name, and corresponding output at the top level (server) of the tree.

TABLE 10–4 Server-Level Monitoring Commands Output

Command	Dotted Name	Output
`list --monitor=true`	`server.*`	Hierarchy of child nodes below server node
`get --monitor=true`	`server.*`	Data for all monitorable objects under the top-level server

The following table shows the command, dotted name, and corresponding output for the JVM level.

TABLE 10–5 JVM-Level Monitoring Commands Output

Command	Dotted Name	Output
`list --monitor=true`	`server.jvm.*`	Monitorable objects under the JVM node
`get --monitor=true`	`server.jvm.*`	Attributes and values corresponding to JVM attributes

The following table shows the command, dotted name, and corresponding output for the web level.

TABLE 10–6 Web Container-Level Monitoring Commands Output

Command	Dotted Name	Output
`list --monitor=true`	`server.web.*`	Monitorable objects under the web service node
`get --monitor=true`	`server.web.*`	Attributes and values that correspond to web attributes

▼ To View Comprehensive Monitoring Data

The `monitor` command is useful in many situations, however, it does not offer the complete list of all monitorable objects. To work with comprehensive data for an object type, use the `list --monitor` and the `get --monitor` commands followed by the dotted name of a monitorable object.

Before You Begin A monitorable object must be configured for monitoring before you can display information on the object. See "To Enable Monitoring " on page 134 if needed..

1. **List the objects that are enabled for monitoring by using the `list(1)` command.**

   For example, the following command lists all components and services that have monitoring enabled for instance `server`:

   ```
   asadmin list --monitor=true server.*
   ```

   Information similar to the following is displayed:

   ```
   server.web
   server.jvm
   server.http-service
   server.applications
   ```

2. **Get data for a monitored component or service by using the `get(1)` command.**

Example 10–2 Viewing Attributes for a Specific Type

The following example command gets information on all the attributes for object type `jvm` on instance `server`:

```
asadmin get --monitor=true server.jvm.*
```

Information similar to the following is displayed:

```
server.jvm.class-loading-system.loadedclasscount-count = 3715
server.jvm.class-loading-system.totalloadedclasscount-count = 3731
server.jvm.class-loading-system.unloadedclasscount-count = 16
server.jvm.compilation-system.name-current = HotSpot Client Compiler
server.jvm.compilation-system.totalcompilationtime-count = 769
server.jvm.garbage-collectors.Copy.collectioncount-count = 285
server.jvm.garbage-collectors.Copy.collectiontime-count = 980
server.jvm.garbage-collectors.MarkSweepCompact.collectioncount-count = 2
server.jvm.garbage-collectors.MarkSweepCompact.collectiontime-count = 383
server.jvm.memory.committedheapsize-count = 23498752
server.jvm.memory.committednonheapsize-count = 13598720
server.jvm.memory.initheapsize-count = 0
server.jvm.memory.initnonheapsize-count = 8585216
server.jvm.memory.maxheapsize-count = 66650112
server.jvm.memory.maxnonheapsize-count = 100663296
server.jvm.memory.objectpendingfinalizationcount-count = 0
server.jvm.memory.usedheapsize-count = 19741184
server.jvm.memory.usednonheapsize-count = 13398352
server.jvm.operating-system.arch-current = x86
server.jvm.operating-system.availableprocessors-count = 2
server.jvm.operating-system.name-current = Windows XP
server.jvm.operating-system.version-current = 5.1
server.jvm.runtime.classpath-current = glassfish.jar
server.jvm.runtime.inputarguments-current = []
```

```
server.jvm.runtime.managementspecversion-current = 1.0
server.jvm.runtime.name-current = 4372@ABBAGANI_WORK
server.jvm.runtime.specname-current = Java Virtual Machine Specification
server.jvm.runtime.specvendor-current = Sun Microsystems Inc.
server.jvm.runtime.specversion-current = 1.0
server.jvm.runtime.uptime-count = 84813
server.jvm.runtime.vmname-current = Java HotSpot(TM) Client VM
server.jvm.runtime.vmvendor-current = Sun Microsystems Inc.
server.jvm.runtime.vmversion-current = 1.5.0_11-b03
```

Example 10–3 Viewing Monitorable Applications

The following example command lists all the monitorable applications for instance `server`:

```
asadmin list --monitor=true server.applications.*
```

Information similar to the following is displayed:

```
server.applications.app1
server.applications.app2
server.applications.app1.virtual-server1
server.applications.app2.virtual-server1
```

Example 10–4 Viewing Attributes From an Application

The following example command gets information on all the attributes for application `hello`:

```
asadmin get --monitor=true server.applications.hello.*
```

Information similar to the following is displayed:

```
server.applications.hello.server.activatedsessionstotal-count = 0
server.applications.hello.server.activejspsloadedcount-count = 1
server.applications.hello.server.activeservletsloadedcount-count = 1
server.applications.hello.server.activesessionscurrent-count = 1
server.applications.hello.server.activesessionshigh-count = 1
server.applications.hello.server.errorcount-count = 0
server.applications.hello.server.expiredsessionstotal-count = 0
server.applications.hello.server.maxjspsloadedcount-count = 1
server.applications.hello.server.maxservletsloadedcount-count = 0
server.applications.hello.server.maxtime-count = 0
server.applications.hello.server.passivatedsessionstotal-count = 0
server.applications.hello.server.persistedsessionstotal-count = 0
server.applications.hello.server.processingtime-count = 0.0
server.applications.hello.server.rejectedsessionstotal-count = 0
server.applications.hello.server.requestcount-count = 0
server.applications.hello.server.sessionstotal-count =
```

```
server.applications.hello.server.totaljspsloadedcount-count = 0
server.applications.hello.server.totalservletsloadedcount-count = 0
```

Example 10–5 Viewing a Specific Attribute

The following example command gets information on `jvm` attribute `runtime.vmversion-current` on an instance `server`:

```
asadmin get --monitor=true server.jvm.runtime.vmversion-current
```

Information similar to the following is displayed:

```
server.jvm.runtime.vmversion-current = 10.0-b23
```

Comprehensive Monitoring Statistics

You can get comprehensive monitoring statistics by forming a dotted name that specifies the statistic you are looking for. For example, the following dotted name will get the cumulative number of requests for the HTTP service on `virutal-server1`:

```
server.http-service.virtual-server1.request.requestcount-count
```

The tables in the following sections list the statistics that are available for each monitorable object:

- "Applications Statistics" on page 142
- "HTTP Service Statistics" on page 143
- "Java Virtual Machine (JVM) Statistics" on page 146
- "Web Statistics" on page 149

Applications Statistics

Applications fits into the tree of objects as shown in "Applications Tree Hierarchy" on page 132. Use the following dotted name pattern to get applications statistics:

`server.applications.`*application-name*`.`*virtual-server*`.request.`*statistic*

Statistics available for applications for are shown in the following table.

TABLE 10–7 Applications Comprehensive Statistics

Statistic	Description
`activatedsessionstotal-count`	Total number of activated sessions

TABLE 10–7 Applications Comprehensive Statistics *(Continued)*

Statistic	Description
activejspsloadedcount-count	Number of currently loaded JSPs
activeservletsloadedcount-count	Number of currently loaded servlets
activesessionscurrent-count	Number of currently active sessions
activesessionshigh-count	Maximum number of concurrently active sessions
errorcount-count	Cumulative value of the error count, with error count representing the number of cases where the response code was greater than or equal to 400
expiredsessionstotal-count	Total number of expired sessions
maxjspsloadedcount-count	Maximum number of JSPs loaded which were active
maxservletsloadedcount-count	Maximum number of servlets loaded which were active
maxtime-count	Longest response time for a request; not a cumulative value, but the largest response time from among the response times
passivatedsessionstotal-count	Total number of passivated sessions
persistedsessionstotal-count	Total number of persisted sessions
processingtime-count	Cumulative value of the times taken to process each request, with processing time representing the average of request processing times over the request count
rejectedsessionstotal-count	Total number of rejected sessions
requestcount-count	Cumulative number of the requests processed so far
sessionstotal-count	Total number of sessions created
totaljspsloadedcount-count	Cumulative number of JSP pages that have been loaded into the web module
totalservletsloadedcount-count	Cumulative number of servlets that have been loaded into the web module

HTTP Service Statistics

The HTTP service fits into the tree of objects as shown in "HTTP Service Tree Hierarchy" on page 133.

The HTTP service statistics are described in the following sections:

- "HTTP Service asadmin Statistics" on page 144
- "HTTP Service Virtual Server Statistics" on page 145

HTTP Service asadmin Statistics

Use the following dotted name pattern for the HTTP service `asadmin` statistics:

`server.http-service._asadmin.request.`*statistic*

The `asadmin` statistics available for HTTP service for are shown in the following table.

TABLE 10–8 HTTP Service `asadmin` Statistics

Statistic	Description
`count200-count`	Number of responses with a status code equal to 200
`count2xx-count`	Number of responses with a status code in the 2xx range
`count302-count`	Number of responses with a status code equal to 302
`count304-count`	Number of responses with a status code equal to 304
`count3xx-count`	Number of responses with a status code equal in the 3xx range
`count400-count`	Number of responses with a status code equal to 400
`count401-count`	Number of responses with a status code equal to 401
`count403-count`	Number of responses with a status code equal to 403
`count4xx-count`	Number of responses with a status code equal in the 4xx range
`count503-count`	Number of responses with a status code equal to 503
`count5xx-count`	Number of responses with a status code equal in the 5xx range
`countother-count`	Number of responses with a status code outside the 2xx, 3xx, 4xx, and 5xx range
`errorcount-count`	Cumulative value of the error count, with error count representing the number of cases where the response code was greater than or equal to 400
`maxtime-count`	Longest response time for a request; not a cumulative value, but the largest response time from among the response times
`processingtime-count`	Cumulative value of the times taken to process each request, with processing time being the average of request processing times over the request count
`requestcount-count`	Cumulative number of requests processed so far

HTTP Service Virtual Server Statistics

Use the following dotted name pattern for HTTP service virtual server statistics:

`server.http-service.`*virtual-server*`.request.`*statistic*

The HTTP service statistics for virtual servers are shown in the following table.

TABLE 10–9 HTTP Service Virtual Server Statistics

Statistic	Description
`count200-count`	Number of responses with a status code equal to 200
`count2xx-count`	Number of responses with a status code in the 2xx range
`count302-count`	Number of responses with a status code equal to 302
`count304-count`	Number of responses with a status code equal to 304
`count3xx-count`	Number of responses with a status code equal in the 3xx range
`count400-count`	Number of responses with a status code equal to 400
`count401-count`	Number of responses with a status code equal to 401
`count403-count`	Number of responses with a status code equal to 403
`count404-count`	Number of responses with a status code equal to 404
`count4xx-count`	Number of responses with a status code equal in the 4xx range
`count503-count`	Number of responses with a status code equal to 503
`count5xx-count`	Number of responses with a status code equal in the 5xx range
`countother-count`	Number of responses with a status code outside the 2xx, 3xx, 4xx, and 5xx range
`errorcount-count`	Cumulative value of the error count, with error count representing the number of cases where the response code was greater than or equal to 400
`maxtime-count`	Longest response time for a request; not a cumulative value, but the largest response time from among the response times
`processingtime-count`	Cumulative value of the times taken to process each request, with processing time being the average of request processing times over the request count
`requestcount-count`	Cumulative number of requests processed so far

Java Virtual Machine (JVM) Statistics

The statistics that are available for the JVM machine are shown in the following sections:

JVM Class Loading System Statistics

Use the following dotted name pattern for JVM class loading system statistics:

`server.jvm.class-loading-system.`*statistic*

The statistics that are available for class loading in the JVM for Java SE are shown in the following table.

TABLE 10–10 JVM Statistics for Java SE Class Loading

Statistic	Description
`loadedclasscount-count`	Number of classes that are currently loaded in the JVM
`totalloadedclasscount-count`	Total number of classes that have been loaded since the JVM began execution
`unloadedclasscount-count`	Number of classes that have been unloaded from the JVM since the JVM began execution

JVM Compilation System Statistics

Use the following dotted name pattern for JVM compilation system statistics:

`server.jvm.compilation-system.`*statistic*

The statistics that are available for compilation in the JVM for Java SE are shown in the following table.

TABLE 10–11 JVM Statistics for Java SE Compilation

Statistic	Description
`name-current`	Name of the current compiler
`totalcompilationtime-count`	Accumulated time (in milliseconds) spent in compilation

JVM Garbage Collectors Statistics

Use the following dotted name pattern for JVM garbage collectors statistics:

`server.jvm.garbage-collectors.`*statistic*

The statistics that are available for garbage collection in the JVM for Java SE are shown in the following table.

TABLE 10–12 JVM Statistics for Java SE Garbage Collectors

Statistic	Description
`collectioncount-count`	Total number of collections that have occurred
`collectiontime-count`	Accumulated collection time (in milliseconds)

JVM Memory Statistics

Use the following dotted name pattern for JVM memory statistics:

`server.jvm.memory.`*statistic*

The statistics that are available for memory in the JVM for Java SE are shown in the following table.

TABLE 10–13 JVM Statistics for Java SE Memory

Statistic	Description
`committedheapsize-count`	Amount of heap memory (in bytes) that is committed for the JVM to use
`committednonheapsize-count`	Amount of non-heap memory (in bytes) that is committed for the JVM to use
`initheapsize-count`	Size of the heap initially requested by the JVM
`initnonheapsize-count`	Size of the non-heap area initially requested by the JVM
`maxheapsize-count`	Maximum amount of heap memory (in bytes) that can be used for memory management
`maxnonheapsize-count`	Maximum amount of non-heap memory (in bytes) that can be used for memory management
`objectpendingfinalizationcount-count`	Approximate number of objects that are pending finalization
`usedheapsize-count`	Size of the heap currently in use

TABLE 10–13 JVM Statistics for Java SE Memory *(Continued)*

Statistic	Description
`usednonheapsize-count`	Size of the non-heap area currently in use

JVM Operating System Statistics

Use the following dotted name pattern for JVM operating system statistics:

`server.jvm.operating-system.`*statistic*

The statistics that are available for the operating system for the JVM machine in Java SE are shown in the following table.

TABLE 10–14 JVM Statistics for Java SE Operating System

Statistic	Description
`arch-current`	Operating system architecture
`availableprocessors-count`	Number of processors available to the JVM
`name-current`	Operating system name
`version-current`	Operating system version

JVM Runtime Statistics

Use the following dotted name pattern for JVM runtime statistics:

`server.jvm.runtime.`*statistic*

The statistics that are available for the runtime in the JVM runtime for Java SE are shown in the following table.

TABLE 10–15 JVM Statistics for Java SE Runtime

Statistic	Description
`classpath-current`	Classpath that is used by the system class loader to search for class files
`inputarguments-current`	Input arguments passed to the JVM; not including arguments to the `main` method
`managementspecversion-current`	Management specification version implemented by the JVM
`name-current`	Name representing the running JVM
`specname-current`	JVM specification name

TABLE 10–15 JVM Statistics for Java SE Runtime *(Continued)*

Statistic	Description
`specvendor-current`	JVM specification vendor
`specversion-current`	JVM specification version
`uptime-count`	Uptime of the JVM (in milliseconds)
`vmname-current`	JVM implementation name
`vmvendor-current`	JVM implementation vendor
`vmversion-current`	JVM implementation version

Web Statistics

The web container fits into the tree of objects as shown in "Web Tree Hierarchy" on page 133.

The available web statistics shown in the following sections:

- "Web JSP Statistics" on page 149
- "Web Request Statistics" on page 149
- "Web Servlet Statistics" on page 150
- "Web Session Statistics" on page 150

Web JSP Statistics

Use the following dotted name pattern for web JSP statistics:

`server.web.jsp.`*statistic*

The available web JSP statistics are shown in the following table.

TABLE 10–16 Web JSP Statistics

Statistic	Description
`activejspsloadedcount-count`	Number of currently loaded JSPs
`maxjspsloadedcount-count`	Maximum number of JSPs loaded which were active
`totaljspsloadedcount-count`	Cumulative number of JSP pages that have been loaded into the web module

Web Request Statistics

Use the following dotted name pattern for web request statistics:

`server.web.request.`*statistic*

The available web request statistics are shown in the following table.

TABLE 10–17 Web Request Statistics

Statistic	Description
`errorcount-count`	Cumulative value of the error count, with error count representing the number of cases where the response code was greater than or equal to 400
`maxtime-count`	Longest response time for a request; not a cumulative value, but the largest response time from among the response times
`processingtime-count`	Cumulative value of the times taken to process each request, with processing time being the average of request processing times over the request count
`requestcount-count`	Cumulative number of the requests processed so far

Web Servlet Statistics

Use the following dotted name pattern for web servlet statistics:

`server.web.servlet.`*statistic*

The available web servlet statistics are shown in the following table.

TABLE 10–18 Web Servlet Statistics

Statistic	Description
`activeservletsloadedcount-count`	Number of currently loaded servlets
`maxservletsloadedcount-count`	Maximum number of servlets loaded which were active
`totalservletsloadedcount-count`	Cumulative number of servlets that have been loaded into the web module

Web Session Statistics

Use the following dotted name pattern for web session statistics:

`server.web.session.`*statistic*

The available web session statistics are shown in the following table.

TABLE 10–19 Web Session Statistics

Statistic	Description
`activatedsessionstotal-count`	Total number of activated sessions

TABLE 10–19 Web Session Statistics *(Continued)*

Statistic	Description
`activesessionscurrent-count`	Number of currently active sessions
`activesessionshigh-count`	Maximum number of concurrently active sessions
`expiredsessionstotal-count`	Total number of expired sessions
`passivatedsessionstotal-count`	Total number of passivated sessions
`persistedsessionstotal-count`	Total number of persisted sessions
`rejectedsessionstotal-count`	Total number of rejected sessions
`sessionstotal-count`	Total number of sessions created

Connecting JConsole to Enterprise Server

Java SE 5 enhances management and monitoring of the virtual machine by including a Platform MBean Server and by including managed beans (MBeans) to configure the virtual machine.

To view all MBeans, Enterprise Server provides a configuration of the standard JMX connector server called System JMX Connector Server. As part of Enterprise Server startup, an instance of this JMX Connector Server is started. Any compliant JMX Connector Client can connect to the server using the JMX Connector Server.

Java SE also provides tools to connect to an MBean Server and view the MBeans registered with the server. JConsole is one such popular JMX Connector Client and is available as part of the standard Java SE distribution. By default, Enterprise Server is configured with a non-secure System JMX Connector Server and cannot be enabled for v3 Prelude. If this is an issue, `jms-connector` can be removed.

When you configure JConsole for use with Enterprise Server, Enterprise Server becomes the JMX Connector's server end and JConsole becomes the JMX Connector's client end.

▼ To Set Up JConsole Connectivity

This procedure describes how to connect JConsole to Enterprise Server. Security is not enabled on the JMX Connector.

1. **Start the domain.**

   For instructions, see "To Start a Domain (or Server)" on page 35.

2. **Start JConsole using this format:** *JDK_HOME*`/bin/jconsole`

   For example:

   ```
   /usr/java/bin/jconsole
   ```

The JConsole Connect to Agent window is displayed.

3 **Click the Remote tab and type the host name and port.**

4 **Click Connect.**

5 **In the Remote Process text box, specify the JMX Service URL.**

For example:

```
service:jmx:rmi:///jndi/rmi://localhost:8686/jmxrmi
```

Note – Another host name can be substituted for `localhost`. The default port number (8686) could change if the `jmx-connector` configuration has been modified.

---

6 **Click Connect.**

In the JConsole window you will see all your MBeans, JVM information, and so on, in various tabs.

7 **Browse Sun GlassFish v3 Prelude MBeans along with Platform MBeans.**

Use either the Remote tab or the Advanced tab to connect to the Enterprise Server.

Remote tab	Specify the user name, password, administration server host, and JMS port number (8686 by default), and then select Connect.
Advanced tab	Specify the `JMXServiceURL` as service: `jmx:rmi:///jndi/rmi://host:jms-port/jmxrmi`, and then select Connect. The `JMXServerURL` is printed in the `server.log` file and is displayed in the command window of the domain creation command.

See Also For more information on JConsole, see `http://java.sun.com/javase/6/docs/technotes/guides/management/jconsole.html`.

◆ ◆ ◆ APPENDIX A

The asadmin Utility Commands

This appendix lists the asadmin commands that are included with this release of the Sun GlassFish™ Enterprise Server v3 Prelude software.

Online help for the asadmin commands can be invoked on the command line, for example, asadmin create-domain --help. The *Sun GlassFish Enterprise Server v3 Prelude Reference Manual* also provides a collection of these help pages.

For general information on the asadmin utility, see "Command-Line Utility for Administration (asadmin)" on page 25 or asadmin(1M).

Note – The common options used with remote commands are described in the asadmin(1M) help page.

Basic Administration Commands

create-domain(1)	Creates the configuration of a domain. A domain can exist independent of other domains. Any user who has access to the asadmin utility on a given host can create a domain and store its configuration in a location of choice. For procedural information in this guide, see "To Create a Domain" on page 32.

`create-system-properties`(1)	The `create-system-properties` command creates or updates system properties. Supported in remote mode only. For procedural information in this guide, see "To Create System Properties" on page 39.
`delete-domain`(1)	Deletes the specified domain. The domain must be stopped before it can be deleted. For procedural information in this guide, see "To Delete a Domain" on page 34.
`delete-system-property`(1)	Deletes system properties of a domain or configuration. Supported in remote mode only. For procedural information in this guide, see "To Delete a System Property" on page 41.
`get`(1)	Gets an attribute of an element in the `domain.xml` file. With the `-m` option, gets the names and values of the monitorable or configurable attributes. For procedural information in this guide, see "Guidelines for Using the `list` and `get` Commands for Monitoring" on page 138.
`list`(1)	Lists the configurable element. On Solaris, quotes are needed when running commands with * as the option value or operand. For procedural information in this guide, see "Guidelines for Using the `list` and `get` Commands for Monitoring" on page 138.
`list-commands`(1)	Lists all the `asadmin` commands, local commands first, then remote commands. You can specify that only remote commands or only local commands be displayed. Supported in remote mode only. For procedural information in this guide, see "To List Commands" on page 42.
`list-containers`(1)	Lists application containers and the status of each container. Supported in remote mode only. For procedural information in this guide, see "To List Containers" on page 43.
`list-domains`(1)	Lists the existing domains and their statuses. If the domain directory is not specified, the domains in the default *as-install*`/domains` directory is displayed. For procedural information in this guide, see "To List Domains" on page 33.
`list-modules`(1)	Lists modules that are accessible to the Enterprise Server subsystem. The status of each module is included.

	Supported in remote mode only. For procedural information in this guide, see "To List Modules" on page 44.
`list-system-properties(1)`	Lists the system properties of a domain or configuration. Supported in remote mode only. For procedural information in this guide, see "To List System Properties" on page 40.
`login(1)`	Allows you to log in to the domain. For procedural information in this guide, see "To Log In to a Domain (or Server)" on page 36
`monitor(1)`	Displays monitoring information for the common Enterprise Server resources. Supported in remote mode only. For procedural information in this guide, see "To View Common Monitoring Data" on page 136.
`set(1)`	Sets the values of one or more configurable attributes. For procedural information in this guide, see "Configuring Monitoring" on page 134.
`start-domain(1)`	Starts a domain. If the domain directory is not specified, the default `domain1` in the default *as-install*`/domains` directory is started. If there are two or more domains, the *domain_name* operand must be specified. For procedural information in this guide, see "To Start a Domain (or Server)" on page 35.
`start-database(1)`	Starts the Java DB server. Use this command only for working with applications deployed to the Enterprise Server. For procedural information in this guide, see "To Start the Database" on page 64.
`stop-database(1)`	Stops a process of the Java DB database server. For procedural information in this guide, see "To Stop the Database" on page 65.
`stop-domain(1)`	Stops the domain administration server (DAS) of the specified domain. Supported in remote mode only. For procedural information in this guide, see "To Stop a Domain (or Server)" on page 35.
`uptime(1)`	Displays the length of time that the domain administration server (DAS) has been running since the last restart. Supported in remote mode only. For procedural information in this guide, see "To Display Domain Uptime" on page 38.

`version(1)`	Displays the version information for the option specified in archive or folder format. If the command cannot communicate with the administration server with the given user/password and host/port, then the command will retrieve the version locally and display a warning message. Supported in remote mode only. For procedural information in this guide, see "To Display the Enterprise Server Version" on page 46.

Deployment Commands

`deploy(1)`	Deploys an enterprise application, web application, EJB module, connector module, or application client module. If the component is already deployed or already exists, you can forcefully redeploy if you set the `--force` option to `true`. Supported in remote mode only.
`deploydir`	This command is deprecated.
`disable(1)`	Immediately disables the named component. If the component has not been deployed, an error message is returned. Supported in remote mode only.
`enable(1)`	Enables the specified component. If the component has not been deployed, an error message is returned. If the component is already enabled, then it is re-enabled. Supported in remote mode only.
`list-applications(1)`	Lists deployed Java™ EE applications. If the `--type` option is not specified, all applications are listed. Supported in remote mode only.
`list-components(1)`	Lists all deployed Java EE 5 components. If the `--type` option is not specified, all components are listed. Supported in remote mode only.
`redeploy(1)`	Redeploys an application that is already deployed. Supported in remote mode only.
`undeploy(1)`	Removes the specified deployed application. Supported in remote mode only.

HTTP Service Commands

`create-http-listener(1)`	Creates a new HTTP listener socket. Restart the server for the change to take effect. Supported in remote mode only. For procedural information in this guide, see "To Create an HTTP Listener" on page 112.
`create-virtual-server(1)`	Creates the specified virtual server element. Restart the server for the change to take effect. Supported in remote mode only. For procedural information in this guide, see "To Create a Virtual Server" on page 117.
`create-ssl(1)`	Creates and configures the SSL element in the selected HTTP listener to enable secure communication on that listener/service. Restart the server for the change to take effect. Supported in remote mode only. For procedural information in this guide, see "To Configure an HTTP Listener for SSL" on page 115.
`delete-http-listener(1)`	Deletes the specified HTTP listener. Restart the server for the change to take effect. Supported in remote mode only. For procedural information in this guide, see "To Delete an HTTP Listener" on page 114.
`delete-ssl(1)`	Deletes the SSL element in the selected HTTP listener. Restart the server for the change to take effect. Supported in remote mode only. For procedural information in this guide, see "To Delete SSL From an HTTP Listener" on page 115.
`delete-virtual-server(1)`	Deletes the specified virtual server element. Restart the server for the change to take effect. Supported in remote mode only. For procedural information in this guide, see "To Delete a Virtual Server" on page 119.
`list-http-listeners(1)`	Lists the existing HTTP listeners. Supported in remote mode only. For procedural information in this guide, see "To List HTTP Listeners" on page 113.
`list-virtual-servers(1)`	Lists the existing virtual servers. Supported in remote mode only. For procedural information in this guide, see "To List Virtual Servers" on page 118.

JVM Commands

`create-jvm-options(1)`	Creates a JVM option in the Java configuration or profiler elements of the `domain.xml` file. If JVM options are created for a profiler, the options are used to record the settings needed to activate a particular profiler. Restart the server for changes to take effect. Supported in remote mode only. For procedural information in this guide, see "To Create JVM Options" on page 56.
`create-profiler(1)`	Creates a profiler element. A server instance is tied to a particular profiler, by the profiler element in the Java configuration. Restart the server for changes to take effect. Supported in remote mode only. For procedural information in this guide, see "To Create a Profiler" on page 60.
`delete-jvm-options(1)`	Deletes the specified JVM option from the Java configuration or profiler elements of the `domain.xml` file. Restart the server for changes to take effect. Supported in remote mode only. For procedural information in this guide, see "To Delete JVM Options" on page 57.
`delete-profiler(1)`	Deletes the specified profiler element. Restart the server for changes to take effect. Supported in remote mode only. For procedural information in this guide, see "To Delete a Profiler" on page 60.
`generate-jvm-report(1)`	Generates a report showing the threads, classes, and memory for the virtual machine that runs Enterprise Server. For procedural information in this guide, see "To Generate a JVM Report" on page 58.
`list-jvm-options(1)`	Lists the command-line options that are passed to the Java application launcher when theEnterprise Server is started. Supported in remote mode only. For procedural information in this guide, see "To List JVM Options" on page 56.

Resource Management Commands

`add-resources(1)`	Creates the resources named in the specified XML file. Supported in remote mode only. For procedural information in this guide, see "To Add Resources" on page 45.

`create-jdbc-connection-pool(1)`	Registers a new JDBC connection pool with the specified JDBC connection pool name. Supported in remote mode only. For procedural information in this guide, see "To Create a JDBC Connection Pool" on page 68.
`create-jdbc-resource(1)`	Creates a new JDBC resource. Supported in remote mode only. For procedural information in this guide, see "To Create a JDBC Resource" on page 71.
`delete-jdbc-connection-pool(1)`	Deletes the specified JDBC connection pool. Supported in remote mode only. For procedural information in this guide, see "To Delete a JDBC Connection Pool" on page 70.
`delete-jdbc-resource(1)`	Deletes a JDBC resource. The specified JNDI name identifies the resource to be deleted. Supported in remote mode only. For procedural information in this guide, see "To Delete a JDBC Resource" on page 72.
`list-jdbc-connection-pools(1)`	Lists the existing JDBC connection pools. Supported in remote mode only. For procedural information in this guide, see "To List JDBC Connection Pools" on page 69.
`list-jdbc-resources(1)`	Lists the existing JDBC resources. Supported in remote mode only. "To List JDBC Resources" on page 72.
`ping-connection-pool(1)`	Tests if a JDBC connection pool is usable. Before you can ping a JDBC connection pool, you must create the connection pool with authentication and ensure that the server or database is started. Supported in remote mode only. For procedural information in this guide, see "To Contact (Ping) a Connection Pool" on page 69.

Security Commands

`change-admin-password(1)`	Modifies the admin password. You are prompted for the old and new admin password (with confirmation). For procedural information in this guide, see "To Change the Administration Password" on page 93.
`create-audit-module(1)`	Adds the named audit module for the plug-in that implements the audit capabilities. Supported in remote mode only. For procedural information in this guide, see "To Create an Audit Module" on page 94.

`delete-audit-module(1)`	Removes the named audit module. Supported in remote mode only. For procedural information in this guide, see "To Delete an Audit Module" on page 95.
`list-audit-modules(1)`	Lists all audit modules. Supported in remote mode only. For procedural information in this guide, see "To List Audit Modules" on page 95.

User Management Commands

`create-auth-realm(1)`	Adds the specified authentication realm. Restart the server for the creation to take effect. Supported in remote mode only. For procedural information in this guide, see "To Create an Authentication Realm" on page 102.
`create-file-user(1)`	Creates a file user in a given file-based authentication realm. An entry is added to the keyfile with the specified user name, password, and groups. Multiple groups can be created by separating each one with a colon (:). Restart the server for changes to take effect. Supported in remote mode only. For procedural information in this guide, see "To Create a File User" on page 106.
`delete-auth-realm(1)`	Deletes the specified authentication realm. Restart the server for changes to take effect. Supported in remote mode only. For procedural information in this guide, see "To Delete an Authentication Realm" on page 103.
`delete-file-user(1)`	Deletes the specified user entry in the keyfile. Restart the server for changes to take effect. Supported in remote mode only. For procedural information in this guide, see "To Delete a File User" on page 108.
`list-auth-realms(1)`	Lists the existing authentication realms. Supported in remote mode only. For procedural information in this guide, see "To List Authentication Realms" on page 103.
`list-file-users(1)`	Lists the file users supported by the `file` realm authentication method. Supported in remote mode only. For procedural information in this guide, see "To List File Users" on page 106.
`list-file-groups(1)`	Lists groups for a file user, or all groups if the `--name` option is not specified. For procedural information in this guide, see "To List File Groups" on page 107.
`update-file-user(1)`	Updates an existing entry in the keyfile using the specified user name, password, and groups. Restart the server for changes to take

effect. Supported in remote mode only. For procedural information in this guide, see "To Update a File User" on page 108.

Index

D

E

F

G

H

I

M

N

O

P

R

S

T

U

V

W