Sun Java System Application Server 9.1 Quick Start Guide

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

Part No: 819–3193–08
September 2007

080611@20490

Contents

Preface

This Quick Start Guide provides basic procedures to start the server, deploy an application, and set up load balancing and failover.

This preface contains information about and conventions for the entire Sun Java™ System Application Server documentation set.

Application Server Documentation Set

The Application Server documentation set describes deployment planning and system installation. The Uniform Resource Locator (URL) for Application Server documentation is `http://docs.sun.com/coll/1343.4`. For an introduction to Application Server, refer to the books in the order in which they are listed in the following table.

TABLE P–1 Books in the Application Server Documentation Set

Book Title	Description
Documentation Center	Application Server documentation topics organized by task and subject.
Release Notes	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	How to get started with the Application Server product.
Installation Guide	Installing the software and its components.
Deployment Planning Guide	Evaluating your system needs and enterprise to ensure that you deploy the Application Server in a manner that best suits your site. General issues and concerns that you must be aware of when deploying the server are also discussed.
Application Deployment Guide	Deployment of applications and application components to the Application Server. Includes information about deployment descriptors.
Developer's Guide	Creating and implementing Java Platform, Enterprise Edition (Java EE platform) applications intended to run on the Application Server that follow the open Java standards model for Java EE components and APIs. Includes information about developer tools, security, debugging, and creating lifecycle modules.

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

Book Title	Description
Java EE 5 Tutorial	Using Java EE 5 platform technologies and APIs to develop Java EE applications.
Java WSIT Tutorial	Developing web applications using the Web Service Interoperability Technologies (WSIT). Describes how, when, and why to use the WSIT technologies and the features and options that each technology supports.
Administration Guide	System administration for the Application Server, including configuration, monitoring, security, resource management, and web services management.
High Availability Administration Guide	Post-installation configuration and administration instructions for the high-availability database.
Administration Reference	Editing the Application Server configuration file, `domain.xml`.
Upgrade and Migration Guide	Upgrading from an older version of Application Server or migrating Java EE applications from competitive application servers. This guide also describes differences between adjacent product releases and configuration options that can result in incompatibility with the product specifications.
Performance Tuning Guide	Tuning the Application Server to improve performance.
Troubleshooting Guide	Solving Application Server problems.
Error Message Reference	Solving Application Server error messages.
Reference Manual	Utility commands available with the Application Server; written in man page style. Includes the `asadmin` command line interface.

Related Documentation

Application Server can be purchased by itself or as a component of Sun Java Enterprise System (Java ES), a software infrastructure that supports enterprise applications distributed across a network or Internet environment. If you purchased Application Server as a component of Java ES, you should be familiar with the system documentation at `http://docs.sun.com/coll/1286.2`. The URL for all documentation about Java ES and its components is `http://docs.sun.com/prod/entsys.5`.

For documentation about other stand-alone Sun Java System server products, go to the following:

- Message Queue documentation (`http://docs.sun.com/coll/1343.4`)
- Directory Server documentation (`http://docs.sun.com/coll/1224.1`)
- Web Server documentation (`http://docs.sun.com/coll/1308.3`)

A Javadoc™ tool reference for packages provided with the Application Server is located at `http://glassfish.dev.java.net/nonav/javaee5/api/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 on creating enterprise applications in the NetBeans™ Integrated Development Environment (IDE), see `http://www.netbeans.org/kb/55/index.html`.

For information about the Java DB database included with the Application Server, see `http://developers.sun.com/javadb/`.

The GlassFish Samples project is a collection of sample applications that demonstrate a broad range of Java EE technologies. The GlassFish Samples are bundled with the Java EE Software Development Kit (SDK), and are also available from the GlassFish Samples project page at `https://glassfish-samples.dev.java.net/`.

Default Paths and File Names

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

TABLE P–2 Default Paths and File Names

Placeholder	Description	Default Value

install-dir	Represents the base installation directory for Application Server.	Application Server installed with Java ES on Solaris™, root user:
		`/opt/SUNWappserver/appserver`
		Application Server installed with Java ES on Linux, root user:
		`/opt/sun/appserver/`
		Other Solaris and Linux installations, non-root user:
		user's-home-directory`/SUNWappserver`
		Other Solaris and Linux installations, root user:
		`/opt/SUNWappserver`
		Windows, all installations:
		SystemDrive`:\Sun\AppServer`

TABLE P–2 Default Paths and File Names *(Continued)*

Placeholder	Description	Default Value

domain-root-dir	Represents the directory containing all domains.	Java ES Solaris installations, root user:
		`/var/opt/SUNWappserver/domains/`
		Java ES Linux installations, root user:
		`/var/opt/sun/appserver/domains/`
		All other installations:
		install-dir`/domains/`

domain-dir	Represents the directory for a domain. In configuration files, you might see *domain-dir* represented as follows: `${com.sun.aas.instanceRoot}`	*domain-root-dir*`/`*domain-dir*
instance-dir	Represents the directory for a server instance.	*domain-dir*`/`*instance-dir*

Typographic Conventions

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

TABLE P–3 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–4 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.

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 819-3193.

Quick Start for Basic Features

Welcome to the *Quick Start Guide*. This guide provides a set of sample procedures that you can use to get started with the Application Server.

Before using the procedures in this document you must have already installed the Application Server software.

This section contains the following topics:

Application Server Administration Tools

To enable administrators to manage server instances and clusters running on multiple hosts, Application Server provides these tools:

- The Admin Console, a browser-based graphical user interface (GUI)
- Command-line tools, such as the `asadmin` utility. See Table 1–1 for the complete list of command-line tools available with Application Server.
- Programmatic Java Management Extensions (JMX™) APIs

These tools connect to a server called the *Domain Administration Server*, a specially designated Application Server instance that intermediates in all administrative tasks. The Domain Administration Server (DAS) provides a single secure interface for validating and executing administrative commands regardless of which interface is used.

A *domain* is a collection of configuration data, deployed applications, and machines with a designated administrator. The domain definition describes and can control the operation of several applications, stand-alone application server instances, and clusters, potentially spread over multiple machines. When the DAS is installed, a default domain called *domain1* is always installed. You work with the default domain in this guide.

The Application Server software offers a variety of command-line tools for performing administrative functions, in addition to the Admin Console. To launch a tool, type the name of the tool in a command window. Table 1–1 lists tools by name in the first column and describes them in the second column.

TABLE 1–1 Command-Line Tools

Name of Tool	Description
`appclient`	Launches the Application Client Container and invokes the client application packaged in the application Java archive (JAR) file.
`asadmin`	Launches the Application Server administration tool that provides a set of subcommands for configuring the Application Server software.
`asant`	Launches the Jakarta Ant tool, so that you can automate repetitive development and deployment tasks.
`asapt`	Compiles Java sources with Java EE annotations. The tool automatically invokes the `wsimport` command.
`asupgrade`	Application Server administration tool for upgrading the Application Server software.
`capture-schema`	Extracts schema information from a database and produces a schema file that the server can use for Container Managed Persistence (CMP).
`jspc`	Compiles JSP pages.
`package-appclient`	Packages the application client container libraries and JAR files.
`schemagen`	Creates a schema file for each namespace referenced in your Java classes.
`verifier`	Validates the Java EE deployment descriptors with the DTDs. This tool also provides a graphical user interface. To see the GUI, specify the `-u` option. Some Windows systems launch a driver verifier utility with the same name. To launch the Application Server verifier, you must be in the *install-dir*`/bin` directory.
`wscompile`	Takes the service definition interface and generates the client stubs or server-side skeletons for JAX-RPC; or generates a Web Services Description Language (WSDL) description for the provided interface.
`wsdeploy`	Generates an implementation-specific, ready-to-deploy WAR file for web services applications that use JAX-RPC.
`wsgen`	Reads a web service endpoint class and generates all the required artifacts for web service deployment and invocation.

TABLE 1–1 Command-Line Tools *(Continued)*

Name of Tool	Description
`wsimport`	Generates JAX-WS portable artifacts, such as service endpoint interfaces (SEIs), services, exception classes mapped from the `wsdl:fault` and `soap:headerfault` tags, asynchronous response beans derived from the `wsdl:message` tag, and JAXB generated value types.
`xjc`	Transforms, or binds, a source XML schema to a set of JAXB content classes in the Java programming language.

Note –

- To run these command-line tools on Windows, ensure that you have an environment variable called PATH that points to the *install-dir*`/bin/` directory.
- You can run the `asadmin` subcommands by prefixing `asadmin` with every sub command or by entering the `asadmin` prompt (type `asadmin` and hit Return) in the *install-dir*`/bin/` directory.

Starting the Server

This topic provides instructions for the following basic tasks: starting the Domain Administration Server (DAS), logging into the Admin Console, examining the log files, and starting the node agent. The steps are presented in the order that you should complete them.

- "To Start the Domain on Solaris and Linux" on page 13
- "To Start the Domain on Windows" on page 14
- "To Log On to the Admin Console" on page 15
- "To View the Domain Administration Server Log File" on page 16
- "To Start the Node Agent" on page 18
- "To Add Cluster Support to a Domain" on page 19
- "To Start the Bundled Java DB Server" on page 19

▼ To Start the Domain on Solaris and Linux

To start the Domain Administration Server, start the default domain, `domain1`.

1 **Add the *install-dir*`/bin/` directory to the `PATH` environment variable.**

2 **Start the server by entering this command:**

```
asadmin start-domain domain1
```

When you are prompted for the user, admin password, and the master password, enter your user name and the passwords that you provided during installation.

When the start up process is complete, the following message appears:

```
Domain domain1 started
```

Each domain has a corresponding profile: enterprise, cluster, or developer. `domain1` is created with the default profile, which is defined in the `AS_ADMIN_PROFILE` variable defined in the `asadminenv.conf` file. For more information on the three profiles see the "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*. If `domain1` was created with developer profile, when you log in to the Admin Console, you cannot use features such as clustering and node agents. To use clusters and server instances, use the `create-domain` command to create a domain with cluster profile. For information on creating domains, type `asadmin create-domain --help` or see `create-domain`(1) in the *Sun Java System Application Server 9.1 Reference Manual*.

▼ To Start the Domain on Windows

To start the Domain Administration Server, start the default domain, `domain 1`.

1. **From the Explorer window or desktop, right click My Computer.**

2. **Choose Properties to display the System Properties dialog.**

3. **Click the Advanced tab.**

4. **Click Environment Variables.**

5. **In the User variables section:**

   - **If a `PATH` variable exists, verify that *install-dir*`\bin` exists in the path: *install-dir*`\bin;`*other_entries*. If *install-dir*`\bin` does not exist in the PATH variable, add it.**

   - **If a `PATH` variable is not present, click New. In Variable Name, type `PATH`. In Variable Value, type the path to the server's `bin` directory: *install-dir*`\bin`. Click OK to commit the change.**

6. **Add a new environment variable `AS_ADMIN_USER` and set it to the Administrative User Name that you assigned during installation.**

7. **Click OK to commit the change and close the remaining open windows.**

8. **Start the Application Server by clicking the Start Admin Server option within the Programs menu.**

9 When a command prompt window opens to prompt you for the admin password and the master password, enter the passwords that you provided during installation.

A window appears with a message telling that you the server is starting:

```
Starting Domain domain1, please wait.
Log redirected to domain_dir\domain1\logs\server.log...
```

When the startup process has completed, you see an additional message:

```
Domain domain1 started.
Press any key to continue ...
```

Each domain has a corresponding profile: enterprise, cluster, or developer. `domain1` is created with the default profile, which is defined in the `AS_ADMIN_PROFILE` variable defined in the `asadminenv.conf` file. For more information on profiles see the "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*. If `domain1` was created with developer profile, when you log in to the Admin Console, you cannot use features such as clustering and node agents. To use clusters and server instances, use the `create-domain` command to create a domain with cluster profile. For information on creating domains, type `asadmin create-domain --help` or see `create-domain`(1) in the *Sun Java System Application Server 9.1 Reference Manual*.

10 Press a key to close the message window.

▼ To Log On to the Admin Console

The Admin Console is a browser interface that simplifies a variety of administration and configuration tasks. It is commonly used to:

- Deploy and undeploy applications
- Enable, disable, and manage applications
- Configure resources and other server settings
- Select and view log files

For further information about using the Admin Console, consult the online help or the *Sun Java System Application Server 9.1 Administration Guide*.

1 Type this URL in your browser:

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

If the browser is on a different system than the server, replace the *localhost* variable with the name of the system that the Domain Admin Server is running on.

`4848` is the Admin Console's default port number. If you changed the port number during the installation or if you have created a domain with another port number, use that number instead.

Note – If a popup window appears with a message such as `Website Certified by an Unknown Authority`, click OK.

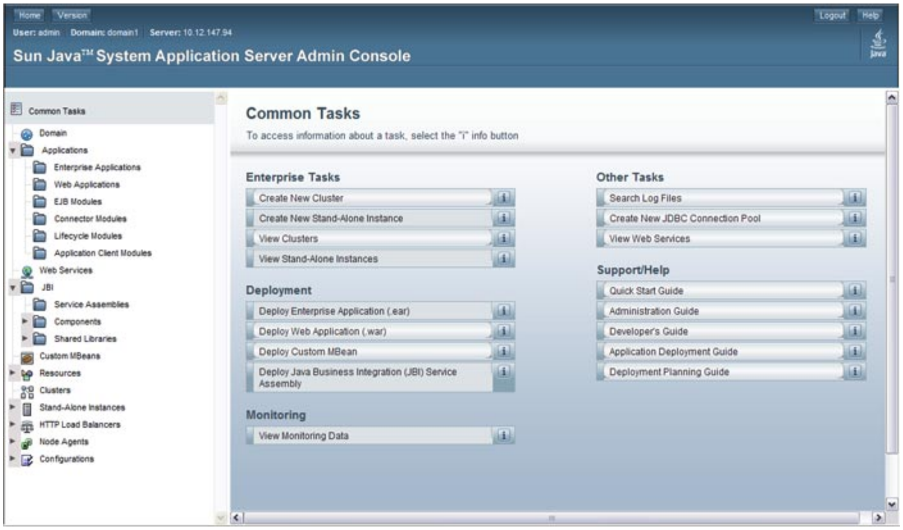
This message appears because your browser does not recognize the self-signed certificate that the Domain Administration Server uses to service the Admin Console over the secure transmission protocol.

---

2 **When the Log in window appears, enter the admin user name and password.**

3 **Click Login.**

The default user name is `admin` and password is `adminadmin`.



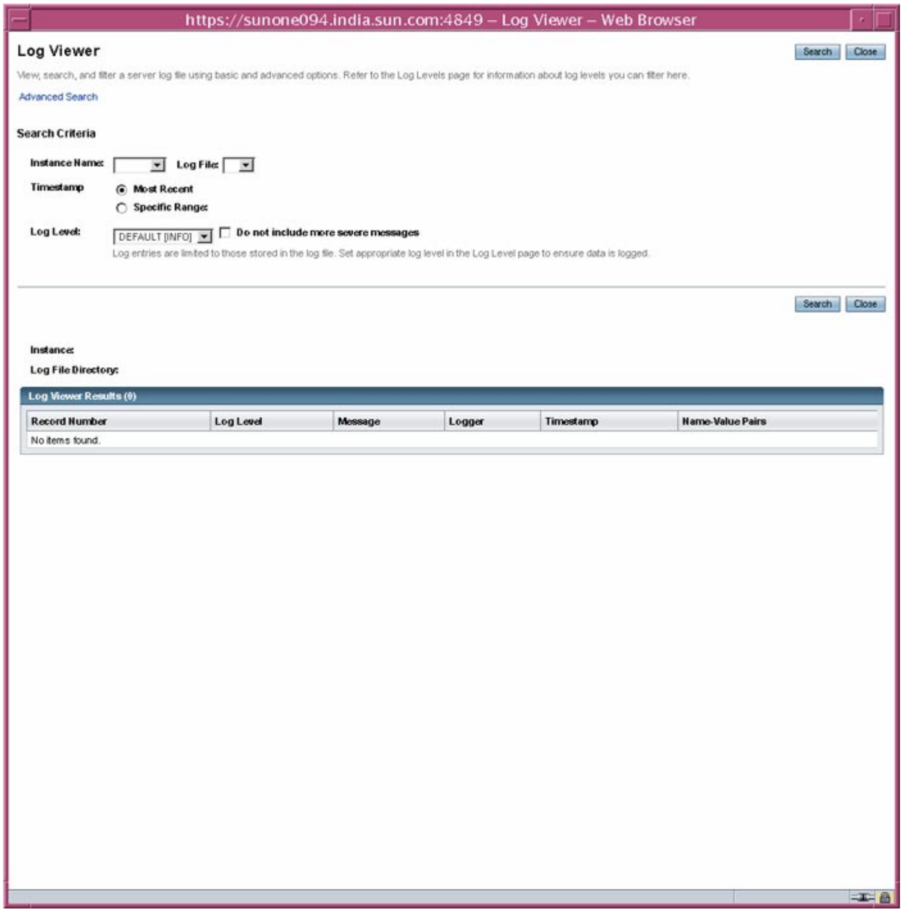
To access information about a task, select the "i" info button

In the left pane, select what you want to manage from the tree provided. In the right pane, various administrative tasks are listed under the Common Tasks heading.

▼ To View the Domain Administration Server Log File

Application Server instances and the Domain Administration Server produce annotated logs on the file system. By default, all errors, warnings or useful informative messages are logged.

1 **From the Common Tasks list in the right pane, click Search Log Files to launch a new browser window for Log Viewer.**


https://sunone094.india.sun.com:4849 – Log Viewer – Web Browser
Search
Close
Advanced Search
Search
Close

Log Viewer

View, search, and filter a server log file using basic and advanced options. Refer to the Log Levels page for information about log levels you can filter here.

Search Criteria

Instance Name:
Log File:
Timestamp: ● Most Recent ○ Specific Range:
Log Level: DEFAULT [INFO] ☐ Do not include more severe messages

Log entries are limited to those stored in the log file. Set appropriate log level in the Log Level page to ensure data is logged.

Instance:
Log File Directory:

Log Viewer Results (0)

Record Number	Log Level	Message	Logger	Timestamp	Name-Value Pairs
No items found.					

2 **In the Log Viewer window, select "server" from the Instance Name drop-down list and click Search.**

The Domain Administration Server's recent log file entries are displayed.

3 **Scan the messages and look for any `WARNING` or `SEVERE` messages indicating that problems were encountered during server startup.**

You can close Log Viewer at any time. After you create clusters and deploy applications, examine log files if any of the operations failed. Use Log Viewer to view the log files of any running Application Server instance in the domain.

▼ To Start the Node Agent

A *node agent* is a lightweight process running on each machine that participates in an Application Server administrative domain. The node agent is responsible for starting and stopping server instances on the host. It also communicates with the Domain Administration Server to create new Application Server instances.

Note – You can use node agents only on domains with the cluster or enterprise profile. If you are running only a developer domain, you need to create a domain with cluster profile before you can create and use node agents. For information on creating domains, type `asadmin create-domain --help` or see `create-domain`(1) in the *Sun Java System Application Server 9.1 Reference Manual*

One node agent is needed on a machine for each Application Server administrative domain that the machine belongs to. If you are using Sun Java System Application Server 9.1 (Enterprise bundle), choose the Node Agent Component during installation and a default node agent called *hostname* is created. If you have installed Glassfish, the node agent is not automatically created. To create a node agent, follow the instructions in “To Create a Node Agent” in *Sun Java System Application Server 9.1 High Availability Administration Guide*.

1 **In a terminal window, type this command:**

```
asadmin start-node-agent nodeagentname
```

Replace the variable *nodeagentname* with the name of the node agent you created. If you want to use the default node agent, which is created by Sun Java System Application Server 9.1, replace the *nodeagentname* with the name of the host on which Application Server is running. If your default domain on port 4848 (`domain1`) is not running on cluster or enterprise profile, use the `--port` option to specify the port of the cluster or enterprise domain.

2 **When you are prompted, provide the master password.**

The node agent starts and connects with the Domain Administration Server.

Note – Ensure that the Domain Administration Server is running before you start the node agent. Check the `server.log` file of a cluster instance to identify problems related to starting node agents or cluster instances.

If you have problems while starting a node agent or a cluster on Windows, see the *Sun Java System Application Server 9.1 Release Notes* for possible solutions to the problem.

After the node agent has been started once, you can view it in the Admin Console.

Next Steps In this section you started the Domain Administration Server and confirmed that it is running. You also logged in to the Admin Console and used the Log Viewer. You started the node agent. You can stop the Quick Start trail here if you do not wish to continue, or you can go on to the next section.

▼ To Add Cluster Support to a Domain

You can use the Admin Console add cluster support to a developer domain.

1. **Log on to the Admin Console. For instructions, see "To Log On to the Admin Console" on page 15.**
2. **From the left pane, click Application Server.**
3. **Click the Add Cluster Support button in the General tab.**
4. **The Add Cluster Support page appears. Review the information on this page and click OK.**
5. **In the Restart Required Page, click the Stop Instance button to stop the domain.**
6. **Log in to the machine where the domain is installed and run the following command:** *install-dir*`/bin/asadmin list-domains`**. Verify that the domain was stopped.**
7. **Run the following command:** *install-dir*`/bin/asadmin start-domain` <*domain-name*>

   The restarted domain supports clustering.

More Information

Troubleshooting

If the restarted domain does not show clustering capabilities, stop the domain, reboot the machine on which the domain is installed, and restart the domain as shown in step 7.

▼ To Start the Bundled Java DB Server

The Application Server includes a bundled Java DB server. To start it, use the following procedure:

1. **Make sure that you are logged in as the user that installed the database.**
2. **Run `asadmin start-database`.**

   For more information on the `start-database` command, see the command's help or the `start-database`(1) information in *Sun Java System Application Server 9.1 Reference Manual*.

Where to Go Next

The next steps of the *Quick Start Guide* are available in the following chapters.

- Chapter 2, "Setting Up a Cluster"

  Steps for creating a sample cluster. Skip to the next chapter if you are running a domain with developer profile.
- Chapter 3, "Deploying an Application"

  Steps for deploying a sample web application. If you are using a domain with cluster capabilities, you can perform the steps for deploying an enterprise application to a cluster.
- Chapter 4, "Working with Load Balancers"

  Steps for creating and using a load balancer. Requires installation of the load balancer plug-in and the Web Server.

Other resources for learning and using Application Server are available. They include:

- Product details at *install-dir*`/docs-ee/about.html`.
- Sample applications at *install-dir*`/samples`
- *Sun Java System Application Server 9.1 Release Notes*
- *Sun Java System Application Server 9.1 Administration Guide*
- *Sun Java System Application Server 9.1 Reference Manual* (man pages)
- *Sun Java System Application Server 9.1 Application Deployment Guide*
- *Sun Java System Application Server 9.1 High Availability Administration Guide*
- *The Java EE 5 Tutorial*
- Java BluePrints (`http://java.sun.com/blueprints`) guidelines for the Enterprise

◆ ◆ ◆

CHAPTER 2

Setting Up a Cluster

This chapter describes how to create and set up clusters. A *cluster* is a group of server instances (typically on multiple hosts) that share the same configurations, resources, and applications. A cluster facilitates load balancing across server instances and high availability through failover. You can create clusters spanning multiple machines and manage them with the help of the node agent process on each machine.

This chapter assumes that you are running a domain with cluster or enterprise profile. This chapter has the following sections:

Creating a Cluster

This section explains how to create a cluster that contains two Application Server instances.

▼ To Create a Cluster using the Admin Console

For simplicity, the cluster runs completely within one machine. This procedure creates the sample cluster one the same host on which the DAS is running.

Before You Begin You must have already started the node agent, as described in "To Start the Node Agent" on page 18. When you specify instances during cluster creation, associate the instance with a running node agent for the machine on which you want the instance to run. If the node agent is not running, the instance will not start. Node agent and instance names must be unique across clusters that are created in a domain.

1. **Log in to the Administration Console at `http://` *localhost*`:`*portnumber* if you have not already done so.**

   Replace the *localhost* variable with the name of the system that the Domain Administration Server is running on. Replace *portnumber* with the port number of the cluster or enterprise domain.

2. **On the right pane, under Common Tasks, click Create New Cluster to display the New Cluster page.**

3. **Type `cluster1` as the name of the new cluster.**

4. **From the drop-down list of available configuration templates, select the default-config configuration and choose `Make a copy of the selected Configuration`.**

5. **Click the New button twice to create two entries to specify two instances for the cluster.**

6. **Type `instance1` and `instance2` as instance names.**

   Alternatively, you can specify the name of a node agent that you created and started. To create a node agent and start it, use the `asadmin` `create-node-agent`(1) command.

7. **Click OK.**

   The Cluster Created Successfully page appears, and `cluster1` appears in the tree in the left pane. A copy of the configuration template `default-config` was made for this cluster, and the name `cluster1-config` was assigned to it.

8. **Start `cluster1` Cluster.**

▼ To Create a Cluster using CLI Commands

1. **Create a cluster called `cluster1` by running the following `asadmin` command:** `asadmin create-cluster --user admin --passwordfile adminpassword.txt --host localhost -port 4848 cluster1`

2. **Create a node-agent called `cluster1-nodeagent` by running the following command:** `asadmin create-node-agent --user admin --passwordfile adminpassword.txt --host localhost --port 4848 cluster1-nodeagent`

3. **Start this node-agent by running the following command:** `asadmin start-node-agent --user admin --passwordfile adminpassword.txt --host localhost --port 4848 cluster1-nodeagent`

4. **Create two instances under the cluster by running the following commands:**

-
  ```
  asadmin create-instance --user admin --passwordfile adminpassword.txt --host
  localhost -port 4848 --cluster cluster1 --nodeagent cluster1-nodeagent
  --systemproperties
  "JMX_SYSTEM_CONNECTOR_PORT=8687:IIOP_LISTENER_PORT=3330:IIOP_SSL_LISTENER_PORT=444
  MUTUALAUTH_PORT=5550:HTTP_LISTENER_PORT=1110:HTTP_SSL_LISTENER_PORT=2220"
  instance1
  ```

-
  ```
  asadmin create-instance --user admin --passwordfile adminpassword.txt --host
  localhost -port 4848 --cluster cluster1 --nodeagent cluster1-nodeagent
  --systemproperties
  "JMX_SYSTEM_CONNECTOR_PORT=8688:IIOP_LISTENER_PORT=3331:IIOP_SSL_LISTENER_PORT=444
  MUTUALAUTH_PORT=5551:HTTP_LISTENER_PORT=1111:HTTP_SSL_LISTENER_PORT=2221"
  instance2
  ```

5 **Start the cluster by running the following command:** `asadmin start-cluster --user admin --passwordfile adminpassword.txt --host localhost --port 4848 cluster1`

See Chapter 6, “Using Application Server Clusters,” in *Sun Java System Application Server 9.1 High Availability Administration Guide* for detailed information on creating and configuring clusters.

SeeChapter 8, “Configuring Node Agents,” in *Sun Java System Application Server 9.1 High Availability Administration Guide* for detailed information on creating and configuring node agents.

Viewing Ports for Clustered Server Instances

▼ To View Ports for Clustered Server Instances

You can view the ports numbers and override the default values for the clustered server instance you created in the previous section, “Creating a Cluster” on page 21. By default, the HTTP ports are 38081 for `instance1` and 38080 for `instance2`. If these ports were busy on your machine when you created these clusters, or if you had already assigned these ports to other instances and clusters, different port numbers were assigned.

To view the port numbers and optionally override the default values, follow these steps:

1 **In the left pane, expand Clusters and click `cluster1` to display the General Information page for clusters.**

2 **Click the Instances tab to display `instance1` and `instance2`, the instances that you created.**

   a. **Click `instance1` to examine this instance.**

   The HTTP and HTTPS ports assigned to the instance are displayed in the HTTP Ports field.

b. **Repeat these steps for `instance2`.**

Next Steps In this section you have created a simple cluster on a single machine. You can also create clusters spanning multiple machines using the same basic steps (as long as you have the software installed and a node agent running on each machine).

◆ ◆ ◆ CHAPTER 3

Deploying an Application

This chapter explains how to deploy an application. If you are running a domain with developer profile, see "Deploying a Sample Web Application" on page 25. If you are running a domain with cluster profile, see "Deploying the Sample Enterprise Application to a Cluster" on page 28.

This chapter presents the following sections:

Deploying a Sample Web Application

These procedures assume that you are running a domain with developer profile.

▼ To Deploy the Hello Application Using Autodeploy

The server installation includes a pre-packaged application that says "Hello," taken from the *The Java EE 5 Tutorial*. The Hello application is contained in a Web ARchive (WAR) file in the `samples/quickstart` directory. Packaged applications can also be in the form of a Java ARchive (JAR) or Enterprise ARchive (EAR) file.

Alternatively, you can download this application from`https://glassfish.dev.java.net/downloads/quickstart/hello.war` .

Copying to a domain's `autodeploy` directory lets you put a pre-packaged application into use immediately, with minimal effort.

1 **Find** ***install-dir*****`/samples/quickstart/hello.war`** **or download from** **`https://glassfish.dev.java.net/downloads/quickstart/hello.war`.**

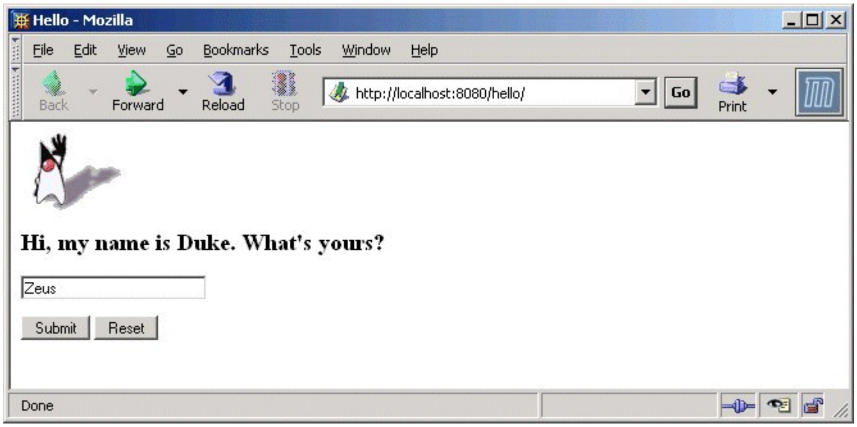
2 **Copy** `hello.war` **to the** *install-dir*`/domains/domain1/autodeploy/` **directory.**

Tip – On Windows, you can create a shortcut to the `autodeploy` directory on your desktop, and then drag and drop the `hello.war` file onto the shortcut.

When the server has finished deploying the application, it creates a file named `hello.war_deployed` in the `autodeploy` directory. Depending on the speed of your system, the process can take a few seconds. Until that file appears, a `404-File Not Found` error occurs when you try to visit the application page.

3 **Access this URL to visit the application page:** `http://`*localhost*`:8080/hello`**.**

You see the application's first page, which prompts you to fill in your name:



http://localhost:8080/hello/

Hi, my name is Duke. What's yours?

Zeus

4 **Type your name and click Submit.**

The application displays a customized response, giving you a personal `Hello`.

▼ To Deploy the Hello Application Using Admin Console

1 **Open** `http://`*localhost*`:4848/asadmin` **in a web browser.**

- *localhost* is used if the both the browser and the Application Server are running on the same system. If the Application Server is running on another system, substitute the name of that system in the URL.
- 4848 is the Admin Console's default port number. If you changed the port number during the installation, use that number instead.

2 **Enter the admin user name and password.**

3 **In the left pane, click the Applications node to expand it.**

4 **Click Web Applications.**

You can see the list of deployed web applications, if any.

5 **If you already have a Hello application deployed, undeploy it now by selecting the checkbox next to it and clicking Undeploy.**

6 **Click Deploy.**

7 **You can use the Hello application bundled with the Application Server or you upload the sample you have downloaded from `https://glassfish.dev.java.net/downloads/quickstart/hello.war`. Do one of the following:**

- **Select the option called Local packaged file or directory that is accessible to Application Server and click Browse Files. Navigate to the *install-dir*`/samples/quickstart` directory and select `hello.war`.**

8 **If you have downloaded the Hello application Select the Package file to upload to the Application Server option and click Browse. Navigate to the directory to which you have downloaded the application.**

9 **Click OK.**

The application appears in the Web Applications list.

10 **To verify that it was deployed properly, in the Web Applications page, select the Hello application and click Launch.**

More Information

Deploying Applications Using CLI Commands

You can deploy applications using the `asadmin deploy` command. See `deploy`(1) in the *Sun Java System Application Server 9.1 Reference Manual*. You can also use the `asant` commands for deployment. For detailed information on deploying applications see “Deploying Modules and Applications” in *Sun Java System Application Server 9.1 Application Deployment Guide*.

Deploying the Sample Enterprise Application to a Cluster

These procedures assume that you are running a domain with cluster profile. The procedures in this chapter assume that you have already created a cluster called `cluster1`, as described in Chapter 2, "Setting Up a Cluster." This section uses the `clusterjsp` sample application to demonstrate HTTP load balancing capabilities.

▼ To Deploy the Sample Enterprise Application Using Admin Console

1. **Click the Home button or click Common Tasks on the left pane to go to the Common Tasks page, if you are not there already.**
2. **On the right pane, under Common Tasks, click Deploy Enterprise Application (.ear).**
3. **If application server is installed on your local machine, click Browse in the Packaged File to Upload text box, and navigate to** *install-dir*`/samples/ee-samples/highavailability/apps/clusterjsp/clusterjsp.ear`**. If you are accessing Admin Console remotely with application server installed on another machine, click Browse in the Local packaged file or directory that is accessible from the Application Server text box. Navigate to** *install-dir*`/samples/quickstart/clusterjsp/clusterjsp.ear`.
4. **In the Targets section of the page, select** `cluster1` **from the Available list, and click Add to move it to the Selected list.**
5. **Click OK.**

   The `clusterjsp` application is now deployed to `cluster1`. Before you launch this application, you need to start the instances in the cluster. See "To Start the Instances in a Cluster" on page 29

More Information

Deploying Applications Using CLI Commands

You can deploy applications using the `asadmin deploy` command. See `deploy`(1) in the *Sun Java System Application Server 9.1 Reference Manual*. You can also use the `asant` commands for deployment. For detailed information on deploying applications see "Deploying Modules and Applications" in *Sun Java System Application Server 9.1 Application Deployment Guide*.

▼ To Start the Instances in a Cluster

Before you launch the deployed application, you need to start the instances in the cluster.

1. **In the tree on the left pane, click Clusters, and click the `cluster1` node.**
2. **In the right pane, navigate to the General tab if it is not already active.**
3. **If the instances are not already running, the Start Instances button is active. Click it to start the cluster's instances.**
4. **Verify that they have started by checking the Status field, which indicates how many instances are running.**

   You are now ready to launch the application. See"To Verify Application Deployment" on page 29

More Information

Starting Instances Using CLI Command

You can start instances by using the `asadmin start-intance` command. See in the *Sun Java System Application Server 9.1 Reference Manual*.

▼ To Verify Application Deployment

1. **Type the following URL in your browser:**

   `http://`*localhost*`:`*port*`/clusterjsp`

   Replace the *localhost* variable with the name of the system that the Domain Admin Server is running on.

   Replace the *port* variable with the value of `HTTP-LISTENER-PORT` for `instance1`. This example uses `http://`*localhost*`:38081/clusterjsp`.

2. **Add some session attribute data.**
3. **Examine the Session and Host information displayed. For example:**
   - `Executed From Server:` *localhost*
   - `Server Port Number: 38081`
   - `Executed Server IP Address: 198.19.255.255`
   - `Session Created: Tue Aug 23 15:26:07 PDT 2005`
4. **Add some session data and click the Add to Session button.**

5 **Repeat this procedure for instance `instance2` by typing this URL in your browser:**

`http://`*localhost*`:38080/clusterjsp`

Next Steps Congratulations! You have completed this section of the *Quick Start Guide*. You can stop the Quick Start trail here if you do not wish to proceed to the steps on load balancing.

◆ ◆ ◆

CHAPTER 4

Working with Load Balancers

This section provides instructions on how to set up the Web Server software to act as a load balancer to the cluster of Application Servers. In addition, it provides steps for configuring a load balancer and exporting it to the Web Server. The load balancer feature is available to you only if you are running a domain with enterprise profile or cluster profile.

A load balancer is deployed with a cluster. A load balancer provides the following features:

- Allows an application or service to be scaled horizontally across multiple physical (or logical) hosts yet still presents the user with a single URL
- Insulates the user from host failures or server crashes, when it is used with session persistence.
- Enhances security by hiding the internal network from the user

Application Server includes load balancing plug-ins for popular web servers such as Sun Java™ System Web Server, Apache, and Microsoft Windows IIS.

To complete this section, you must have sufficient memory to run a Web Server on your system in addition to the Domain Administration Server and the two Application Server instances you have created so far in this guide. A system with 512 Mbytes to 1024 Mbytes of memory is recommended.

This topic presents the following steps:

- "Setting up Load Balancing" on page 32
- "Starting Load Balancing" on page 33
- "Verifying Load Balancing" on page 33
- "High Availability and Failover Using the In-memory Replication Feature" on page 34

Setting up Load Balancing

Before you set up load balancing, you need to install the load balancer plug-in. For instructions on installing the load balancing plugin, see "To Install the Load Balancing Plug-in (standalone)" in *Sun Java System Application Server 9.1 Installation Guide*. These procedures assume you are running a domain with cluster or enterprise profile.

▼ To Set Up Load Balancing

1. **Create a load balancer using the Admin Console. Alternatively, you can use the** `asadmin create-http-lb`**(1) command.**
    - a. **Click the HTTP Load Balancers node in the Admin Console.**
    - b. **Click New.**
    - c. **Type** `lb1` **as the name of the load balancer, and select the Apply Changes Automatically check box. If you choose this option, you do not have to export the load balancer configuration. All changes you make to the load balancer configuration are propagated automatically.**
    - d. **Type the host on which Web Server is installed, and the Web Server instance port. In this sample scenarios Web Server host is localhost and the port is 38000.**
    - e. **Select** `cluster1` **as target.**

      "Creating a Cluster" on page 21 explains how to create a sample cluster (`cluster1`)
    - f. **Click Save.**
2. **Enable** `cluster1` **for load balancing:**

   ```
   asadmin enable-http-lb-server cluster1
   ```
3. **Enable the** `clusterjsp` **application for load balancing:**`asadmin enable-http-lb-application --name clusterjsp cluster1`**.**

See Also For information on advanced topics, such as changing the load balancer configuration or creating health checkers, see the Chapter 5, "Configuring HTTP Load Balancing," in *Sun Java System Application Server 9.1 High Availability Administration Guide*.

Starting Load Balancing

Start load balancing by starting or restarting the Web Server.

- If the Web Server instance serving as load balancer is not already running, start the Web Server.

  If you are using Web Server 7.0, use the `wadm start-instance` command.

  For Web Server 6.1, run the `start` script in the <*websvr-instance-dir*> directory.
- If the Web Server instance serving as load balancer is already running, stop the Web Server and restart.

  For Web Server 6.1, use the stop program in *web_server_install_dir*`/https-`*hostname* and restart the server by running the start program.

  For Web Server 7.0, use the `wadm stop-instance` followed by the `wadm start-instance` command.

Verifying Load Balancing

Once the application is deployed and the load balancer is running, verify that the load balancing is working.

▼ To Verify Load Balancing

1. **To display the first page of the `clusterjsp` application, type this URL in your browser:**

   `http://`*localhost*`:`*web_server_port*`/clusterjsp`

   Replace the *localhost* variable with the name of the system that the Web Server is running on.

   Replace the *web_server_port* variable with the value of the port attribute of the `LS` element in *web_server_install_dir*`/https-`*hostname*`/config/server.xml`. For this example, port 38000 is used.

   A page similar to what you saw in "To Verify Application Deployment" on page 29. appears.

2. **Examine the Session and Host information displayed. For example:**

   - `Executed From Server:` *localhost*
   - `Server Port Number: 38000`
   - `Executed Server IP Address: 192.18.145.133`
   - `Session Created: Day Mon 05 14:55:34 PDT 2005`

3. **The Server Port Number is 38000, the Web Server's port. The load balancer has forwarded the request on the two instances in the cluster.**

4 **Using different browser software, or a browser on a different machine, create a new session. Requests from the same browser are "sticky" and go to the same instance.**

These sessions should be distributed to the two instances in the cluster. You can verify this by looking at the server access log files located here:

- Solaris Java Enterprise System installation:

  `/var/opt/SUNWappserver/nodeagents/`*nodeagent_name*`/instance1/logs/access/server_access_log`

  `/var/opt/SUNWappserver/nodeagents/`*nodeagent_name*`/instance2/logs/access/server_access_log`
- Linux Java Enterprise System installation:

  `/var/opt/sun/appserver/nodeagents/`*nodeagent_name*`/instance1/logs/access/server_access_log`

  `/var/opt/sun/appserver/nodeagents/`*nodeagent_name*`/instance2/logs/access/server_access_log`
- Windows Java Enterprise System installation:

  install-dir`\nodeagents\`*nodeagent_name* `\instance1\logs\access\server_access_log`

  install-dir`\nodeagents\`*nodeagent_name*`\instance1\logs\access\server_access_log`
- Stand-alone Application Server installations:

  install-dir`/nodeagents/`*nodeagent_name*`/instance1/logs/access/server_access_log`

  install-dir`/nodeagents/`*nodeagent_name*`/instance2/logs/access/server_access_log`

5 **Add a name and value pair (**`Name=Name Value=Duke`**) for storing in** `HttpSession`**.**

6 **Click the Add to Session Data button.**

7 **Verify that the session data was added**

High Availability and Failover Using the In-memory Replication Feature

GlassFish v2 does not offer HADB. For high availability and failover, GlassFish offers the in-memory replication feature. The following procedure illustrates this feature:

1. Restart the web server that has the load balancer plugin installed before deploying an application. This ensures that requests are served by instances in the order set in the `loadbalancer.xml` file. If you use the `loadbalancer.xml` file provided in this chapter, `instance1` serves the first request.

2. You have already deployed the `clusterjsp` web application, which stores session data. You should be able to see that successive requests are served by the same instance that served the first request and the session data is maintained across the requests.
3. Send few requests and note down the instance that served those requests and shutdown that particular instance. Use this command to stop the instance: `asadmin stop-instance --user adminuser --password adminpassword instance1`
4. Send in the next request and verify that the new data is stored and that the previously added data is still there in the session. If one of the server serving requests is not available, another server in the same cluster takes over the request with all earlier session data and completes the request.

◆ ◆ ◆

CHAPTER 5

Setting Up High Availability Failover

With the configuration used in the previous chapter, if a server instance goes down, users lose session state. This section, the second of two advanced topics, provides the steps for installing the high-availability database (HADB), creating a highly available cluster, and testing HTTP session persistence.

GlassFish v2 does not offer HADB. For high availability and failover, GlassFish offers the in-memory replication feature. For more information, see

Application Server supports both HTTP session persistence and persistence for Stateful Session Beans. The procedures in this chapter cover high availability using in-memory replication or HADB.

These steps assume you have already performed the steps in the previous sections of this Quick Start. The steps are presented in the order that you should complete them. To use the HADB feature, you need to be running a domain with enterprise profile.

Note – Completing the procedures in this section may require additional hardware resources.

This topic contains the following sections:

High-availability Clusters and HADB

A highly availability cluster inSun Java System Application Server integrates a state replication service with the clusters and load balancer created earlier, enabling failover of HTTP sessions.

`HttpSession` objects and Stateful Session Bean state is stored in HADB, a high-availability database for storing session state. This horizontally scalable state management service can be managed independently of the application server tier. It was designed to support up to 99.999% service and data availability with load balancing, failover and state recovery capabilities.

Keeping state management responsibilities separated from Application Server has significant benefits. Application Server instances spend their cycles performing as a scalable and high performance Java™ Platform, Enterprise Edition 5 (Java EE™ 5 platform) containers delegating state replication to an external high availability state service. Due to this loosely coupled architecture, application server instances can be easily added to or deleted from a cluster. The HADB state replication service can be independently scaled for optimum availability and performance. When an application server instance also performs replication, the performance of J2EE applications can suffer and can be subject to longer garbage collection pauses.

Because each HADB node requires 512 Mbytes of memory, you need 1 Gbyte of memory to run two HADB nodes on the same machine. If you have less memory, set up each node on a different machine. Running a two-node database on only one host is not recommended for deployment since it is not fault tolerant.

HADB Preinstallation Steps

This procedure covers the most common preinstallation tasks. For information on other preinstallation topics, including prerequisites for installing HADB, configuring network redundancy, and file system support, see Chapter 2, “Installing and Setting Up High Availability Database,” in *Sun Java System Application Server 9.1 High Availability Administration Guide*.

The recommended system configuration values in this section are sufficient for running up to six HADB nodes and do not take into consideration other applications on the system that also use shared memory.

▼ To Configure Your System for HADB

1. **Get root access.**

2. **Define variables related to shared memory and semaphores**

   - **On Solaris:**

     a. **Add these lines to the `/etc/system` file (or if these lines are in the file as comments, uncomment them and make sure that the values match these):**

     ```
     set shmsys:shminfo_shmmax=0x80000000

     set shmsys:shminfo_shmseg=36

     set semsys:seminfo_semmnu=600
     ```

     Set `shminfo_shmmax` to the total memory in your system (in hexadecimal notation the value `0x80000000` shown is for 2 Gigabytes of memory).

     If the `seminfo_*` variables are already defined, increment them by the amounts shown. The default values for `seminfo_semmni` and `seminfo_semmns` do not need to be changed. The variable `shminfo_shmeg` is obsolete after Solaris 8.

     b. **Reboot, using this command:**

     ```
     sync; sync; reboot
     ```

   - **On Linux:**

     a. **Add these lines to the `/etc/sysctl.conf` file (or if they are in the file as comments, uncomment them). Set the value to the amount physical memory on the machine. Specify the value as a decimal number of bytes. For example, for a machine having 2 GB of physical memory:**

     ```
     echo 2147483648 > /proc/sys/shmmax

     echo 2147483648 > /proc/sys/shmall
     ```

     b. **Reboot, using this command:**

     ```
     sync; sync; reboot
     ```

   - **On Windows: No special system settings are needed.**

3. **If you used existing JDK software when you installed a standalone Application Server, check the JDK version.**

   HADB requires Sun JDK 1.4.1_03 or higher (for the latest information on JDK versions, see the *Sun Java System Application Server 9.1 Release Notes*). Check the version installed, and if it is not done already, set the `JAVA_HOME` environment variable to the directory where the JDK is installed.

4. **If necessary after the reboot, restart the domain, Web Server, and node agent.**

   To restart the domain, use the command `asadmin start-domain domain1`.

   To restart the Web Server, execute the start program in *web_server_install_dir*/`https-`*hostname*.

   To restart the node agent, use the command `asadmin start-node-agent` *hostname*. Replace the variable *hostname* with the name of the host where the Application Server is running.

Installing HADB

This section provides the steps for installing the high-availability database (HADB).

Note – If you plan to run the high-availability database on the Application Server machine, and if you installed HADB when you installed Application Server, skip to "Starting HADB" on page 41.

You can install the HADB component on the same machine as your Application Server system if you have 2 Gbytes of memory and 1-2 CPUs. If not, use additional hardware. For example:

- Two 1 CPU systems with 512 Mbytes to 1 Gbyte memory each
- One 1-2 CPU system with 1 Gbytes to 2 Gbytes memory

▼ To Install HADB

1. **Run the Application Server or Java Enterprise System installer.**

2. **Choose the option to install HADB.**

3. **Complete the installation on your hosts.**

Starting HADB

This section describes starting the HADB management agent in most cases by running the `ma-initd` script. For a production deployment, start the management agent as a service to ensure its availability. For more information, see "Starting the HADB Management Agent" in *Sun Java System Application Server 9.1 High Availability Administration Guide*.

If starting a database with HADB nodes on several hosts, start the management agent on each host.

▼ To Start HADB in a Java Enterprise System Installation on Solaris or Linux

1. **Change to the `/etc/init.d` directory:**

   ```
   cd /etc/init.d
   ```

2. **Run the command to start the agent:**

   ```
   ./ma-initd start
   ```

▼ To Start HADB in a Java Enterprise System Installation on Windows

HADB is started by default when Sun Java System is configured and running. However, if you need to start it manually, follow these steps:

1. **Go to Start⇒Settings⇒Control Panel, and double click Administrative Tools.**
2. **Double click Services shortcut.**
3. **Select HADBMgmtAgent Service from the Services list.**
4. **From the Action menu, select Start.**

▼ To Start HADB in a Stand-Alone Installation on Solaris or Linux

1. **Change to the HADB `bin` directory in the Application Serverinstallation:** *install-dir*`/hadb/4/bin`

2. **Run the command to start the agent:**

```
./ma-initd start
```

▼ To Start HADB in a Stand-Alone Installation on Windows

1. **In a terminal window, change to the HADB `bin` directory in the Application Serverinstallation:** *install-dir*`\hadb\4.`*x*`\bin`

   The *x* represents the release number of HADB.

2. **Run the command to start the agent:**

```
ma -i ma.cfg
```

Configuring a Cluster and Application for High Availability

The FirstCluster cluster must be configured to use HADB and high-availability must be enabled for the clusterjsp application before you can verify HTTP session persistence. Use the `asadmin configure-ha-cluster` command to configure an existing cluster for high availability. For more information on how to use this command, type `configure-ha-cluster --help` at the `asadmin` command prompt or see the `configure-ha-cluster`(1) man page.

Restarting the Cluster

Before the changes made in the previous section take effect, the cluster's instances must be restarted.

▼ To Restart the Cluster

1. **In the Admin Console, expand the Clusters node.**

2. **Click `FirstCluster`.**

3 **In the right pane, click Stop Instances.**

4 **Once the instances are stopped, click Start Instances.**

Verifying HTTP Session Failover

The steps for testing session data failover are similar for testing load balancing as described in the topic "Verifying Load Balancing" on page 33. This time Session Data is preserved after failure. Failover is transparent to the user because the sample application is configured for automatic retry after failure.

▼ To Verify HTTP Session Failover

1 **To display the first page of the clusterjsp application, type this URL in your browser:**

`http://`*localhost*`:`*web_server_port*`/clusterjsp`

Replace the *localhost* variable with the name of the system that the Web Server is running on.

Replace the *web_server_port* variable with the value of the port attribute of the `LS` element in *web_server_install_dir*`/https-`*hostname*`/config/server.xml`. For this example, port 38000 is used.

A page similar to what you saw in "To Verify Application Deployment" on page 29 appears.

2 **Examine the Session and Host information displayed. For example:**

- `Executed From Server:` *localhost*
- `Server Port Number: 38000`
- `Executed Server IP Address: 192.18.145.133`
- `Session ID: 41880f618e4593e14fb5d0ac434b1`
- `Session Created: Wed Feb 23 15:23:18 PST 2005`

3 **View the server access log files to determine which application server instance is serving the application. The log files are located here:**

- Solaris Java Enterprise System installation:

  `/var/opt/SUNWappserver/nodeagents/`*nodeagent_name*`/i1/logs/access/server_access_log`

  `/var/opt/SUNWappserver/nodeagents/`*nodeagent_name*`/i2/logs/access/server_access_log`

- Linux Java Enterprise System installation:

  `/var/opt/sun/appserver/nodeagents/`*nodeagent_name*`/i1/logs/access/server_access_log`

`/var/opt/sun/appserver/nodeagents/`*nodeagent_name*`/i2/logs/access/server_access_log`

- Windows Java Enterprise System installation:

  install-dir`\nodeagents\`*nodeagent_name*`\i1\logs\access\server_access_log`

  install-dir`\nodeagents\`*nodeagent_name*`\i2\logs\access/server_access_log`
- Standalone Application Server installations:

  install-dir`/nodeagents/`*nodeagent_name*`/i1/logs/access/server_access_log`

  install-dir`/nodeagents/`*nodeagent_name*`/i2/logs/access/server_access_log`

4. **Stop the Application Server instance that is serving the page.**

   a. **In the Admin Console, in the left pane, expand Clusters.**

   b. **Click** `FirstCluster`**.**

   c. **In the right pane, click the Instances tab.**

   d. **Click the checkbox next to the server instance that served the request and click the Stop button.**

5. **Reload the** `clusterjsp` **sample application page.**

   The session ID and session attribute data is retained.

6. **Check the access log of the other Application Server instance, and notice that it is now servicing the request.**

   The state failover features work because the HTTP session is stored persistently in the HADB. In addition to the HTTP session state, the Application Server also can store the state of EJB in the HADB.

Next Steps Congratulations! You have now completed the Quick Start for Application Server.

In this section, you have installed, configured, and started HADB and configured a cluster and an application to use high availability. See“Where to Go Next” on page 20 for additional information on Application Server.