



Sun Java™ System

Application Server  
Enterprise Edition 8.1  
Quick Start Guide

---

2005Q1

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

Part No: 819-1003

Copyright © 2004-2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

THIS PRODUCT CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF SUN MICROSYSTEMS, INC. USE, DISCLOSURE OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF SUN MICROSYSTEMS, INC.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, the Duke logo, the Java Coffee Cup logo, the Solaris logo, the SunTone Certified logo and the Sun ONE logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc.

Legato and the Legato logo are registered trademarks, and Legato NetWorker, are trademarks or registered trademarks of Legato Systems, Inc. The Netscape Communications Corp logo is a trademark or registered trademark of Netscape Communications Corporation.

The OPEN LOOK and Sun(TM) Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this service manual are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

---

Copyright © 2004-2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs des brevets américains listés à l'adresse <http://www.sun.com/patents> et un ou des brevets supplémentaires ou des applications de brevet en attente aux Etats - Unis et dans les autres pays.

CE PRODUIT CONTIENT DES INFORMATIONS CONFIDENTIELLES ET DES SECRETS COMMERCIAUX DE SUN MICROSYSTEMS, INC. SON UTILISATION, SA DIVULGATION ET SA REPRODUCTION SONT INTERDITES SANS L'AUTORISATION EXPRESSE, ECRITE ET PREALABLE DE SUN MICROSYSTEMS, INC.

Cette distribution peut comprendre des composants développés par des tierces parties.

Des parties de ce produit peuvent être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, le logo Duke, le logo Java Coffee Cup, le logo Solaris, le logo SunTone Certified et le logo Sun[tm] ONE sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

Legato, le logo Legato, et Legato NetWorker sont des marques de fabrique ou des marques déposées de Legato Systems, Inc. Le logo Netscape Communications Corp est une marque de fabrique ou une marque déposée de Netscape Communications Corporation.

L'interface d'utilisation graphique OPEN LOOK et Sun(TM) a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de ce manuel d'entretien et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes biologiques et chimiques ou du nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

# Quick Start Guide

Welcome to the *Sun Java™ System Application Server Enterprise Edition 8.1 2005Q1 Quick Start Guide*. This guide is for developers, system administrators, and Application Server administrators who are interested in learning about the capabilities of the Sun Java™ System Application Server Enterprise Edition 8.1 2005Q1 software.

This guide describes basic and advanced steps for using Application Server. The steps are presented in the order that you should complete them. The basic steps, which usually require less than 45 minutes to complete, are in these sections:

- [Starting the Domain Administration Server](#)
- [Creating a Cluster](#)
- [Deploying an Application](#)

The advanced steps, which usually require about 45 minutes to complete, are in these sections:

- [Setting up Load Balancing](#)
- [Setting Up High Availability Failover](#)

The final sections of this guide include instructions for [Cleaning Up](#) and information about [Where to Go Next](#).

The following table describes what the variable names and default paths are for the directories used in this guide. Variable names are in the first column, and default paths are in the second column.

Variable Name	Description and Path
<i>install_dir</i>	<p>By default, the Application Server installation directory is located here:</p> <ul style="list-style-type: none"> <li>• Solaris Sun Java™ Enterprise System installations: <i>/opt/SUNWappserver/appserver</i></li> <li>• Linux Java Enterprise System installations: <i>/opt/sun/appserver/</i></li> <li>• Solaris and Linux standalone Application Server installations, non-root user: <i>user_home_directory/SUNWappserver</i></li> <li>• Solaris and Linux standalone Application Server installations, root user: <i>/opt/SUNWappserver</i></li> </ul>
<i>domain_root_dir</i>	<p>By default, the directory containing all domains is located here:</p> <ul style="list-style-type: none"> <li>• Solaris Java Enterprise System installations: <i>/var/opt/SUNWappserver/domains/</i></li> <li>• Linux Java Enterprise System installations: <i>/var/opt/sun/appserver/domains/</i></li> <li>• All other installations: <i>install_dir/domains/</i></li> </ul>
<i>domain_dir</i>	<p>By default, domain directories are located here: <i>domain_root_dir/domain_dir</i></p>

## About Application Server Administration

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)
- The *asadmin* utility, a command-line tool
- 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 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 Domain Administration Server is installed, a default domain called *domain1* is always installed. You work with the default domain in this guide.

To complete most of the steps presented in this guide, you will use the Admin Console.

## Starting the Domain Administration Server

This topic, the first of three basic topics, provides the following steps:

- [Starting the Admin Server](#)
- [Logging in to the Admin Console](#)
- [Examining the Log File](#)

## Starting the Admin Server

To start the admin server, follow the procedure below.

1. Add the *install\_dir/bin/* directory to the PATH environment variable:

C Shell:

```
setenv PATH install_dir/bin:$PATH
```

Bourne Shell:

```
PATH=install_dir/bin:$PATH
export PATH
```

2. Set the admin user environment variable so that you do not need to type it for every command:

C Shell:

```
setenv AS_ADMIN_USER admin_user
```

Bourne Shell:

```
AS_ADMIN_USER=admin_user
export AS_ADMIN_USER
```

3. Start the server by entering this command from the *install\_dir*:

```
asadmin start-domain domain1
```

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

4. A message appears telling you that the Domain Administration Server is starting:

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

5. When the startup process has completed, you see an additional message:  
Domain domain1 started

---

**NOTE** Because this guide instructs you to set an environment variable `AS_ADMIN_USER` for administrative user name, it does not supply a user name argument when the `asadmin` command is used. You can supply this argument when you type the command. For example:

```
asadmin command_verb --user username command_arguments
```

---

## Logging in 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
- Configure clusters and node agents
- Manage server instances and clusters
- Select and view log files

For further information about using the Admin Console, consult the online help or the *Sun Java System Application Server Enterprise Edition 8.1 2005Q1 Administration Guide*.

To log in to the Admin Console:

1. Type this URL in your browser:

`https://localhost:4849/asadmin`

Because the Admin Console is a secure web application, you must use https instead of http.

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

4849 is the Admin Console's default port number. If you changed the port number during the installation, use that number instead.

---

**NOTE** If a popup window appears with a message such as Website Certified by an Unknown Authority, click OK.

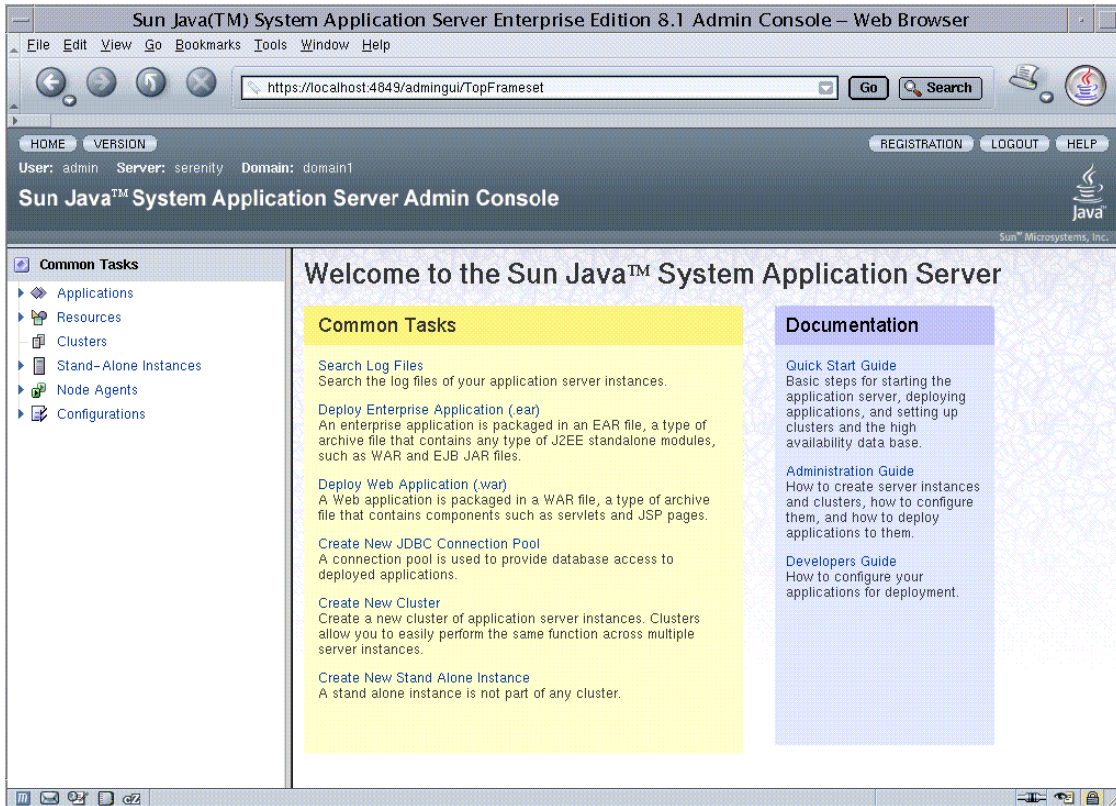
This message appears because your browser does not recognize the self-signed certificate that the Domain Administration Server uses to service the Admin Console over the secure transmission protocol.

---

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

### 3. Click Log In.

When the Admin Console appears, it looks like this:



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

---

**TIP** Click the Registration tab to register your software if you have not already done so.

---



## Examining the Log File

Application Server instances and the Domain Admin Server produce annotated logs on the file system. By default, all errors, warnings or useful informative messages are logged. To look at the Admin Server Log File:

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



**Log Viewer** Search Close

\* Indicates required field

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**

[Advanced Search](#)

\* Instance Name:  Log File:

Timestamp:  Most Recent  
 Specific Range:

Log Level:   Do not include more severe messages  
Use the Log Levels page to set the level for future logging. Requesting a level finer than the level specified on the Log Levels page will have no effect.

---

Search Close

**Search Results**

Log Viewer Results (0)				
Log Level	Message	Logger	Timestamp	Name-Value Pairs
No Matching Records. Try changing the search options or try next / previous buttons.				

2. In the Log Viewer window, select the 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 start-up.

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

For more information about the log file, see the *Sun Java System Application Server Enterprise Edition 8.1 2005Q1 Administration Guide*.

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 can stop the Quick Start trail here if you do not wish to continue.

## Creating a Cluster

This section, the second of three basic topics, explains how to create a cluster that contains two Application Server instances. For simplicity, the cluster runs completely within one machine. This topic provides the following steps:

- [Starting the Node Agent](#)
- [Defining the Cluster](#)

### Starting the Node Agent

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

One node agent is needed on a machine, for each Application Server administrative domain that the machine belongs to. If you chose the Node Agent Component during installation, a default node agent called *hostname* was created.

To start the default node agent:

1. In a terminal window, type this command:

```
asadmin start-node-agent hostname
```

Replace the variable *hostname* with the name of the host where the Application Server is running.

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

The node agent starts and connects with the Domain Administration Server. If the Domain Admin Server is not running, the node agent might fail to start.

## Defining the Cluster

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. In this guide, for simplicity, our sample cluster will be on one host, the same one where the Domain Administration Server is running.

You must have already started the Node Agent process on each machine, as described in previous section. When you specify instances during cluster creation, you must associate the instance with a running Node Agent for the machine on which you want the instance to run. Node Agent and instance names must be unique across clusters that are created in a domain.

To create a cluster:

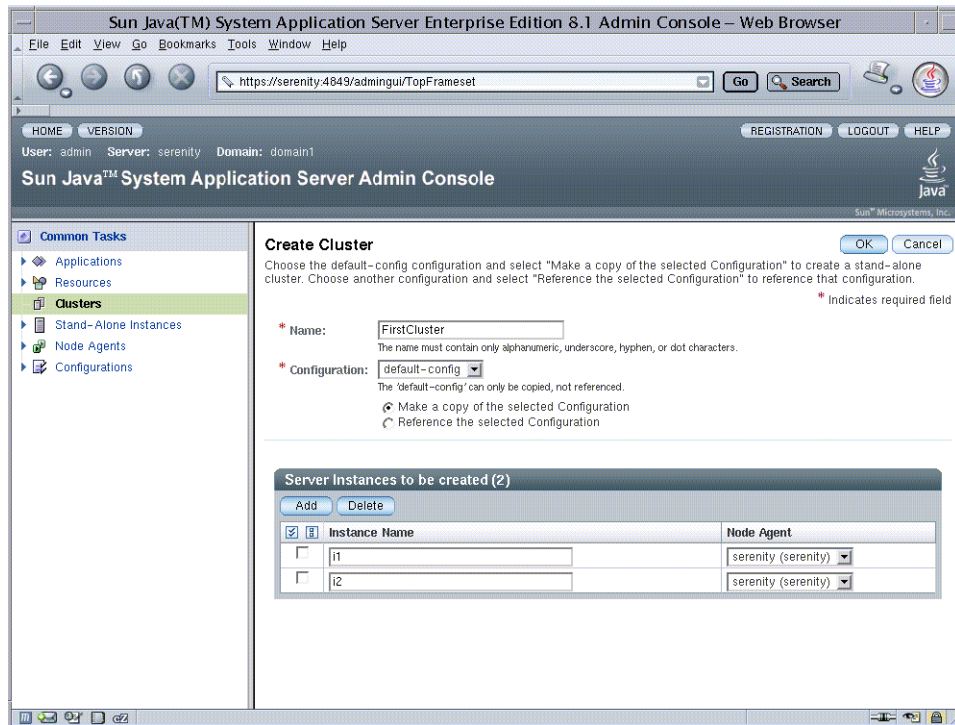
1. Log in to the Administration Console at `https://localhost:4849` if you have not already done so.

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

4849 is the Admin Console's default port number. If you changed the port number during the installation, use that number instead.

2. On the right pane, under Common Tasks, click Create New Cluster to display the Create Cluster input page.
3. Type **FirstCluster** 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 Add button twice, to create two entries to specify two instances for the cluster.
6. Type **i1** and **i2** as instance names. The node agent name is automatically populated with the name of the local machine.

You see a screen like this:



- Click OK. The create process can take a few minutes.

---

**NOTE** This exercise requires automatically assigned port numbers for HTTP, HTTPS, IIOP and IIOPS. You can change them later, if desired.

---

When the create process is completed, the Cluster Created Successfully page appears, and FirstCluster appears in the tree in the left pane. A copy of the configuration template default-config was made for this cluster, and the name FirstCluster-config was assigned to it.

- In the left pane, expand Clusters and click FirstCluster to display the General Information page for clusters.
- Click the Instances tab to display i1 and i2, the instances that you created.
  - Click i1 to examine this instance.

- b. From the tabs above the General Information heading of the right pane, click Properties and see the value for `HTTP_LISTENER_PORT`.
- c. Repeat these steps for `i2`.

---

**NOTE** By default, the HTTP ports are 38081 for `i1` and 38080 for `i2`. 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.

---

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 the Node Agent running on each machine).

You can stop the Quick Start trail here if you do not wish to continue.

## Deploying an Application

This section, the third of three basic topics, presents the following steps:

- [Deploying the Sample Application](#)
- [Verifying the Clustered Application](#)

### Deploying the Sample Application

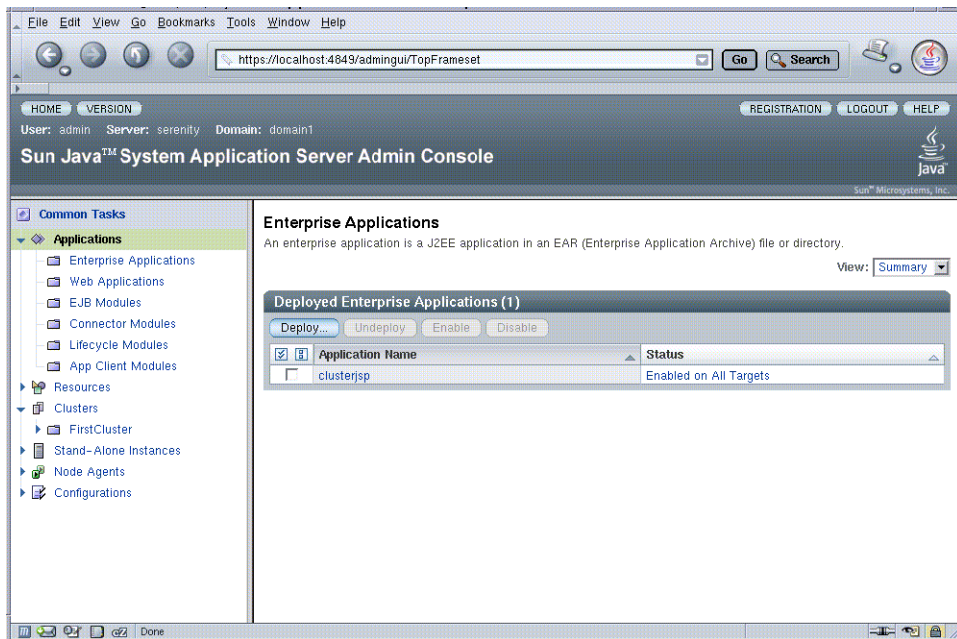
This guide uses the `clusterjsp` sample application to demonstrate web path load balancing capabilities.

1. Click the Home button 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.
3. In the File to Upload text box, click Browse, and navigate to `install_dir/samples/ee-samples/highavailability/apps/clusterjsp/clusterjsp.ear`.
4. Click Next to display the Deploy Enterprise Application page.
5. Scroll down to the Targets section of the page.

6. Select FirstCluster from the Available list, and click Add to move it to the Selected list.
7. Click OK.

The clusterjsp application is now deployed to FirstCluster.

This screen confirms that the clusterjsp application is now deployed to FirstCluster.



## Verifying the Clustered Application

In this step, you start the cluster and check whether the application is accessible on both the instances.

To start the cluster:

1. In the tree on the left pane, click the FirstCluster node under Clusters.
2. In the right pane, click the General tab if it is not already active.
3. Click the Start Instances button to start the cluster.

4. Verify that it has started by checking that the Status field changes to indicates that what instances are running.

To access the application on each instance:

1. Type this 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 i1. This example uses `http://localhost:38081/clusterjsp`.

2. Add some session attribute data.
3. Examine the Session and Host information displayed. For example:
  - o Executed From Server: *localhost*
  - o Server Port Number: 38081
  - o Executed Server IP Address: 192.18.145.133
  - o Session Created: Day Mon 05 14:55:34 PDT 2005
4. Add some session data and click the Add to Session button.
5. Repeat this for instance i2 by typing this URL in your browser:

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

In this section you deployed an application to a cluster and tested that the application is available on all instances in the cluster.

Congratulations! You have completed the basic steps in this Quick Start Guide. You can stop the Quick Start trail here if you do not wish to proceed to the advanced steps.

## Setting up Load Balancing

A load balancer is typically deployed in front of a cluster. It:

- 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 replication

- Enhances security by hiding the internal network from the user

Application Server includes load balancing plugins for popular web servers like Apache, and Sun Java™ System Web Server.

This section provides instructions on how to download and set up the Sun Java System Web Server software to act as a load balancer to the cluster of Application Servers. 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 to complete this section.

This topic presents the following steps:

- [Installing Web Server Software](#)
- [Installing the Load Balancer Plugin](#)
- [Creating a Load Balancer Configuration](#)
- [Starting the Load Balancer](#)
- [Verifying Load Balancing](#)

## Installing Web Server Software

If you already have Web Server software installed, and if you can identify a Web Server instance to serve as the load balancer, note the location of this instance in the file system and skip to [Installing the Load Balancer Plugin](#).

If you are using Sun Java Enterprise System software, the Web Server is selected for installation automatically when you choose to install the Load Balancer Plugin. However, you can also install Web Server software using these steps:

1. Launch the Java Enterprise System installer.
2. Select the Sun Java System Web Server component in the Component Selection page.
3. Choose “Configure Now” to be prompted for Web Server Configuration during installation.
4. When you are prompted, define a default Web Server instance.



---

**TIP** Remember the port number you choose for this default web server instance. This guide assumes that port 38000 is selected as the HTTP port for default instance.

---

If you are using a stand-alone Sun Java System Application Server, or if you do not have access to the Sun Java Enterprise System installer, install Web Server software using these steps:

1. Go to <http://www.sun.com/downloads>. Scroll down to the Web & Proxy Servers heading and click Web Servers.
2. Download Web Server 6.1 Service Pack 2 or higher for the locale and platform of your choice.

To download, you must log in with username and password registered with MySun<sup>SM</sup>, Sun Store<sup>SM</sup>, SunSolve<sup>SM</sup>, or the Online Support Center. If you do not have a login account, you can register online.

3. Follow the instructions to install Web Server software. You must:
  - a. Extract the software from the compressed archive.
  - b. Run the setup program. If you need additional information, see the Web Server installation instructions at <http://docs.sun.com/doc/819-0131-10>.
4. The Web Server installation process configures the Administration Server for Web Server and prompts you to define a default Web Server instance.

---

**TIP** Remember the port number you choose for this default web server instance. This guide assumes that port 38000 is selected as the HTTP port for default instance.

---

## Installing the Load Balancer Plugin

To install the load balancer plugin:

1. Run the installer for the software distribution you are using—Sun Java Enterprise System software or the stand-alone Sun Java System Application Server software.

2. When you are asked which components you want to install, select Load Balancing Plugin.

On the Sun Java Enterprise System installer, you must expand the Application Server item to see the Load Balancing Plugin. It is not selected for installation by default.

## Creating a Load Balancer Configuration

Now return to interacting with the Application Server's Domain Admin Server. You need a shell execution environment for this section.

1. Create a load balancer configuration called `MyLbConfig` targeted to the cluster `FirstCluster`:

```
asadmin create-http-lb-config --target FirstCluster MyLbConfig
```

2. Enable the `FirstCluster` cluster and the `clusterjsp` application deployed in it for HTTP load balancing:

```
asadmin enable-http-lb-server FirstCluster
```

```
asadmin enable-http-lb-application --name clusterjsp FirstCluster
```

3. Create a health checker for the load balancer, which signals when an instance that goes down recovers.

```
asadmin create-http-health-checker --interval 10 --config MyLbConfig FirstCluster
```

The interval is the number of seconds the health checker waits between checks of an unhealthy instance.

4. Export the configuration to a file `loadbalancer.xml`

```
asadmin export-http-lb-config --config MyLbConfig loadbalancer.xml
```

5. Copy `loadbalancer.xml` to `web_server_install_dir/https-hostname/config/loadbalancer.xml`

## Starting the Load Balancer

If the Web Server instance serving as load balancer is not already running, start the Web Server software by executing this command:

```
web_server_install_dir/https-hostname/start
```

If the Web Server instance serving as load balancer is not already running, restart the Web Server software by executing this command

```
web_server_install_dir/https-hostname/restart
```

## Verifying 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 the [Verifying the Clustered Application](#) section appears.

2. Examine the Session and Host information displayed. For example:
  - o Executed From Server: *localhost*
  - o Server Port Number: 38000
  - o Executed Server IP Address: 192.18.145.133
  - o 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:

- o 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
  - **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
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

In this section you created an instance to use as a load balancer and set up a load balancing configuration. You also verified load balancing.

To configure and verify HTTP session failover, continue to [Setting Up High Availability Failover](#).

## Cleaning Up

To clean up, you can uninstall the Application Server installation by completing the steps in [Option 1: Uninstalling Your Installation](#), or you can simply delete the sample cluster you have just created by completing the steps in [Option 2: Removing the Sample Cluster](#).

---

**CAUTION** If you plan to complete the [About High Availability Clusters and HADB](#) section, do not clean up the installation now. Instead, start the [About High Availability Clusters and HADB](#) section, and when you are done clean up using the procedures you’ll find there.

---

## Option 1: Uninstalling Your Installation

To uninstall completely:

1. Stop the Application Server processes using these commands:

```
asadmin stop-cluster FirstCluster
```

```
asadmin stop-node-agent hostname
```

```
asadmin stop-domain domain1
```

At this point all processes related to Application Server are stopped.

2. Uninstall the Application Server.

In a Java Enterprise System installation, run

`var/sadm/prod/entsys/uninstall` and follow the steps in the uninstallation wizard.

In a standalone Application Server installation, run `install_dir/uninstall` and follow the steps in the uninstallation wizard.

3. If you installed Web Server for this exercise, stop the web server instance acting as load balancer and uninstall the Web Server product. You can stop the instance as follows:

```
web_server_install_dir/https-hostname/stop
```

4. If you wish to uninstall the Web Server product, run the uninstall program from the `web_server_install_dir`.

## Option 2: Removing the Sample Cluster

To remove only the FirstCluster (the sample highly available cluster) and the sample application used during this exercise:

1. Stop the Application Server processes and clean up configuration:

```
asadmin stop-cluster FirstCluster  
  
asadmin disable-http-lb-server FirstCluster  
  
asadmin delete-http-lb-ref --config MyLbConfig FirstCluster  
  
asadmin delete-http-lb-config MyLbConfig  
  
asadmin delete-instance i1  
  
asadmin delete-instance i2  
  
asadmin delete-cluster FirstCluster  
  
asadmin undeploy clusterjsp
```

2. Stop the web server instance acting as load balancer:

```
web_server_install_dir/https-hostname/stop
```

3. Rename the loadbalancer.xml file in

```
web_server_install_dir/https-hostname/config to loadbalancer.xml.sav
```

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

## Where to Go Next

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

- Product details at *install\_dir/docs-ee/about.html*.  
See this document for the latest information on what is new, and pointers to tutorials and other educational services.
- The *Sun Java System Application Server Release Notes*  
See this document for late-breaking information regarding this release.
- The *Sun Java System Application Server Administration Guide*

See this document for information on performing administrative functions using the Admin Console.

- *The Utility Reference* (man pages)

See this document for reference information on Application Server's command-line utilities, such as `asadmin`.

- *The Sun Java System Application Server High Availability Administration Guide*

See this document for information on the Sun Java System Application Server's high availability features.

- *The J2EE 1.4 Tutorial*

See this document for a tutorial that covers the process for building and deploying Java™ 2 Platform, Enterprise Edition (J2EE™ platform) applications.

- Java™ BluePrints guidelines for the Enterprise

See this document for a comprehensive set of examples that demonstrate operations of the Application Server software and that can be used as application templates.





# 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.

Application Server supports both HTTP session persistence and persistence for Stateful Session Beans. The procedures in this chapter cover HTTP session persistence.

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.

---

**NOTE**      Completing this section may require additional hardware resources.

---

## About High Availability Clusters and HADB

A highly availability cluster in Sun Java System Application Server Enterprise Edition 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™ 2 Platform, Enterprise Edition (J2EE™ 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 the *High Availability Administration Guide*.

1. Get root access.
2. Define variables related to shared memory.

### 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=20
set semsys:seminfo_semmni=16
set semsys:seminfo_semmns=128
set semsys:seminfo_semmnu=1000
```

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.

- 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 512 Mbytes of physical memory:

```
kernel.shmmax=536870912
```

```
kernel.shmall=536870912
```

- b. Reboot, using this command:

```
sync; sync; reboot
```

3. If you used existing JDK software when you installed a standalone Application Server, HADB requires Sun JDK 1.4.1\_03 or higher (for the latest information on JDK versions, see the *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, use the command

```
web_server_install_dir/https-hostname/start.
```

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](#).

---

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. In the Component Selection page, choose the option to install HADB.
3. Complete the installation on your hosts.

## Starting HADB

This section describes starting the HADB management agent 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 the *High Availability Administration Guide*.

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

In a Java Enterprise System installation:

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

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

2. Run the command to start the agent:

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

In a standalone Application Server installation:

1. Change to the HADB `bin` directory in the Application Server installation:

```
cd install_dir/hadb/4/bin
```

2. Run the command to start the agent:

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

# Configuring a Cluster and Application for High Availability

1. From the machine on which the Domain Administration Server is running, configure FirstCluster to use HADB using this command:

```
asadmin configure-ha-cluster --hosts hadb_hostname, hadb_hostname
--devicesize 256 FirstCluster
```

Replace the *hadb\_hostname* variable with the host name of the machine where HADB is to run. If you are using just one machine, you must name it twice.

This simplified example runs two nodes of HADB on the same machine. In production settings, use more than one machine.

---

**NOTE** To reduce the memory footprint of HADB **for demonstration purposes**, execute the following `hadbm` command. You are prompted for the administration password.

In Java Enterprise System installations:

```
/opt/SUNWhadb/4/bin/hadbm set
DataBufferPoolSize=64,LogBufferSize=25 FirstCluster
```

In standalone Application Server installations:

```
install_dir/hadb/4/bin/hadbm set
DataBufferPoolSize=64,LogBufferSize=25 FirstCluster
```

---

2. Configure the clusterjsp application for HTTP session persistence by enabling high availability:
  - a. In the Admin Console, expand the Applications node
  - b. Expand Enterprise Applications.
  - c. Click clusterjsp.
  - d. In the right pane, on the General tab, click the Availability Enabled checkbox.
  - e. Click Save.

Availability is enabled at the server instance and container level by default.

## Restarting the Cluster

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 [Verifying Load Balancing](#) topic. 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.

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 the [Verifying the Clustered Application](#) section appears.

2. Examine the Session and Host information displayed. For example:
  - o Executed From Server: *localhost*
  - o Server Port Number: 38000
  - o Executed Server IP Address: 192.18.145.133
  - o Session ID: 41880f618e4593e14fb5d0ac434b1
  - o Session Created: Wed Feb 23 15:23:18 PST 2005

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
- **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

3. 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.
4. Reload the clusterjsp sample application page.  
     The session ID and session attribute data is retained.
5. 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™ enterprise beans in the HADB.

# Cleaning Up

To clean up, you can uninstall the Application Server installation by completing the steps in [Option 1: Uninstalling Your Installation](#), or you can simply delete the sample cluster you have just created by completing the steps in [Option 2: Removing the Sample Cluster](#).

## Option 1: Uninstalling Your Installation

To uninstall completely:

1. Stop the Application Server processes using these commands:

```
asadmin stop-cluster FirstCluster
```

```
asadmin remove-ha-cluster --hosts hadb_hostname,hadb_hostname
FirstCluster
```

Replace the *hadb\_hostname* variable with the host name of the machine where HADB is to run. If you are using just one machine, you must name it twice.

```
asadmin stop-node-agent hostname
```

```
asadmin stop-domain domain1
```

2. Stop the HADB Management Agent by one of the following methods:

In a Java Enterprise System installation:

- a. Change to the `/etc/init.d` directory:

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

- b. Run the command to stop the agent:

```
./ma-initd stop
```

In a standalone Application Server installation:

- a. Change to the HADB `bin` directory in the Application Server installation:

```
cd install_dir/hadb/4/bin
```





1. Stop the Application Server processes and clean up configuration:

```
asadmin stop-cluster FirstCluster
```

```
asadmin remove-ha-cluster --hosts hadb_hostname,hadb_hostname  
FirstCluster
```

Replace the *hadb\_hostname* variable with the host name of the machine where HADB is to run. If you are using just one machine, you must name it twice.

```
asadmin disable-http-lb-server FirstCluster
```

```
asadmin delete-http-lb-ref --config MyLbConfig FirstCluster
```

```
asadmin delete-http-lb-config MyLbConfig
```

```
asadmin delete-instance i1
```

```
asadmin delete-instance i2
```

```
asadmin delete-cluster FirstCluster
```

```
asadmin undeploy clusterjsp
```

2. Stop the web server instance acting as load balancer:

```
web_server_install_dir/https-hostname/stop
```

3. Rename the loadbalancer.xml file in

```
web_server_install_dir/https-hostname/config to loadbalancer.xml.sav
```

4. Stop the HADB Management Agent by one of the following methods:

In a Java Enterprise System installation:

- a. Change to the `/etc/init.d` directory:

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

- b. Run the command to stop the agent:

```
./ma-initd stop
```

In a standalone Application Server installation:

- a. Change to the HADB `bin` directory in the Application Server installation:

```
cd install_dir/hadb/4/bin
```

- b. Run the command to stop the agent:

```
./ma-initd stop
```

In this section, you have installed, configured, and started HADB and configured a cluster and an application to use high availability. You have also cleaned up so that your system is ready for other work.

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

## Where to Go Next

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

- Product details at *install\_dir/docs-ee/about.html*.  
See this document for the latest information on what is new, and pointers to tutorials and other educational services.
- *The Sun Java System Application Server Release Notes*  
See this document for late-breaking information regarding this release.
- *The Sun Java System Application Server Administration Guide*  
See this document for information on performing administrative functions using the Admin Console.
- *The Utility Reference* (man pages)  
See this document for reference information on Application Server's command-line utilities, such as `asadmin`.
- *The Sun Java System Application Server High Availability Administration Guide*  
See this document for information on the Sun Java System Application Server's high availability features.
- *The J2EE 1.4 Tutorial*  
See this document for a tutorial that covers the process for building and deploying Java™ 2 Platform, Enterprise Edition (J2EE™ platform) applications.
- Java™ BluePrints guidelines for the Enterprise  
See this document for a comprehensive set of examples that demonstrate operations of the Application Server software and that can be used as application templates.

Where to Go Next