



Sun Java™ System

Application Server 7

Administrator's Configuration File

Reference

---

2004Q2

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

Part No: 817-5446

Copyright © 2004 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.

Use is subject to license terms. This distribution may include materials developed by third parties.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, Sun™ ONE, Sun™ ONE Studio, iPlanet, J2EE, J2SE, Enterprise JavaBeans, EJB, JavaServer Pages, JSP, JDBC, JDK, JVM, Java Naming and Directory Interface, JavaMail, and the Java Coffee Cup 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.

UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

This product is covered and 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 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 plus des brevets américains listés à l'adresse <http://www.sun.com/patents> et un ou les brevets supplémentaires ou les 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.

L'utilisation est soumise aux termes de la Licence. Cette distribution peut comprendre des composants développés par des tierces parties.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, Sun™ ONE, Sun™ ONE Studio, iPlanet, J2EE, J2SE, Enterprise JavaBeans, EJB, JavaServer Pages, JSP, JDBC, JDK, JVM, Java Naming and Directory Interface, JavaMail, et le logo Java Coffee Cup 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.

UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Ce produit est soumis à la législation américaine en matière de contrôle des exportations et peut être soumis à la réglementation en vigueur dans d'autres pays dans le domaine des exportations et importations. Les utilisations, 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 les pays sous embargo américain, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exhaustive, 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.

# Contents

<b>About This Book</b>	<b>9</b>
Who Should Use This Guide	9
Using the Documentation	10
How This Guide Is Organized	12
Documentation Conventions	13
General Conventions	13
Conventions Referring to Directories	14
Contacting Sun	15
Give Us Feedback	15
Obtain Training	15
Contact Product Support	15
 <b>Chapter 1 Basics of Server Operation</b>	 <b>17</b>
Configuration Files	17
server.xml	18
init.conf	18
obj.conf	18
mime.types	19
Dynamic Reconfiguration	19
Manually Editing Configuration Files	20
Reconfiguring a Cluster	21
 <b>Chapter 2 Server Configuration Files</b>	 <b>23</b>
The sun-server_1_1.dtd File	24
Subelements	25
Data	25
Attributes	26
Elements in the server.xml File	26
General Elements	27
server	27

property .....	28
description .....	29
admin-service .....	30
server-instance .....	30
Listener Service Elements .....	30
http-service .....	30
http-listener .....	32
ssl .....	33
mime .....	35
acl .....	35
virtual-server-class .....	36
virtual-server .....	37
http-qos .....	41
auth-db .....	42
iiop-service .....	42
orb .....	43
ssl-client-config .....	44
iiop-listener .....	44
Container Elements .....	45
web-container .....	46
ejb-container .....	46
mdb-container .....	50
Availability Elements .....	51
availability-service .....	52
iiop-cluster .....	53
iiop-server-instance .....	54
iiop-endpoint .....	54
persistence-store .....	55
session-config .....	55
session-manager .....	56
manager-properties .....	57
store-properties .....	58
session-properties .....	60
J2EE Service Elements .....	61
jms-service .....	61
log-service .....	63
security-service .....	64
auth-realm .....	66
transaction-service .....	68
Java Configuration Elements .....	69
java-config .....	69
profiler .....	71
jvm-options .....	72

Resource Elements	73
resources	73
custom-resource	74
external-jndi-resource	75
jdbc-resource	76
mail-resource	76
jms-resource	78
persistence-manager-factory-resource	80
jdbc-connection-pool	80
Application Elements	84
applications	84
lifecycle-module	85
j2ee-application	86
ejb-module	87
web-module	88
connector-module	89
User Database Selection	90
The Sun Java System LDAP Schema	91
The Convergence Tree	91
The Domain Component (dc)Tree	92
Variables	92
Format of a Variable	93
The id Variable	93
Other Important Variables	93
General Variables	93
send-cgi Variables	94
Variable Evaluation	94
Sample server.xml File	95
 <b>Chapter 3 Syntax and Use of init.conf</b>	 <b>101</b>
Init Functions	102
Server Information	103
NetSiteRoot	103
TempDir	103
TempDirSecurity	103
User (UNIX only)	104
DNS Lookup	104
AsyncDNS	105
DNS	105
Threads, Processes and Connections	105
ConnQueueSize	107
HeaderBufferSize	107
IOTimeout	107

KeepAliveThreads	107
KeepAliveTimeout	108
ListenQ	108
MaxKeepAliveConnections	108
PostThreadsEarly	108
RcvBufSize	109
RqThrottle	109
RqThrottleMin	109
SndBufSize	109
StackSize	109
StrictHttpHeaders	110
TerminateTimeout	110
ThreadIncrement	110
Native Thread Pools	110
NativePoolStackSize	111
NativePoolMaxThreads	111
NativePoolMinThreads	111
NativePoolQueueSize	111
CGI	112
CGIExpirationTimeout	112
CGIStubIdleTimeout	112
MaxCGIStubs	113
MinCGIStubs	113
Error Logging	113
ErrorLogDateFormat	113
LogFlushInterval	114
PidLog	114
ACL	114
ACLCacheLifetime	115
ACLUserCacheSize	115
ACLGroupCacheSize	115
Security	115
Security	116
SSLCacheEntries	116
SSLClientAuthDataLimit	116
SSLClientAuthTimeout	117
SSLSessionTimeout	117
SSL3SessionTimeout	117
Chunked Encoding	117
UseOutputStreamSize	118
ChunkedRequestBufferSize	118
ChunkedRequestTimeout	118
Miscellaneous	119

ChildRestartCallback .....	119
HTTPVersion .....	119
MaxRqHeaders .....	119
ReentrantTimeFunctions (Solaris only) .....	120
Umask (UNIX only) .....	120
 <b>Chapter 4 MIME Types .....</b>	<b>121</b>
Introduction .....	121
Determining the MIME Type .....	122
How the Type Affects the Response .....	122
What Does the Client Do with the MIME Type? .....	123
Syntax of the MIME Types File .....	123
Sample MIME Types File .....	124
 <b>Chapter 5 Other Configuration Files .....</b>	<b>125</b>
dbswitch.conf .....	125
Deployment Descriptors .....	127
generated.instance.acl .....	128
loadbalancer.xml .....	129
nsfc.conf .....	131
password.conf .....	133
server.policy .....	133
 <b>Appendix A Time Formats .....</b>	<b>135</b>
 <b>Appendix B Alphabetical List of Server Configuration Elements .....</b>	<b>137</b>
 <b>Appendix C Alphabetical List of Directives in init.conf .....</b>	<b>141</b>
 <b>Index .....</b>	<b>145</b>





# About This Book

The book discusses the purpose and use of the configuration files for Sun Java™ System Application Server 7, Enterprise Edition, including `server.xml`, `init.conf`, and `mime.types`, and provides comprehensive lists of the elements and directives in these configuration files.

This preface contains information about the following topics:

- [Who Should Use This Guide](#)
- [Using the Documentation](#)
- [How This Guide Is Organized](#)
- [Documentation Conventions](#)
- [Contacting Sun](#)

## Who Should Use This Guide

The intended audience for this guide is the person who administers and maintains the Sun Java System Application Server.

This guide assumes you are familiar with the following topics:

- Java™ 2 Platform, Enterprise Edition (J2EE™ platform) specification
- HTTP
- HTML
- XML
- Java programming

- Java APIs as defined in the Java™ Servlet, JavaServer Pages™ (JSP™), Enterprise JavaBeans™ (EJB™), and Java™ Database Connectivity (JDBC™) specifications
- Relational database concepts

# Using the Documentation

The Sun Java System Application Server Standard and Enterprise Edition manuals are available as online files in Portable Document Format (PDF) and Hypertext Markup Language (HTML).

The following table lists tasks and concepts described in the Sun Java System Application Server manuals. The manuals marked *(updated for 7 2004Q2)* have been updated for the Sun Java System Application Server Standard and Enterprise Edition 7 2004Q2 release. The manuals not marked in this way have not been updated since the version 7 Enterprise Edition release.

**Table 1** Sun Java System Application Server Documentation Roadmap

For information about	See the following
<i>(Updated for 7 2004Q2)</i> Late-breaking information about the software and the documentation. Includes a comprehensive, table-based summary of supported hardware, operating system, JDK, and JDBC/RDBMS.	<i>Release Notes</i>
Sun Java System Application Server 7 overview, including the features available with each product edition.	<i>Product Overview</i>
Diagrams and descriptions of server architecture and the benefits of the Sun Java System Application Server architectural approach.	<i>Server Architecture</i>
<i>(Updated for 7 2004Q2)</i> How to get started with the Sun Java System Application Server product. Includes a sample application tutorial. There are two guides, one for Standard Edition and one for Enterprise Edition.	<i>Getting Started Guide</i>
<i>(Updated for 7 2004Q2)</i> Installing the Sun Java System Application Server Standard Edition and Enterprise Edition software and its components, such as sample applications and the Administration interface. For the Enterprise Edition software, instructions are provided for implementing the high-availability configuration.	<i>Installation Guide</i>
<i>(Updated for 7 2004Q2)</i> Evaluating your system needs and enterprise to ensure that you deploy Sun Java System Application Server in a manner that best suits your site. General issues and concerns that you must be aware of when deploying an application server are also discussed.	<i>System Deployment Guide</i>

**Table 1** Sun Java System Application Server Documentation Roadmap (*Continued*)

For information about	See the following
Creating and implementing Java™ 2 Platform, Enterprise Edition (J2EE™ platform) applications intended to run on the Sun Java System Application Server that follow the open Java standards model for J2EE components such as servlets, Enterprise JavaBeans™ (EJBs™), and JavaServer Pages™ (JSPs™). Includes general information about application design, developer tools, security, assembly, deployment, debugging, and creating lifecycle modules. A comprehensive Sun Java System Application Server glossary is included.	<i>Developer's Guide</i>
( <i>Updated for 7 2004Q2</i> ) Creating and implementing J2EE web applications that follow the Java™ Servlet and JavaServer Pages (JSP) specifications on the Sun Java System Application Server. Discusses web application programming concepts and tasks, and provides sample code, implementation tips, and reference material. Topics include results caching, JSP precompilation, session management, security, deployment, SHTML, and CGI.	<i>Developer's Guide to Web Applications</i>
( <i>Updated for 7 2004Q2</i> ) Creating and implementing J2EE applications that follow the open Java standards model for enterprise beans on the Sun Java System Application Server. Discusses Enterprise JavaBeans (EJB) programming concepts and tasks, and provides sample code, implementation tips, and reference material. Topics include container-managed persistence, read-only beans, and the XML and DTD files associated with enterprise beans.	<i>Developer's Guide to Enterprise JavaBeans Technology</i>
( <i>Updated for 7 2004Q2</i> ) Creating Application Client Container (ACC) clients that access J2EE applications on the Sun Java System Application Server.	<i>Developer's Guide to Clients</i>
Creating web services in the Sun Java System Application Server environment.	<i>Developer's Guide to Web Services</i>
( <i>Updated for 7 2004Q2</i> ) Java™ Database Connectivity (JDBC™), transaction, Java Naming and Directory Interface™ (JNDI), Java™ Message Service (JMS), and JavaMail™ APIs.	<i>Developer's Guide to J2EE Services and APIs</i>
Creating custom NSAPI plug-ins.	<i>Developer's Guide to NSAPI</i>
( <i>Updated for 7 2004Q2</i> ) Information and instructions on the configuration, management, and deployment of the Sun Java System Application Server subsystems and components, from both the Administration interface and the command-line interface. Topics include cluster management, the high-availability database, load balancing, and session persistence. A comprehensive Sun Java System Application Server glossary is included.	<i>Administration Guide</i>
( <i>Updated for 7 2004Q2</i> ) Editing Sun Java System Application Server configuration files, such as the <code>server.xml</code> file.	<i>Administrator's Configuration File Reference</i>
Configuring and administering security for the Sun Java System Application Server operational environment. Includes information on general security, certificates, and SSL/TLS encryption. HTTP server-based security is also addressed.	<i>Administrator's Guide to Security</i>

**Table 1** Sun Java System Application Server Documentation Roadmap (*Continued*)

For information about	See the following
Configuring and administering service provider implementation for J2EE™ Connector Architecture (CA) connectors for the Sun Java System Application Server. Topics include the Administration Tool, Pooling Monitor, deploying a JCA connector, and sample connectors and sample applications.	<i>J2EE CA Service Provider Implementation Administrator's Guide</i>
(Updated for 7 2004Q2) Migrating your applications to the new Sun Java System Application Server programming model, specifically from iPlanet Application Server 6.x and Sun ONE Application Server 7.0. Includes a sample migration.	<i>Migrating and Redeploying Server Applications Guide</i>
(Updated for 7 2004Q2) How and why to tune your Sun Java System Application Server to improve performance.	<i>Performance Tuning Guide</i>
(Updated for 7 2004Q2) Information on solving Sun Java System Application Server problems.	<i>Troubleshooting Guide</i>
(Updated for 7 2004Q2) Information on solving Sun Java System Application Server error messages.	<i>Error Message Reference</i>
(Updated for 7 2004Q2) Utility commands available with the Sun Java System Application Server; written in manpage style.	<i>Utility Reference Manual</i>
Using the Sun™ Java System Message Queue 3.5 software.	The Sun Java System Message Queue documentation at: <a href="http://docs.sun.com/db?p=prod/s1.s1msgqu">http://docs.sun.com/db?p=prod/s1.s1msgqu</a>

# How This Guide Is Organized

This book has the following chapters and appendixes:

- [Chapter 1, “Basics of Server Operation”](#)  
This chapter introduces the major configuration files that control the Sun Java System Application Server and describes how to activate and edit them.
- [Chapter 2, “Server Configuration Files”](#)  
This chapter discusses the `server.xml` file, which controls most aspects of server operation.
- [Chapter 3, “Syntax and Use of `init.conf`”](#)  
This chapter discusses the directives you can set in the `init.conf` file to configure the Sun Java System Application Server during initialization.

- [Chapter 4, “MIME Types”](#)

This chapter discusses the MIME types file, which maps file extensions to file types.

- [Chapter 5, “Other Configuration Files”](#)

This chapter lists other important configuration files and provides a quick reference of their contents.

- [Appendix A, “Time Formats”](#)

This appendix describes the format strings used for dates and times in the server log.

- [Appendix B, “Alphabetical List of Server Configuration Elements”](#)  
[Appendix C, “Alphabetical List of Directives in `init.conf`”](#)

These appendixes provide alphabetical lists for easy lookup of elements in `server.xml` and directives in `init.conf`.

## Documentation Conventions

This section describes the types of conventions used throughout this guide:

- [General Conventions](#)
- [Conventions Referring to Directories](#)

## General Conventions

The following general conventions are used in this guide:

- **File and directory paths** are given in UNIX® format (with forward slashes separating directory names). For Windows versions, the directory paths are the same, except that backslashes are used to separate directories.

- **URLs** are given in the format:

`http://server.domain/path/file.html`

In these URLs, *server* is the server name where applications are run; *domain* is your Internet domain name; *path* is the server’s directory structure; and *file* is an individual filename. Italic items in URLs are placeholders.

- **Font conventions** include:

- The monospace font is used for sample code and code listings, API and language elements (such as function names and class names), file names, pathnames, directory names, and HTML tags.
- *Italic* type is used for code variables.
- *Italic* type is also used for book titles, emphasis, variables and placeholders, and words used in the literal sense.
- **Bold** type is used as either a paragraph lead-in or to indicate words used in the literal sense.
- **Installation root directories** for most platforms are indicated by *install\_dir* in this document. Exceptions are noted in [“Conventions Referring to Directories” on page 14](#).

By default, the location of *install\_dir* on **most** platforms is:

- Solaris and Linux file-based installations:

*user's home directory/sun/appserver7*

- Windows, all installations:

*system drive:\Sun\AppServer7*

For the platforms listed above, *default\_config\_dir* and *install\_config\_dir* are identical to *install\_dir*. See [“Conventions Referring to Directories” on page 14](#) for exceptions and additional information.

- **Instance root directories** are indicated by *instance\_dir* in this document, which is an abbreviation for the following:

*default\_config\_dir/domains/domain/instance*

- **UNIX-specific descriptions** throughout this manual apply to the Linux operating system as well, except where Linux is specifically mentioned.

## Conventions Referring to Directories

By default, when using the Solaris package-based or Linux RPM-based installation, the application server files are spread across several root directories. This guide uses the following document conventions to correspond to the various default installation directories provided:

- *install\_dir* refers to */opt/SUNWappserver7*, which contains the static portion of the installation image. All utilities, executables, and libraries that make up the application server reside in this location.

- *default\_config\_dir* refers to `/var/opt/SUNWappserver7/domains`, which is the default location for any domains that are created.
- *install\_config\_dir* refers to `/etc/opt/SUNWappserver7/config`, which contains installation-wide configuration information such as licenses and the master list of administrative domains configured for this installation.

## Contacting Sun

You might want to contact Sun Microsystems in order to:

- [Give Us Feedback](#)
- [Obtain Training](#)
- [Contact Product Support](#)

### Give Us Feedback

If you have general feedback on the product or documentation, please send this to:

<http://www.sun.com/hwdocs/feedback/>

### Obtain Training

Application Server training courses are available at:

[http://training.sun.com/US/catalog/enterprise/web\\_application.html/](http://training.sun.com/US/catalog/enterprise/web_application.html/)

Visit this site often for new course availability on the Sun Java System Application Server.

### Contact Product Support

If you have problems with your system, contact customer support using one of the following mechanisms:

- The online support web site at:  
<http://www.sun.com/supporttraining/>
- The telephone dispatch number associated with your maintenance contract

Please have the following information available prior to contacting support. This helps to ensure that our support staff can best assist you in resolving problems:

- Description of the problem, including the situation where the problem occurs and its impact on your operation
- Machine type, operating system version, and product version, including any patches and other software that might be affecting the problem. Here are some of the commonly used commands:
  - **Solaris:** `pkginfo, showrev`
  - **Linux:** `rpm`
  - **All:** `asadmin version --verbose`
- Detailed steps on the methods you have used to reproduce the problem
- Any error logs or core dumps
- Configuration files such as:
  - `instance_dir/config/server.xml`
  - a web application's `web.xml` file, when a web application is involved in the problem
- For an application, whether the problem appears when it is running in a cluster or standalone



# Basics of Server Operation

The configuration and behavior of Sun Java™ System Application Server 7, Enterprise Edition, is determined by a set of configuration files. When you use the Administration interface or the command line, you change settings in these configuration files behind the scenes.

This chapter has the following sections:

- [Configuration Files](#)
- [Dynamic Reconfiguration](#)
- [Manually Editing Configuration Files](#)
- [Reconfiguring a Cluster](#)

## Configuration Files

The configuration and operation of the Sun Java System Application Server is controlled by configuration files. The configuration files reside in the directory *instance\_dir/config*. This directory contains various configuration files for controlling different components. The exact number and names of configuration files depends on which components have been enabled or loaded into the server.

However, this directory always contains four configuration files that are essential for the server to operate:

- [server.xml](#) - contains most of the server configuration.
- [init.conf](#) - contains global server initialization information.
- [obj.conf](#) - contains instructions for handling HTTP requests from clients.
- [mime.types](#) - contains information for determining the content type of requested resources.

For information about other important configuration files, such as `dbswitch.conf` and `loadbalancer.xml`, see [Chapter 5, “Other Configuration Files.”](#)

## server.xml

This file contains most of the server configuration. A schema file, `sun-server_1_1.dtd`, defines its format and content. For more information about how the server uses `sun-server_1_1.dtd` and `server.xml`, see [Chapter 2, “Server Configuration Files.”](#)

## init.conf

This file sets values of variables that configure the server during initialization. The server looks at this file and executes the settings on startup. The server does not look at this file again until it is restarted. For more information about how the server uses `init.conf`, see [Chapter 3, “Syntax and Use of init.conf.”](#)

## obj.conf

This file contains instructions for the server about how to process HTTP requests from clients and service web server content such as native server plug-ins and CGI programs. The file’s directives tell the Sun Java System Application Server what to do at each stage in the request-response process. The server looks at the configuration defined by this file every time it processes a request from a client.

All `obj.conf` files are located in the `instance_dir/config` directory. There is one `obj.conf` file for each virtual server, unless several virtual servers are configured to share an `obj.conf` file. Whenever this guide refers to “the `obj.conf` file,” it refers to all `obj.conf` files or to the `obj.conf` file for the virtual server being described.

The file named `obj.conf` that lacks a prefix is a template that Sun Java System Application Server uses to create `obj.conf` files for each virtual server. Editing this file does not affect any existing virtual servers, but does affect any subsequently created virtual servers.

By default, each active `obj.conf` file is named `virtual_server_name-obj.conf`. Because the default virtual server for a server instance is named after the instance, when you first create a server instance, its `obj.conf` file is named `instance_name-obj.conf`.

For detailed information about how the server uses `obj.conf`, see the *Sun Java System Application Server Developer’s Guide to NSAPI*.

## mime.types

This file maps file extensions to MIME types to enable the server to determine the content type of a requested resource. For example, requests for resources with `.html` extensions indicate that the client is requesting an HTML file, while requests for resources with `.gif` extensions indicate that the client is requesting an image file in GIF format. For more information about how the server uses `mime.types`, see [Chapter 4, “MIME Types.”](#)

## Dynamic Reconfiguration

Changes take effect immediately when you change the `applications` and `resources` elements of the `server.xml` file (for example, you deploy a Java™ 2 Platform, Enterprise Edition (J2EE™ platform) application or register a Java™ Database Connectivity (JDBC™) Connection Pool). You do not need to restart the server or apply the changes.

You can set up the `loadbalancer.xml` file so that changes to it take effect immediately by changing its `reload-poll-interval-in-seconds` property. For details, see [“loadbalancer.xml” on page 129.](#)

When you change the following files, you do not need to restart the server for the changes to take effect, but you must apply the changes:

- `obj.conf`
- `mime.types`
- ACL files
- `server.xml` (all elements except `applications` and `resources`)

To apply the changes, you can use the Administration interface as follows:

1. Go to the server instance page.
2. Select the General tab.
3. Select the Apply Changes button.

Or you can use the command line as follows:

```
asadmin reconfig --user user [--password password] [--passwordfile password_file] [--host machine]
[--port port] [--discardmanualchanges=false | --keepmanualchanges=false] instance_name
```

For example:

```
asadmin reconfig --user joeuser --password secret --keepmanualchanges=true server1
```

When you change the configuration and apply the changes, the new configuration, which contains all the information from the dynamically configurable files, is loaded into memory.

---

**NOTE** Sometimes when you apply configuration changes, the server instance displays an error message. For example, when security is turned on and you apply changes, you may see an `Invalid configuration` error.

If this happens, restart the instance as soon as a restart message such as `Instance restart is required` or `Server restart needed` is displayed. Deferring the restart may cause similar error messages to be shown repeatedly.

---

## Manually Editing Configuration Files

When you change the server instance using the Administration interface or the command line, these changes are held in memory. When you apply the changes, they are written to the configuration files behind the scenes.

When you change the server instance by manually editing configuration files, the server instance does not recognize the changes until you apply them.

Therefore, if you make both kinds of changes without applying them, you must keep one set of changes and discard the other. Because of this possibility of overwriting changes, manually editing configuration files is risky.

The `loadbalancer.xml` file is an exception; it can only be edited manually. For details, see [“loadbalancer.xml” on page 129](#).

Here are some cautionary guidelines for manually editing configuration files:

- If you have made both kinds of changes without applying them, you have the following alternatives:
  - Apply the changes using the Administration interface, which forces you to choose between your manual and Administration interface changes.
  - Apply the changes using the `asadmin reconfig` command, and use the `--keepmanualchanges` or `--discardmanualchanges` option to choose. If you don't use either option, manual changes are discarded. For example:

```
asadmin reconfig --user joeuser --password secret --keepmanualchanges=true server1
```

To avoid this dilemma, always apply changes immediately after making them.

- Edit only the files in the *instance\_dir/config* directory. Do not copy files to the server's backup directory or remove the timestamp files (which prevent overwriting of manual changes). However, making your own backup files before making manual changes is a good practice, in case your changes result in an incorrect configuration.
- Whenever the configuration in *server.xml* is modified using the Administration interface, it is always validated against the *sun-server\_1\_1.dtd* file. Additional validation is done that pertains to checking the file's cross-references. If you manually edit *server.xml*, it is up to you to make sure the configuration is complete and correct.
- When you manually edit *applications* or *resources* subelements of *server.xml*, the changes take effect immediately. You do not need to apply the changes. Although in this case you don't have to worry about future changes overwriting these changes, it is especially important to make sure the configuration is complete and correct.
- If you have made manual changes, you must apply those changes before restarting the server.
- The Administration Server remembers unapplied changes across its own restarts. Therefore, at times the Administration Server may have a view of a server instance's configuration that is not the same as the actual configuration of the server instance. To avoid this inconsistency, always apply changes to all server instances before restarting the Administration Server.

## Reconfiguring a Cluster

To change the configuration of a cluster of server instances, use the `cladmin` command. This command can perform the following operations on a cluster:

- Starting or stopping all server instances
- Deploying or undeploying an application
- Creating or deleting a JDBC resource and connection pool
- Configuring high-availability HTTP session persistence

For details, see the *Sun Java System Application Server Administrator's Guide*.



# Server Configuration Files

The `server.xml` file contains most of the server configuration. The encoding is UTF-8 to maintain compatibility with regular UNIX text editors. The `server.xml` file is located in the `instance_dir/config` directory. A schema file, `sun-server_1_1.dtd`, determines the format and content of the `server.xml` file.

This chapter describes `server.xml` and `sun-server_1_1.dtd` in these sections:

- [The sun-server\\_1\\_1.dtd File](#)
- [Elements in the server.xml File](#)
- [General Elements](#)
- [Listener Service Elements](#)
- [Container Elements](#)
- [Availability Elements](#)
- [J2EE Service Elements](#)
- [Java Configuration Elements](#)
- [Resource Elements](#)
- [Application Elements](#)
- [User Database Selection](#)
- [The Sun Java System LDAP Schema](#)
- [Variables](#)
- [Sample server.xml File](#)

---

**NOTE** Virtual servers are not the same thing as server instances. Each server instance is a completely separate server that contains one or more virtual servers.

---

## The sun-server\_1\_1.dtd File

The sun-server\_1\_1.dtd file defines the structure of the server.xml file, including the elements it can contain and the subelements and attributes these elements can have. The sun-server\_1\_1.dtd file is located in the *install\_dir/lib/dtds* directory.

---

**NOTE** Do not edit the sun-server\_1\_1.dtd file; its contents change only with new versions of Sun Java System Application Server.

---

---

**NOTE** The sun-server\_1\_1.dtd interface is Unstable. An unstable interface may be experimental or transitional, and hence may change incompatibly, be removed, or be replaced by a more stable interface in the next release.

---

For general information about DTD files and XML, see the XML specification at:

<http://www.w3.org/TR/REC-xml>

Each element defined in a DTD file (which may be present in the corresponding XML file) can contain the following:

- [Subelements](#)
- [Data](#)
- [Attributes](#)



# Subelements

Elements can contain subelements. For example, the following file fragment defines the `iiop-listener` element.

```
<!ELEMENT iiop-listener (ssl?, property*)>
```

The `ELEMENT` tag specifies that an `iiop-listener` element can contain `ssl` and `property` elements in that order.

The following table shows how optional suffix characters of subelements determine the requirement rules, or number of allowed occurrences, for the subelements.

**Table 2-1** Requirement Rules and Subelement Suffixes

Subelement Suffix	Requirement Rule
<i>element*</i>	Can contain <i>zero or more</i> of this subelement.
<i>element?</i>	Can contain <i>zero or one</i> of this subelement.
<i>element+</i>	Must contain <i>one or more</i> of this subelement.
<i>element</i> (no suffix)	Must contain <i>only one</i> of this subelement.

If an element cannot contain other elements, you see `EMPTY` or `(#PCDATA)` instead of a list of element names in parentheses.

# Data

Some elements contain character data instead of subelements. These elements have definitions of the following format:

```
<!ELEMENT element-name (#PCDATA)>
```

For example:

```
<!ELEMENT description (#PCDATA)>
```

In the `server.xml` file, white space is treated as part of the data in a data element. Therefore, there should be no extra white space before or after the data delimited by a data element. For example:

```
<description>shopping cart bean</description>
```

## Attributes

Elements that have ATTLIST tags contain attributes (name-value pairs). For example:

```
<!ATTLIST iiop-listener      id          CDATA      #REQUIRED
                             address     CDATA      #REQUIRED
                             port        CDATA      "3700"
                             enabled     %boolean;  "true">
```

An `iiop-listener` element can contain `id`, `address`, `port`, and `enabled` attributes.

The `#REQUIRED` label means that a value must be supplied. The `#IMPLIED` label means that the attribute is optional, and that Sun Java System Application Server generates a default value. Wherever possible, explicit defaults for optional attributes (such as `"true"`) are listed.

Attribute declarations specify the type of the attribute. For example, `CDATA` means character data, and `%boolean` is a predefined enumeration.

## Elements in the server.xml File

This section describes the XML elements in the `server.xml` file. Elements are grouped as follows:

- [General Elements](#)
- [Listener Service Elements](#)
- [Container Elements](#)
- [J2EE Service Elements](#)
- [Java Configuration Elements](#)
- [Resource Elements](#)
- [Application Elements](#)

---

**NOTE** Subelements must be defined in the order in which they are listed under each **Subelements** heading unless otherwise noted.

---

For an alphabetical listing of elements in `server.xml`, see [Appendix B, “Alphabetical List of Server Configuration Elements.”](#)

# General Elements

General elements are as follows:

- `server`
- `property`
- `description`
- `admin-service`
- `server-instance`

## server

Defines a server. This is the root element; there can only be one `server` element in a `server.xml` file.

### Subelements

The following table describes subelements for the `server` element.

**Table 2-2** `server` Subelements

Element	Required	Description
<code>http-service</code>	only one	Defines the HTTP service.
<code>iiop-service</code>	only one	Defines the IIOP service.
<code>admin-service</code>	zero or one	Not implemented.
<code>web-container</code>	only one	Configures the web container.
<code>ejb-container</code>	only one	Configures the Enterprise JavaBeans™ (EJB™) container.
<code>mdb-container</code>	only one	Configures the message-driven bean (MDB) container.
<code>jms-service</code>	only one	Configures the Java™ Message Service (JMS) provider.
<code>log-service</code>	only one	Configures the system logging service.
<code>security-service</code>	only one	Defines information needed by the J2EE security service.
<code>transaction-service</code>	only one	Configures the transaction service.
<code>java-config</code>	only one	Contains the Java™ Virtual Machine (JVM™) configuration.
<code>resources</code>	only one	Contains configured resources.

**Table 2-2**    `server` Subelements (*Continued*)

Element	Required	Description
<code>applications</code>	only one	Contains deployed J2EE applications, J2EE modules, and lifecycle modules.
<code>availability-service</code>	zero or one	Enables high-availability features.
<code>property</code>	zero or more	Specifies a property or a variable.

**Attributes**

The following table describes attributes for the `server` element.

**Table 2-3**    `server` Attributes

Attribute	Default	Description
<code>name</code>	<code>none</code>	Specifies the name of the server instance.
<code>locale</code>	operating system default	(optional) Specifies the server instance language.
<code>log-root</code>	<code>instance_dir/logs</code>	(optional) Specifies where the server instance's log files are kept. The directory in which the server log is kept must be writable by whatever user account the server runs as. See the <code>log-service</code> description for details about logs.
<code>application-root</code>	<code>instance_dir/applications</code>	(optional) Specifies the absolute path where deployed applications reside for this server instance.
<code>session-store</code>	<code>instance_dir/session-store</code>	(optional) Specifies the directory where passivated beans are stored in the file system.

**property**

Specifies a property, or a variable that is defined in `server.xml` and referenced in `obj.conf`. For information about variables, see “[Variables](#)” on page 92.

A property adds configuration information to its parent element that is one or both of the following:

- Optional with respect to Sun Java System Application Server
- Needed by a system or object that Sun Java System Application Server doesn't have knowledge of, such as an LDAP server or a Java class

For example, an `auth-realm` element can include `property` subelements:

```
<auth-realm name="file"
  classname="com.iplanet.ias.security.auth.realm.file.FileRealm">
  <property name="file" value="instance_dir/config/keyfile" />
  <property name="jaas-context" value="fileRealm" />
</auth-realm>
```

Which properties an `auth-realm` element uses depends on the value of the `auth-realm` element's `name` attribute. The `file` realm uses `file` and `jaas-context` properties. Other realms use different properties.

**Subelements**

The following table describes subelements for the `property` element.

**Table 2-4**    `property` Subelements

Element	Required	Description
<a href="#">description</a>	zero or one	Contains a text description of this element.

**Attributes**

The following table describes attributes for the `property` element.

**Table 2-5**    `property` Attributes

Attribute	Default	Description
<code>name</code>	<code>none</code>	Specifies the name of the property or variable.
<code>value</code>	<code>none</code>	Specifies the value of the property or variable.

**description**

Contains a text description of the parent element.

**Subelements**

`none`

**Attributes**

`none`

## **admin-service**

This element is not implemented and should not be used.

## **server-instance**

This element is not implemented and should not be used.

# Listener Service Elements

Listener service elements are as follows:

- `http-service`
- `http-listener`
- `ssl`
- `mime`
- `acl`
- `virtual-server-class`
- `virtual-server`
- `http-qos`
- `auth-db`
- `iiop-service`
- `iiop-listener`
- `orb`

## **http-service**

Defines the HTTP service.

For more information about the quality of service features defined in this element's attributes, see the *Sun Java System Application Server Performance Tuning Guide*.

### **Subelements**

The following table describes subelements for the `http-service` element.

**Table 2-6** http-service Subelements

Element	Required	Description
<code>http-listener</code>	zero or more	Defines an HTTP listen socket.
<code>mime</code>	zero or more	Defines MIME types.
<code>acl</code>	zero or more	References an ACL file.
<code>virtual-server-class</code>	zero or more	Defines a virtual server class.
<code>http-qos</code>	zero or one	Defines quality of service parameters.
<code>property</code>	zero or more	Specifies a property or a variable.

**NOTE** The `http-listener`, `mime`, `acl`, and `virtual-server-class` elements can occur in any order, but `http-qos` and `property` elements must occur second to last and last, respectively.

### Attributes

The following table describes attributes for the `http-service` element.

**Table 2-7** http-service Attributes

Attribute	Default	Description
<code>qos-metrics-interval-in-seconds</code>	30	(optional) Specifies the interval during which traffic is measured.
<code>qos-recompute-time-interval-in-millis</code>	100	(optional) Specifies the period in which the bandwidth gets recomputed for all server entities.
<code>qos-enabled</code>	true	(optional) Enables quality of service features, which let you set limits on server entities or view server statistics for bandwidth and connections.

# http-listener

Defines an HTTP listen socket.

**NOTE**

When you create a secure listener through the Administration interface, security is automatically turned on globally in `init.conf`. When you create a secure listener manually in `server.xml`, you must manually turn on security by editing the `init.conf` file's [Security](#) directive.

## Subelements

The following table describes subelements for the `http-listener` element.

**Table 2-8** http-listener Subelements

Element	Required	Description
<a href="#">ssl</a>	zero or one	Defines SSL parameters.

## Attributes

The following table describes attributes for the `http-listener` element.

**Table 2-9** http-listener Attributes

Attribute	Default	Description
<code>id</code>	<code>none</code>	The unique listener name. An <code>http-listener</code> name cannot begin with a number.
<code>address</code>	<code>none</code>	IP address of the listener. Can be in dotted-pair or IPv6 notation. Can be <code>any</code> (for <code>INADDR_ANY</code> ) to listen on all IP addresses. Can be a hostname.
<code>port</code>	<code>none</code>	Port number on which the listener listens. Legal values are 1 - 65535. On UNIX, creating sockets that listen on ports 1 - 1024 requires superuser privileges. Configuring an SSL listener to listen on port 443 is standard.
<code>family</code>	<code>inet</code>	(optional) The socket family type. Legal values are <code>inet</code> , <code>inet6</code> , and <code>nca</code> . Use the value <code>inet6</code> for IPv6 listeners. If this value is <code>inet6</code> , IPv4 addresses are prefixed with <code>::ffff:</code> in the server log. Specify <code>nca</code> to make use of the Solaris Network Cache and Accelerator.
<code>acceptor-threads</code>	<code>1</code>	(optional) Number of acceptor threads for the listener, typically the number of processors in the machine. Legal values are 1 - 1024.



**Table 2-9** http-listener Attributes (*Continued*)

Attribute	Default	Description
blocking-enabled	false	(optional) Determines whether the listener and the accepted socket are put in to blocking mode. Use of blocking mode may improve benchmark scores.
security-enabled	false	(optional) Determines whether the listener runs SSL. You can turn SSL2 or SSL3 on or off and set ciphers using an <code>ssl</code> element.  The <code>Security</code> setting in the <code>init.conf</code> file globally enables or disables SSL by making certificates available to the server instance. Therefore, <code>Security</code> in <code>init.conf</code> must be on or <code>security-enabled</code> in <code>server.xml</code> does not work. For more information, see <a href="#">Chapter 3, "Syntax and Use of <code>init.conf</code>."</a>
default-virtual-server	none	The <code>id</code> attribute of the default virtual server for this particular listener.
server-name	none	Tells the server what to put in the host name section of any URLs it sends to the client. This affects URLs the server automatically generates; it doesn't affect the URLs for directories and files stored in the server. This name should be the alias name if your server uses an alias.  If you append a colon and port number, that port will be used in URLs the server sends to the client.
enabled	true	(optional) Determines whether the listener is active.

---

**CAUTION** Blocking mode sockets should not be used in real world deployments. Use of blocking mode sockets precludes dynamic reconfiguration and exposes the server to denial of service attacks.

---

## ssl

Defines SSL (Secure Socket Layer) parameters.

An `ssl` element is required inside an `http-listener` element that has its `security-enabled` attribute set to on.

An `ssl` element is only allowed inside an `http-listener` or `iiop-listener` element.

Subelements

none

Attributes

The following table describes attributes for the `ssl` element.

**Table 2-10** `ssl` Attributes

Attribute	Default	Description
<code>cert-nickname</code>	<code>none</code>	The nickname of the server certificate in the certificate database or the PKCS#11 token. In the certificate, the name format is <i>tokenname:nickname</i> . Including the <i>tokenname</i> : part of the name in this attribute is optional.
<code>ssl2-enabled</code>	<code>false</code>	(optional) Determines whether SSL2 is enabled.  If both SSL2 and SSL3 are enabled for a virtual server, the server tries SSL3 encryption first. If that fails, the server tries SSL2 encryption.
<code>ssl2-ciphers</code>	<code>none</code>	(optional) A comma-separated list of the SSL2 ciphers used, with the prefix + to enable or - to disable, for example <code>+rc4</code> . Allowed values are <code>rc4</code> , <code>rc4export</code> , <code>rc2</code> , <code>rc2export</code> , <code>idea</code> , <code>des</code> , <code>desede3</code> .
<code>ssl3-enabled</code>	<code>true</code>	(optional) Determines whether SSL3 is enabled. The default is <code>true</code> .  If both SSL2 and SSL3 are enabled for a virtual server, the server tries SSL3 encryption first. If that fails, the server tries SSL2 encryption.
<code>ssl3-tls-ciphers</code>	<code>none</code>	(optional) A comma-separated list of the SSL3 ciphers used, with the prefix + to enable or - to disable, for example <code>+rsa_des_sha</code> . Allowed SSL3 values are <code>rsa_rc4_128_md5</code> , <code>rsa_3des_sha</code> , <code>rsa_des_sha</code> , <code>rsa_rc4_40_md5</code> , <code>rsa_rc2_40_md5</code> , <code>rsa_null_md5</code> . Allowed TLS values are <code>rsa_des_56_sha</code> , <code>rsa_rc4_56_sha</code> .
<code>tls-enabled</code>	<code>true</code>	(optional) Determines whether TLS is enabled.
<code>tls-rollback-enabled</code>	<code>true</code>	(optional) Determines whether TLS rollback is enabled. TLS rollback should be enabled for Microsoft Internet Explorer 5.0 and 5.5. For more information, see the <i>Sun Java System Application Server Administrator's Guide</i> .
<code>client-auth-enabled</code>	<code>false</code>	(optional) Determines whether SSL3 client authentication is performed on every request, independent of ACL-based access control.

## mime

Defines MIME types.

The most common way that the server determines the MIME type of a requested resource is by invoking the `type-by-extension` directive in the `ObjectType` section of the `obj.conf` file. The `type-by-extension` function does not work if no `mime` element has been defined in the `server` element.

---

<b>NOTE</b>	The <code>mime.types</code> interface is Unstable. An unstable interface may be experimental or transitional, and hence may change incompatibly, be removed, or be replaced by a more stable interface in the next release.
-------------	---

---

### Subelements

none

### Attributes

The following table describes attributes for the `mime` element.

**Table 2-11** mime Attributes

Attribute	Default	Description
id	none	Internal name for the MIME types listing. Used in a <code>virtual-server</code> element to define the MIME types used by the virtual server. The MIME types name cannot begin with a number.
file	none	The name of a MIME types file. For information about the format of this file, see <a href="#">Chapter 4, "MIME Types."</a>

## acl

References an ACL file.

---

<b>NOTE</b>	The ACL file interface is Unstable. An unstable interface may be experimental or transitional, and hence may change incompatibly, be removed, or be replaced by a more stable interface in the next release.
-------------	--

---

**Subelements**

none

**Attributes**

The following table describes attributes for the `acl` element.

**Table 2-12** `acl` Attributes

Attribute	Default	Description
<code>id</code>	none	Internal name for the ACL file listing. Used in a <code>virtual-server</code> element to define the ACL file used by the virtual server. An ACL file listing name cannot begin with a number.
<code>file</code>	none	An ACL file. Each ACL file must have a unique name. For information about the format of an ACL file, see the <i>Sun Java System Application Server Administrator's Guide to Security</i> .

**virtual-server-class**

Defines a virtual server class.

**Subelements**

The following table describes subelements for the `virtual-server-class` element.

**Table 2-13** `virtual-server-class` Subelements

Element	Required	Description
<code>virtual-server</code>	zero or more	Defines a virtual server.
<code>http-qos</code>	zero or one	Defines quality of service parameters.
<code>property</code>	zero or more	Specifies a property or a variable.

**Attributes**

The following table describes attributes for the `virtual-server-class` element.

**Table 2-14** `virtual-server-class` Attributes

Attribute	Default	Description
<code>id</code>	<code>none</code>	Virtual server class ID. This is a unique ID that allows lookup of a specific virtual server class. A virtual server class ID cannot begin with a number.
<code>config-file</code>	<code>none</code>	(optional) The file name of the <code>obj.conf</code> file for this class of virtual servers. Can be overridden in a <code>virtual-server</code> element.
<code>default-object</code>	<code>default</code>	<p>(optional) Tells the server which object loaded from an <code>obj.conf</code> file is the default. Can be overridden in a <code>virtual-server</code> element.</p> <p>The default object is expected to have all the name translation (<code>NameTrans</code>) directives for the virtual server; any server behavior that is configured in the default object affects the entire virtual server class.</p> <p>If you specify an object that doesn't exist, the server doesn't report an error until a client tries to retrieve a document. The Administration interface assumes the default to be the object named <code>default</code>. Don't deviate from this convention if you use (or plan to use) the Administration interface.</p>
<code>accept-language</code>	<code>false</code>	(optional) If <code>true</code> , the server parses the <code>Accept-Language</code> header and sends an appropriate language version based on which language the client can accept. You should set this value to <code>true</code> only if the server supports multiple languages. Can be overridden in a <code>virtual-server</code> element. Can be overridden in a <code>virtual-server</code> element.
<code>enabled</code>	<code>true</code>	(optional) Determines whether the virtual server class is active.

## virtual-server

Defines a virtual server. A virtual server, also called a virtual host, is a virtual web server that serves content targeted for a specific URL. Multiple virtual servers may serve content using the same or different host names, port numbers, or IP addresses. The HTTP service can direct incoming web requests to different virtual servers based on the URL.

When you first install Sun Java System Application Server, a default virtual server is created. (You can also assign a default virtual server to each new [http-listener](#) you create.)

---

**NOTE** Virtual servers are not the same thing as server instances. Each server instance is a completely separate server that contains one or more virtual servers.

---

Before the Sun Java System Application Server can process a request, it must accept the request via a listener, then direct the request to the correct virtual server. The virtual server is determined as follows:

- If the listener is configured to only a default virtual server, that virtual server is selected.
- If the listener has more than one virtual server configured to it, the request `Host` header is matched to the `hosts` attribute of a virtual server. If no `Host` header is present or no `hosts` attribute matches, the default virtual server for the listener is selected.

If a virtual server is configured to an SSL listener, its `hosts` attribute is checked against the subject pattern of the certificate at server startup, and a warning is generated and written to the server log if they don't match.

### Subelements

The following table describes subelements for the `virtual-server` element.

**Table 2-15** `virtual-server` Subelements

Element	Required	Description
<code>http-qos</code>	zero or one	Defines quality of service parameters.
<code>auth-db</code>	zero or more	Defines the user database for the virtual server; not applicable to J2EE applications.
<code>property</code>	zero or more	Specifies a property or a variable.

### Attributes

The following table describes attributes for the `virtual-server` element.

**Table 2-16** virtual-server Attributes

Attribute	Default	Description
id	none	Virtual server ID. This is a unique ID that allows lookup of a specific virtual server. Can also be referred to as the variable <code>\$id</code> in an <code>obj.conf</code> file. A virtual server ID cannot begin with a number.
http-listeners	none	(optional) A comma-separated list of <code>http-listener</code> ids that specify the connection(s) the virtual server uses. Required only for a <code>virtual-server</code> that is not the <code>default-virtual-server</code> of an <code>http-listener</code> .
default-web-module	none	<p>(optional) The default <code>web-module</code> for this virtual server, which responds to all requests that cannot be resolved to other web modules deployed to this virtual server.</p> <p>If this attribute is empty, the <code>web-module</code> assigned to this virtual server that has an empty <code>context-root</code> attribute is used. If no <code>web-module</code> assigned to this virtual server has an empty <code>context-root</code>, the system default web module is used.</p>
config-file	<i>virtual_server_name-obj.conf</i>	(optional) The file name of the <code>obj.conf</code> file for this virtual server. Can override the value in a <code>virtual-server-class</code> element.
default-object	default	<p>(optional) Tells the server which object loaded from an <code>obj.conf</code> file is the default. Can override the value in a <code>virtual-server-class</code> element.</p> <p>The default object is expected to have all the name translation (<code>NameTrans</code>) directives for the virtual server; any server behavior that is configured in the default object affects the entire server. The default value is <code>default</code>.</p> <p>If you specify an object that doesn't exist, the server doesn't report an error until a client tries to retrieve a document. The Administration interface assumes the default to be the object named <code>default</code>. Don't deviate from this convention if you use (or plan to use) the Administration interface.</p>

**Table 2-16** virtual-server Attributes (Continued)

Attribute	Default	Description
hosts	none	A comma-separated list of values allowed in the Host request header to select the current virtual server. Each virtual-server that is configured to the same http-listener must have a unique hosts value for that listener.
mime	none	The id of the mime element used by the virtual server.
state	on	(optional) Determines whether a virtual-server is active (on) or inactive (off, disabled). The default is on (active). When inactive, a virtual-server does not service requests.  If a virtual-server is disabled, only the global server administrator can turn it on.
acls	none	(optional) One or more id attributes of acl elements, separated by commas. Specifies the ACL file(s) used by the virtual server.
accept-language	false	(optional) If true, the server parses the Accept-Language header and sends an appropriate language version based on which language the client can accept. You should set this value to on only if the server supports multiple languages. Can override the value in a virtual-server-class element.
log-file	server.log in the directory specified by the log-root attribute of the <a href="#">server</a> element	(optional) Writes this virtual server's log messages to a log file separate from the server log. The file and directory in which the virtual server log is kept must be writable by whatever user account the server runs as. See the <a href="#">log-service</a> description for details about logs.

**Properties**

The following table describes properties for the virtual-server element.

**Table 2-17** virtual-server Properties

Property	Default	Description
sso-enabled	true	If true, single sign-on is enabled for web applications on this virtual server that are configured for the same realm. If false, single sign-on is disabled for this virtual server, and users must authenticate separately to every application on the virtual server.



**Table 2-17** virtual-server Properties (*Continued*)

Property	Default	Description
sso-max-inactive-se conds	300	Specifies the time after which a user's single sign-on record becomes eligible for purging if no client activity is received. Since single sign-on applies across several applications on the same virtual server, access to any of the applications keeps the single sign-on record active. Higher values provide longer single sign-on persistence for the users at the expense of more memory use on the server.
sso-reap-interval-s econds	60	Specifies the interval between purges of expired single sign-on records.

## http-qos

Defines quality of service parameters of an `http-service`, `virtual-server-class`, or `virtual-server` element.

Attributes in the `http-service` element activate the quality of service features.

For more information, see the *Sun Java System Application Server Performance Tuning Guide*.

### Subelements

none

### Attributes

The following table describes attributes for the `http-qos` element.

**Table 2-18** http-qos Attributes

Attribute	Default	Description
bandwidth-limit	none	(required if <code>enforce-bandwidth-limit</code> is true) The maximum bandwidth limit for the server, <code>virtual-server-class</code> , or <code>virtual-server</code> in bytes per second.
enforce-bandwidth-lim it	false	(optional) Specifies whether the bandwidth limit should be enforced or not.
connection-limit	none	(required if <code>enforce-connection-limit</code> is true) The maximum number of concurrent connections for the server, <code>virtual-server-class</code> , or <code>virtual-server</code> .
enforce-connection-li mit	false	(optional) Specifies whether the connection limit should be enforced or not.

## auth-db

Defines the user database used by the `virtual-server`; not applicable to J2EE applications.

See [“User Database Selection” on page 90](#) for more information about how a user database is selected for a given virtual server.

---

<b>NOTE</b>	The user database applies only to the security of the server itself. It is not related to J2EE application and module security. For more information, see the <i>Sun Java System Application Server Administrator's Guide to Security</i> .
-------------	---

---

### Subelements

none

### Attributes

The following table describes attributes for the `auth-db` element.

**Table 2-19** `auth-db` Attributes

Attribute	Default	Description
<code>id</code>	<code>none</code>	The user database name in the virtual server's ACL file. A user database name cannot begin with a number.
<code>database</code>	<code>none</code>	The user database name in the <code>dbswitch.conf</code> file.
<code>basedn</code>	<code>none</code>	(optional) Overrides the base DN lookup in the <code>dbswitch.conf</code> file. However, the <code>basedn</code> value is still relative to the base DN value from the <code>dbswitch.conf</code> entry.
<code>certmaps</code>	<code>none</code>	(optional) Specifies which certificate to LDAP entry mappings (defined in <code>certmap.conf</code> ) to use. If not present, all mappings are used. All lookups based on mappings in <code>certmap.conf</code> are relative to the final base DN of the <code>virtual-server</code> .

---

## iiop-service

Defines the IIOP service.

### Subelements

The following table describes subelements for the `iiop-service` element.

**Table 2-20** `iiop-service` Subelements

Element	Required	Description
<code>orb</code>	only one	Configures the ORB.
<code>ssl-client-config</code>	zero or one	Defines SSL parameters for the ORB.
<code>iiop-listener</code>	zero or more	Defines an IIOP listen socket.

**Attributes**

none

**orb**

Configures the ORB.

To enable SSL for outbound connections, include an `ssl-client-config` subelement in the parent `iiop-service` element.

**Subelements**

The following table describes subelements for the `orb` element.

**Table 2-21** `orb` Subelements

Element	Required	Description
<code>property</code>	zero or more	Specifies a property or a variable.

**Attributes**

The following table describes attributes for the `orb` element.

**Table 2-22** `orb` Attributes

Attribute	Default	Description
<code>message-fragment-size</code>	1024	(optional) GIOPv1.2 messages larger than this number of bytes are fragmented.
<code>steady-thread-pool-size</code>	10	(optional) The minimum number of worker threads in the ORB.
<code>max-thread-pool-size</code>	200	(optional) The maximum number of worker threads in the ORB.

**Table 2-22**    orb Attributes (*Continued*)

Attribute	Default	Description
idle-thread-timeout-in-seconds	300	(optional) Idle worker threads are removed from the pool after this amount of time.
max-connections	1024	(optional) The maximum number of incoming connections on all IIOP listeners. Legal values are integers.
log-level	Value of level attribute of <a href="#">log-service</a> element	(optional) Controls the type of messages logged by this element to the server log. For details, see the description of the level attribute of the <a href="#">log-service</a> element.
monitoring-enabled	false	(optional) Determines whether monitoring of the ORB is enabled.

## ssl-client-config

Defines SSL parameters for the ORB when it makes outbound SSL connections and behaves as a client.

### Subelements

The following table describes subelements for the `ssl-client-config` element.

**Table 2-23**    `ssl-client-config` Subelements

Element	Required	Description
<a href="#">ssl</a>	only one	Defines SSL parameters.

### Attributes

none

## iiop-listener

Defines an IIOP listen socket.

To enable SSL for this listener, include an `ssl` subelement.

<b>NOTE</b>	When you create a secure listener through the Administration interface, security is automatically turned on globally in <code>init.conf</code> . When you create a secure listener manually in <code>server.xml</code> , you must manually turn on security by editing the <code>init.conf</code> file's <a href="#">Security</a> directive.
-------------	--

### Subelements

The following table describes subelements for the `iiop-listener` element.

**Table 2-24** `iiop-listener` Subelements

Element	Required	Description
<code>ssl</code>	zero or one	Defines SSL parameters.
<code>property</code>	zero or more	Specifies a property or a variable.

### Attributes

The following table describes attributes for the `iiop-listener` element.

**Table 2-25** `iiop-listener` Attributes

Attribute	Default	Description
<code>id</code>	<code>none</code>	The listener name. An <code>iiop-listener</code> name cannot begin with a number.
<code>address</code>	<code>none</code>	IP address of the listener. Can be in dotted-pair or IPv6 notation or just a name.
<code>port</code>	3700 (for the first server instance)	(optional) Port number to create the listener on. Legal values are 1 - 65535. On UNIX, creating sockets that listen on ports 1 - 1024 requires superuser privileges.
<code>enabled</code>	<code>true</code>	(optional) Determines whether the listener is active.

## Container Elements

Container configuration elements are as follows:

- `web-container`
- `ejb-container`
- `mdb-container`

# web-container

Configures the web container.

## Subelements

The following table describes subelements for the web-container element.

**Table 2-26** web-container Subelements

Element	Required	Description
<a href="#">session-config</a>	zero or one	Specifies session configuration information for the web container.
<a href="#">property</a>	zero or more	Specifies a property or a variable.

## Attributes

The following table describes attributes for the web-container element.

**Table 2-27** web-container Attributes

Attribute	Default	Description
monitoring-enabled	false	(optional) Determines whether monitoring of the web container is enabled.
log-level	Value of level attribute of <a href="#">log-service</a> element	(optional) Controls the type of messages logged by this element to the server log. For details, see the description of the level attribute of the <a href="#">log-service</a> element.  ServletContext.log messages are logged at the INFO level by default.
availability-enabled	Value of same attribute in <a href="#">availability-service</a> element	(optional) Determines whether availability is enabled at the web container level. For more information about availability levels, see the <a href="#">availability-service</a> element description.

# ejb-container

Configures the EJB container. Stateless session beans are maintained in pools. Stateful session beans have session affinity and are cached. Entity beans associated with a database primary key are also cached. Entity beans not yet associated with a primary key are maintained in pools. Pooled entity beans are used to run `ejbCreate()` and finder methods.

## Subelements

The following table describes subelements for the ejb-container element.

**Table 2-28** `ejb-container` Subelements

Element	Required	Description
<code>property</code>	zero or more	Specifies a property or a variable.

### Attributes

The following table describes attributes for the `ejb-container` element.

**Table 2-29** `ejb-container` Attributes

Attribute	Default	Description
<code>steady-pool-size</code>	32	<p>(optional) Specifies the initial and minimum number of beans maintained in the pool. Must be 0 or greater and less than <code>max-pool-size</code>.</p> <p>Bean instances are removed from the pool and returned after use. The pool is replenished or cleaned up periodically to maintain this size.</p> <p>Applies to stateless session beans and entity beans.</p>
<code>pool-resize-quantity</code>	16	<p>(optional) Specifies the number of beans to be:</p> <ul style="list-style-type: none"> <li>created if a request arrives when the pool has no available beans (subject to the <code>max-pool-size</code> limit)</li> <li>removed when the <code>pool-idle-timeout-in-seconds</code> timer expires and a cleaner thread removes any unused instances</li> </ul> <p>Must be 0 or greater and less than <code>max-pool-size</code>. The pool is not resized below the <code>steady-pool-size</code>.</p> <p>Applies to stateless session beans and entity beans.</p>
<code>max-pool-size</code>	64	<p>(optional) Specifies the maximum number of beans that can be created to satisfy client requests. A value of 0 indicates an unbounded pool.</p> <p>Applies to stateless session beans and entity beans.</p>

**Table 2-29**   `ejb-container` Attributes (*Continued*)

Attribute	Default	Description
<code>cache-resize-quantity</code>	16	<p>(optional) Specifies the number of beans to be:</p> <ul style="list-style-type: none"><li>• created if a request arrives when the pool has no available beans (subject to the <code>max-cache-size</code> limit)</li><li>• passivated when the <code>cache-idle-timeout-in-seconds</code> timer expires and a cleaner thread removes any unused instances, or when the cache size exceeds <code>max-cache-size</code>.</li></ul> <p>Must be greater than 1 and less than <code>max-cache-size</code>.</p> <p>Applies to stateful session beans and entity beans.</p>
<code>max-cache-size</code>	512	<p>(optional) Specifies the maximum number of beans in the cache. A value of 0 indicates an unbounded cache.</p> <p>Applies to stateful session beans and entity beans.</p>
<code>pool-idle-timeout-in-seconds</code>	600	<p>(optional) Specifies the maximum time that a bean can remain idle in the pool. After this amount of time, the pool can remove this bean. A value of 0 specifies that idle beans can remain in the pool indefinitely.</p> <p>Applies to stateless session beans and entity beans.</p>
<code>cache-idle-timeout-in-seconds</code>	600	<p>(optional) Specifies the maximum time that a bean can remain idle in the cache. After this amount of time, the container can passivate this bean. A value of 0 specifies that beans may never become candidates for passivation.</p> <p>Applies to stateful session beans and entity beans.</p>



**Table 2-29** `ejb-container` Attributes (*Continued*)

Attribute	Default	Description
<code>removal-timeout-in-seconds</code>	5400	<p>(optional) Specifies the amount of time that a bean can remain passivated before it is removed from the session store. A value of 0 specifies that the container does not remove inactive beans automatically.</p> <p>If <code>removal-timeout-in-seconds</code> is less than or equal to <code>cache-idle-timeout-in-seconds</code>, beans are removed immediately without being passivated.</p> <p>The <code>session-store</code> attribute of the <a href="#">server</a> element determines the location of the session store.</p> <p>Applies to stateful session beans and entity beans.</p>
<code>victim-selection-policy</code>	<code>nru</code>	<p>(optional) Specifies how entity and stateful session beans are selected for passivation. Allowed values are <code>fifo</code>, <code>lru</code>, and <code>nru</code>:</p> <ul style="list-style-type: none"> <li>• <code>fifo</code> selects the oldest instance.</li> <li>• <code>lru</code> selects the least recently accessed instance.</li> <li>• <code>nru</code> selects a not recently used instance.</li> </ul>
<code>commit-option</code>	<code>B</code>	(optional) Determines which commit option is used for entity beans. Legal values are <code>B</code> or <code>C</code> .
<code>log-level</code>	Value of <code>level</code> attribute of <a href="#">log-service</a> element	(optional) Controls the type of messages logged by this element to the server log. For details, see the description of the <code>level</code> attribute of the <a href="#">log-service</a> element.
<code>monitoring-enabled</code>	<code>false</code>	(optional) Determines whether monitoring of the EJB container is enabled.
<code>availability-enabled</code>	Value of same attribute in <a href="#">availability-service</a> element	(optional) Determines whether availability is enabled at the EJB container level. For more information about availability levels, see the <a href="#">availability-service</a> element description.

## mdb-container

Configures the message-driven bean (MDB) container.

### Subelements

The following table describes subelements for the `mdb-container` element.

**Table 2-30** `mdb-container` Subelements

Element	Required	Description
<code>property</code>	zero or more	Specifies a property or a variable.

### Attributes

The following table describes attributes for the `mdb-container` element.

**Table 2-31** `mdb-container` Attributes

Attribute	Default	Description
<code>steady-pool-size</code>	10	(optional) Specifies the initial and minimum number of beans maintained in the pool.
<code>pool-resize-quantity</code>	2	(optional) Specifies the number of beans to be created if a request arrives when the pool is empty (subject to the <code>max-pool-size</code> limit), or the number of beans to remove if idle for more than <code>idle-timeout-in-seconds</code> .
<code>max-pool-size</code>	60	(optional) Specifies the maximum number of beans that can be created to satisfy client requests.
<code>idle-timeout-in-seconds</code>	600	(optional) Specifies the maximum time that a bean can remain idle in the pool. After this amount of time, the bean is destroyed.
<code>log-level</code>	Value of <code>level</code> attribute of <code>log-service</code> element	(optional) Controls the type of messages logged by this element to the server log. For details, see the description of the <code>level</code> attribute of the <code>log-service</code> element.
<code>monitoring-enabled</code>	false	(optional) Determines whether monitoring of the message-driven bean (MDB) container is enabled.

## Properties

The following table describes properties for the `mdb-container` element.

**Table 2-32** `mdb-container` Properties

Property	Default	Description
<code>cmt-max-runtime-exceptions</code>	1	Specifies the maximum number of <code>RuntimeException</code> occurrences allowed from a message-driven bean's <code>onMessage()</code> method when container-managed transactions are used. Deprecated.
<code>reconnect-enabled</code>	<code>true</code>	If true, the MDB container automatically tries to reconnect to the JMS provider when the connection is broken.  When the connection is broken, depending on the message processing stage, the <code>onMessage()</code> method may not be able to complete successfully or the transaction may be rolled back due to a JMS exception. When the MDB container reestablishes the connection, JMS message redelivery semantics apply.
<code>reconnect-delay-in-seconds</code>	60	Specifies the delay between reconnect attempts.
<code>reconnect-max-retries</code>	60	Specifies the maximum number of reconnect attempts.

# Availability Elements

Availability elements are as follows:

- [availability-service](#)
- [persistence-store](#)
- [session-config](#)
- [session-manager](#)
- [manager-properties](#)
- [store-properties](#)
- [session-properties](#)

## availability-service

Enables high-availability features such as HTTP session and stateful session bean state persistence to the Sun Java System high-availability database (HADB). Also determines whether single sign-on, if configured, is highly available.

### Subelements

The following table describes subelements for the `availability-service` element.

**Table 2-33** `availability-service` Subelements

Element	Required	Description
<code>iiop-cluster</code>	zero or one	Defines IIOP endpoints of application server instances forming a cluster.
<code>persistence-store</code>	zero or one	Configures the HADB.
<code>property</code>	zero or more	Specifies a property or a variable.

### Attributes

The following table describes attributes for the `availability-service` element.

**Table 2-34** `availability-service` Attributes

Attribute	Default	Description
<code>availability-enabled</code>	false	(optional) If set to true, high-availability features apply to all applications deployed to the server instance that do not have availability disabled. You should set the same availability value for all instances in a cluster to ensure consistent behavior.

Availability can be enabled or disabled at the following levels:

1. The server instance (attribute of `availability-service`)
2. The EJB or web container (attribute of `ejb-container` or `web-container`)
3. The application (attribute of `j2ee-application`)
4. The EJB or web module (attribute of `ejb-module` or `web-module`)

For availability to be enabled at a given level, it must be enabled at all higher levels as well. For example, to enable availability at the application level, you must also enable it at the server instance and container levels.

The default for a given level is the setting at the next level up. For example, if availability is enabled at the EJB and web container level, it is enabled by default at the application level. Availability is disabled by default at the server instance level.

---

**NOTE**      Setting `availability-enabled` to `true` at all levels is not sufficient for configuring session persistence. You must also configure this element's subelements and `session-config` and its subelements.

---

If availability is disabled, there is no high availability for HTTP session persistence. In other words, `persistence-type=memory`.

If availability is enabled, high availability for HTTP session persistence is enabled for all applications by default. If no further configuration exists either in the `server.xml` or `sun-web.xml` file, the default session persistence configuration is as follows:

```
persistence-type=ha
persistenceFrequency=time-based
persistenceScope=session
```

You can override this default configuration for all applications in `server.xml` by setting the various `session-config` subelements. You can override this default configuration for a specific application in `sun-web.xml`. For details, see the *Sun Java System Application Server Developer's Guide to Web Applications*.

## iiop-cluster

Defines IIOP endpoints of application server instances forming a cluster.

### Subelements

The following table describes subelements for the `iiop-cluster` element.

**Table 2-35**    `iiop-cluster` Subelements

Element	Required	Description
<code>iiop-server-instance</code>	zero or more	Defines a server instance in the cluster.

### Attributes

none

# iiop-server-instance

Defines a server instance in the cluster.

## Subelements

The following table describes subelements for the `iiop-server-instance` element.

**Table 2-36** `iiop-server-instance` Subelements

Element	Required	Description
<code>iiop-endpoint</code>	zero or more	Defines IIOP endpoints of the server instance.

## Attributes

The following table describes attributes for the `iiop-server-instance` element.

**Table 2-37** `iiop-server-instance` Attributes

Attribute	Default	Description
<code>name</code>	<code>none</code>	Specifies the name of the server instance.

# iiop-endpoint

Defines IIOP endpoints of the server instance.

## Subelements

`none`

## Attributes

The following table describes attributes for the `iiop-endpoint` element.

**Table 2-38** `iiop-endpoint` Attributes

Attribute	Default	Description
<code>id</code>	<code>none</code>	Specifies a unique identifier for this endpoint.
<code>host</code>	<code>none</code>	Specifies an IP address or host name for this endpoint.
<code>port</code>	<code>3600</code>	(optional) Specifies the port number at which the endpoint listens for IIOP requests.

## persistence-store

Configures the Sun Java System high-availability database (HADB).

### Subelements

The following table describes subelements for the `persistence-store` element.

**Table 2-39** `persistence-store` Subelements

Element	Required	Description
<a href="#">property</a>	zero or more	Specifies a property or a variable.

### Attributes

none

### Properties

The following table describes properties for the `persistence-store` element.

**Table 2-40** `persistence-store` Properties

Property	Default Value	Description
<code>store-pool-jndi-name</code>	<code>jdbc/hastore</code>	Specifies the <code>jndi-name</code> of the <a href="#">jdbc-resource</a> used for connections to the HADB for session persistence.  For more information about setting up a connection pool and JDBC resource for the HADB, see the <i>Sun Java System Application Server Installation Guide</i> and the <i>Sun Java System Application Server Administrator's Guide</i> .

## session-config

Specifies session configuration information for the entire web container. Individual web applications can override these settings using the corresponding elements in their `sun-web.xml` files.

### Subelements

The following table describes subelements for the `session-config` element.

**Table 2-41** `session-config` Subelements

Element	Required	Description
<a href="#">session-manager</a>	zero or one	Specifies session manager configuration information.
<a href="#">session-properties</a>	zero or one	Specifies session properties.

**Attributes**

none

**session-manager**

Specifies session manager information.

---

**NOTE**      The session manager interface is Unstable. An unstable interface may be experimental or transitional, and hence may change incompatibly, be removed, or be replaced by a more stable interface in the next release.

---

**Subelements**

The following table describes subelements for the `session-manager` element.

**Table 2-42**    `session-manager` Subelements

Element	Required	Description
<code>manager-properties</code>	zero or one	Specifies session manager properties.
<code>store-properties</code>	zero or one	Specifies session persistence (storage) properties.

**Attributes**

The following table describes attributes for the `session-manager` element.

**Table 2-43**    `session-manager` Attributes

Attribute	Default Value	Description
<code>persistence-type</code>	<code>memory</code>	(optional) Specifies the session persistence mechanism. Allowed values are <code>memory</code> , <code>file</code> , and <code>ha</code> .  For production environments that require session persistence, use <code>ha</code> .  The <code>custom</code> value is not implemented and should not be used.

---

**NOTE**      Setting the web container’s `persistence-type` to `ha` is not sufficient for configuring session persistence. You must also enable the `availability-service`. For more information about session persistence, see the *Sun Java System Application Server Developer’s Guide to Web Applications*.

---



# manager-properties

Specifies session manager properties.

## Subelements

The following table describes subelements for the `manager-properties` element.

**Table 2-44** `manager-properties` Subelements

Element	Required	Description
<code>property</code>	zero or more	Specifies a property, which has a name and a value.

## Attributes

none

## Properties

The following table describes properties for the `manager-properties` element.

**Table 2-45** `manager-properties` Properties

Property Name	Default Value	Description
<code>reapIntervalSeconds</code>	60	<p>Specifies the number of seconds between checks for expired sessions.</p> <p>If the <code>persistence-type</code> attribute of the <code>session-manager</code> element is <code>file</code> or <code>ha</code>, sessions are passivated if <code>maxSessions</code> has been exceeded.</p> <p>If <code>persistenceFrequency</code> is set to time-based, active sessions are stored at this interval.</p> <p>You should set this value lower than the frequency at which session data changes. For example, this value should be as low as possible (1 second) for a hit counter servlet on a frequently accessed website, or you could lose the last few hits each time you restart the server.</p>
<code>maxSessions</code>	-1	<p>Specifies the maximum number of sessions that can be in cache, or -1 for no limit. After this, an attempt to create a new session causes an <code>IllegalStateException</code> to be thrown.</p> <p>If the <code>persistence-type</code> attribute of the <code>session-manager</code> element is <code>file</code> or <code>ha</code>, the session manager passivates sessions to the persistent store when this maximum is reached.</p>

**Table 2-45** manager-properties Properties (Continued)

Property Name	Default Value	Description
sessionFilename	none; state is not preserved across restarts	<p>Specifies the absolute or relative path to the directory in which the session state is preserved between application restarts, if preserving the state is possible. A relative path is relative to the temporary directory for this web application.</p> <p>Applicable only if the persistence-type attribute of the session-manager element is memory.</p>
persistenceFrequency	web-method	<p>Specifies how often the session state is stored. Allowed values are as follows:</p> <ul style="list-style-type: none"><li>web-method - The session state is stored at the end of each web request prior to sending a response back to the client. This mode provides the best guarantee that the session state is fully updated in case of failure.</li><li>time-based - The session state is stored in the background at the frequency set by reapIntervalSeconds. This mode provides less of a guarantee that the session state is fully updated. However, it can provide a significant performance improvement because the state is not stored after each request.</li></ul> <p>Applicable only if the persistence-type attribute of the session-manager element is ha.</p>

store-properties

Specifies session persistence (storage) properties.

Subelements

The following table describes subelements for the store-properties element.

**Table 2-46** store-properties Subelements

Element	Required	Description
property	zero or more	Specifies a property, which has a name and a value.

Attributes

none

Properties

The following table describes properties for the store-properties element.

**Table 2-47** store-properties Properties

Property Name	Default Value	Description
directory	<i>instance_dir/generated/jsp/j2ee-apps/appname/appname_war</i>	<p>Specifies the absolute or relative pathname of the directory into which individual session files are written. A relative path is relative to the temporary work directory for this web application.</p> <p>Applicable only if the persistence-type attribute of the <a href="#">session-manager</a> element is file.</p>
persistenceScope	session	<p>Specifies how much of the session state is stored. Allowed values are as follows:</p> <ul style="list-style-type: none"> <li>• <code>session</code> - The entire session state is stored every time. This mode provides the best guarantee that your session data is correctly stored for any distributable web application.</li> <li>• <code>modified-session</code> - The entire session state is stored if it has been modified. A session is considered to have been modified if <code>HttpSession.setAttribute()</code> or <code>HttpSession.removeAttribute()</code> was called. You must guarantee that <code>setAttribute()</code> is called every time an attribute is changed. This is not a J2EE specification requirement, but it is required for this mode to work properly.</li> <li>• <code>modified-attribute</code> - Only modified session attributes are stored. For this mode to work properly, you must follow some guidelines, which are explained immediately following this table.</li> </ul> <p>Applicable only if the persistence-type attribute of the <a href="#">session-manager</a> element is ha.</p>

If the persistenceScope store property is set to modified-attribute, your web application must follow these guidelines:

- Call `setAttribute()` every time you modify the session state.
- Make sure there are no cross-references between attributes. The object graph under each distinct attribute key is serialized and stored separately. If there are any object cross references between the objects under each separate key, they are not serialized and deserialized correctly.
- Distribute the session state across multiple attributes, or at least between a read-only attribute and a modifiable attribute.

**NOTE**

The session persistence scope `modified-attribute` is not certified as a full production quality feature. You should evaluate the performance and stability of the web container under expected peak load with this persistence scope. If exceptions are logged or response time is too high, do not use this persistence scope for your production environment. Use `session` or `modified-session` instead.

## session-properties

Specifies session properties.

**Subelements**  
The following table describes subelements for the `session-properties` element.

**Table 2-48** `session-properties` Subelements

Element	Required	Description
<code>property</code>	zero or more	Specifies a property, which has a name and a value.

**Attributes**  
`none`

**Properties**  
The following table describes properties for the `session-properties` element.

**Table 2-49** `session-properties` Properties

Property Name	Default Value	Description
<code>timeoutSeconds</code>	600	<p>Specifies the default maximum inactive interval (in seconds) for all sessions created in this web module. If set to 0 or less, sessions in this web module never expire.</p> <p>If a <code>session-timeout</code> element is specified in the <code>web.xml</code> file, the <code>session-timeout</code> value overrides any <code>timeoutSeconds</code> value. If neither <code>session-timeout</code> nor <code>timeoutSeconds</code> is specified, the <code>timeoutSeconds</code> default is used.</p> <p>Note that the <code>session-timeout</code> element in <code>web.xml</code> is specified in minutes, not seconds.</p>
<code>enableCookies</code>	true	Uses cookies for session tracking if set to <code>true</code> .

**Table 2-49** session-properties Properties (*Continued*)

Property Name	Default Value	Description
enableURLRewriting	true	Enables URL rewriting. This provides session tracking via URL rewriting when the browser does not accept cookies. You must also use an <code>encodeURL</code> or <code>encodeRedirectURL</code> call in the servlet or JavaServer Pages™ (JSP™) page.
idLengthBytes	128	Specifies the number of bytes in this web module's session ID.

## J2EE Service Elements

J2EE service elements are as follows:

- `jms-service`
- `log-service`
- `security-service`
- `auth-realm`
- `transaction-service`

### jms-service

Configures the built-in Java Message Service (JMS) that is managed by the Sun Java System Application Server.

#### Subelements

The following table describes subelements for the `jms-service` element.

**Table 2-50** `jms-service` Subelements

Element	Required	Description
<code>property</code>	zero or more	Specifies a property or a variable.

#### Attributes

The following table describes attributes for the `jms-service` element.

**Table 2-51** `jms-service` Attributes

Attribute	Default	Description
<code>port</code>	7676	(optional) Specifies the port number used by the JMS provider.
<code>admin-user-name</code>	admin	Specifies the administrator user name for the JMS provider.
<code>admin-password</code>	admin	Specifies the administrator password for the JMS provider.
<code>init-timeout-in-seconds</code>	60	(optional) Specifies the amount of time the server instance waits at startup for the corresponding JMS instance to respond. If there is no response, startup is aborted. If set to 0, the server instance waits indefinitely.
<code>start-args</code>	none	(optional) Specifies the string of arguments supplied for startup of the corresponding JMS instance.
<code>log-level</code>	Value of <code>level</code> attribute of <code>log-service</code> element	(optional) Controls the type of messages logged by this element to the server log. For details, see the description of the <code>level</code> attribute of the <code>log-service</code> element.
<code>enabled</code>	true	(optional) If set to <code>true</code> , the Sun Java System Application Server instance is responsible for starting up and shutting down the JMS provider. If set to <code>false</code> , the Sun Java System Application Server instance does not start up nor shut down the JMS provider (either because the JMS provider is not used or because it is managed independently of the Sun Java System Application Server).

### Properties

The following table describes properties for the `jms-service` element.

**Table 2-52** `jms-service` Properties

Property	Default	Description
<code>instance-name</code>	<i>domain_instance</i>	Specifies the full Sun Java™ System Message Queue broker instance name, which by default is a concatenation of the domain and server instance names. For example: <code>domain1_server1</code> .
<code>instance-name-suffix</code>	none	Specifies a suffix to add to the full Sun Java System Message Queue broker instance name. The suffix is separated from the instance name by an underscore character ( <code>_</code> ). For example, if the instance name is <code>domain1_server1</code> , appending the suffix <code>xyz</code> changes the instance name to <code>domain1_server1_xyz</code> .

**Table 2-52** `jms-service` Properties (*Continued*)

Property	Default	Description
<code>append-version</code>	<code>false</code>	If <code>true</code> , appends the major and minor version numbers, preceded by underscore characters ( <code>_</code> ), to the full Sun Java System Message Queue broker instance name. For example, if the instance name is <code>domain1_server1</code> , appending the version numbers changes the instance name to <code>domain1_server1_7_0</code> .

## log-service

Configures the system logging service, which includes the following log files:

- The **server log** file stores messages from the default virtual server. Messages from other configured virtual servers also go here, unless the `log-file` attribute is explicitly specified in the `virtual-server-class` or `virtual-server` element. The default name is `server.log`.
- The **access log** file stores HTTP access messages from the default virtual server. The default name is `access.log`. To configure the access log, you use server application functions in the `init.conf` and `obj.conf` files. For more information, see the *Sun Java System Application Server Developer's Guide to NSAPI*.
- The **transaction log** files store transaction messages from the default virtual server. The default name of the directory for these files is `tx`.
- A **virtual server log** file stores messages from a `virtual-server-class` or `virtual-server` element that has an explicitly specified `log-file` attribute.

## Subelements

The following table describes subelements for the `log-service` element.

**Table 2-53** `log-service` Subelements

Element	Required	Description
<code>property</code>	zero or more	Specifies a property or a variable.

## Attributes

The following table describes attributes for the `log-service` element.

**Table 2-54** log-service Attributes

Attribute	Default	Description
file	server.log  in the directory specified by the log-root attribute of the <a href="#">server</a> element	(optional) Overrides the name or location of the server log. The file and directory in which the server log is kept must be writable by whatever user account the server runs as.  If you specify an absolute path, this value overrides the log-root attribute of the <a href="#">server</a> element.  If you specify a relative path, it is relative to the log-root attribute of the <a href="#">server</a> element. If no log-root value is specified, it is relative to <i>instance_dir/config</i> .
level	INFO	(optional) Controls the default type of messages logged by other elements to the server log. Many other elements can override this default in their log-level attributes. Allowed values are, from highest to lowest: FINEST, FINER, FINE, CONFIG, INFO, WARNING, SEVERE, ALERT, FATAL.  Each value logs all messages for all lower values; for example, FINEST logs all messages, and FATAL logs only FATAL messages. The default value is SEVERE, which logs all SEVERE, ALERT, and FATAL messages.
log-stdout	true	(optional) If true, redirects stdout output to the server log.
log-stderr	true	(optional) If true, redirects stderr output to the server log.
echo-log-messages-to-stderr	true	(optional) If true, sends log messages to stderr in addition to the server log.
log-virtual-server-id	false	(optional) If true, virtual server IDs are displayed in the virtual server logs. This is useful if multiple <a href="#">virtual-server</a> elements share the same log file.
use-system-logging	false	(optional) If true, uses the UNIX syslog service to produce and manage logs.

**security-service**

Defines parameters and configuration information needed by the J2EE security service.

**Subelements**

The following table describes subelements for the `security-service` element.

**Table 2-55** security-service Subelements

Element	Required	Description
<a href="#">auth-realm</a>	one or more	Defines a realm for authentication.



**Table 2-55** security-service Subelements (*Continued*)

Element	Required	Description
<a href="#">property</a>	zero or more	Specifies a property or a variable.

### Attributes

The following table describes attributes for the security-service element.

**Table 2-56** security-service Attributes

Attribute	Default	Description
default-realm	file	(optional) Specifies the active authentication realm (an auth-realm name attribute) for this server instance.
default-principal	none	(optional) Used as the identity of the default security context when necessary and when no principal is provided. This attribute need not be set for normal server operation.
default-principal-password	none	(optional) The password of the default principal. This attribute need not be set for normal server operation.
anonymous-role	ANYONE	(optional) Used as the name for default, or anonymous, role. The anonymous role is always assigned to all principals. This role value can be used in J2EE deployment descriptors to grant access to anyone.
audit-enabled	false	(optional) If true, additional access logging is performed to provide audit information. Audit information consists of: <ul style="list-style-type: none"> <li>• Authentication success and failure events</li> <li>• Servlet and EJB access grants and denials</li> </ul>
log-level	Value of level attribute of <a href="#">log-service</a> element	(optional) Controls the type of messages logged by this element to the server log. For details, see the description of the level attribute of the <a href="#">log-service</a> element.

## auth-realm

Defines a realm for authentication.

Authentication realms require provider-specific properties, which vary depending on what a particular implementation needs.

For more information about how to define realms, see the *Sun Java System Application Server Developer's Guide*.

Here is an example of the default `file` realm:

```
<auth-realm name="file"
  classname="com.iplanet.ias.security.auth.realm.file.FileRealm">
  <property name="file" value="instance_dir/config/keyfile"/>
  <property name="jaas-context" value="fileRealm"/>
</auth-realm>
```

Which properties an `auth-realm` element uses depends on the value of the `auth-realm` element's `name` attribute. The `file` realm uses `file` and `jaas-context` properties. Other realms use different properties.

### Subelements

The following table describes subelements for the `auth-realm` element.

**Table 2-57** `auth-realm` Subelements

Element	Required	Description
<code>property</code>	zero or more	Specifies a property or a variable.

### Attributes

The following table describes attributes for the `auth-realm` element.

**Table 2-58** `auth-realm` Attributes

Attribute	Default	Description
<code>name</code>	<code>none</code>	Specifies the name of this realm.
<code>classname</code>	<code>none</code>	Specifies the Java class that implements this realm.

### Properties

The standard realms provided with Sun Java System Application Server have required and optional properties. A custom realm may have different properties.

The following table describes properties for the `auth-realm` element.

**Table 2-59** `auth-realm` Properties

Property	Realms	Description
<code>jaas-context</code>	<code>file</code> , <code>ldap</code> , <code>solaris</code>	Specifies the JAAS (Java Authentication and Authorization Service) context.
<code>file</code>	<code>file</code>	Specifies the file that stores user names. The default is <code>instance_dir/config/keyfile</code> .
<code>assign-groups</code>	<code>certificate</code>	(optional) If this property is set, its value is taken to be a comma-separated list of group names. All clients who present valid certificates are assigned membership to these groups for the purposes of authorization decisions in the web and EJB containers.
<code>directory</code>	<code>ldap</code>	Specifies the LDAP URL to your server.
<code>base-dn</code>	<code>ldap</code>	Specifies the LDAP base DN for the location of user data. This base DN can be at any level above the user data, since a tree scope search is performed. The smaller the search tree, the better the performance.
<code>search-filter</code>	<code>ldap</code>	(optional) Specifies the search filter to use to find the user. The default is <code>uid=%s</code> ( <code>%s</code> expands to the subject name).
<code>group-base-dn</code>	<code>ldap</code>	(optional) Specifies the base DN for the location of groups data. By default it is same as the <code>base-dn</code> , but it can be tuned if necessary.
<code>group-search-filter</code>	<code>ldap</code>	(optional) Specifies the search filter to find group memberships for the user. The default is <code>uniquemember=%d</code> ( <code>%d</code> expands to the user element DN).
<code>group-target</code>	<code>ldap</code>	(optional) Specifies the LDAP attribute name that contains group name entries. The default is <code>CN</code> .
<code>search-bind-dn</code>	<code>ldap</code>	(optional) Specifies an optional DN used to authenticate to the directory for performing the <code>search-filter</code> lookup. Only required for directories that do not allow anonymous search.
<code>search-bind-password</code>	<code>ldap</code>	(optional) Specifies the LDAP password for the DN given in <code>search-bind-dn</code> .

## transaction-service

Configures the Java Transaction Service (JTS).

### Subelements

The following table describes subelements for the `transaction-service` element.

**Table 2-60** `transaction-service` Subelements

Element	Required	Description
<code>property</code>	zero or more	Specifies a property or a variable.

### Attributes

The following table describes attributes for the `transaction-service` element.

**Table 2-61** `transaction-service` Attributes

Attribute	Default	Description
<code>automatic-recovery</code>	<code>false</code>	(optional) If <code>true</code> , the server instance attempts transaction recovery during startup.
<code>timeout-in-seconds</code>	<code>0</code>	(optional) Specifies the amount of time after which the transaction is aborted. If set to <code>0</code> , the transaction never times out.
<code>tx-log-dir</code>	directory specified by the <code>log-root</code> attribute of the <code>server</code> element	(optional) Overrides the location of the transaction log directory. The directory in which the transaction logs are kept must be writable by whatever user account the server runs as. See the <code>log-service</code> description for details about logs.
<code>heuristic-decision</code>	<code>rollback</code>	(optional) During recovery, if the outcome of a transaction cannot be determined from the logs, this property determines the outcome. Allowed values are <code>rollback</code> and <code>commit</code> .
<code>keypoint-interval</code>	<code>2048</code>	(optional) Specifies the number of transactions between keypoint operations in the log. Keypoint operations reduce the size of the transaction log file by compressing it. A larger value for this attribute (for example, <code>4096</code> ) results in a larger transaction log file, but fewer keypoint operations and potentially better performance. A smaller value (for example, <code>100</code> ) results in smaller log files, but slightly reduced performance due to the greater frequency of keypoint operations.
<code>log-level</code>	Value of <code>level</code> attribute of <code>log-service</code> element	(optional) Controls the type of messages logged by this element to the server log. For details, see the description of the <code>level</code> attribute of the <code>log-service</code> element.
<code>monitoring-enabled</code>	<code>false</code>	(optional) Determines whether monitoring of the JTS is enabled.

## Properties

The following table describes properties for the `transaction-service` element.

**Table 2-62** `transaction-service` Properties

Property	Default	Description
<code>oracle-xa-recovery-workaround</code>	false	If true, the Oracle XA Resource workaround is used in transaction recovery.
<code>sybase-xa-recovery-workaround</code>	false	If true, the Sybase XA Resource workaround is used in transaction recovery.
<code>disable-distributed-transaction-logging</code>	false	If true, disables transaction logging, which may improve performance. If the <code>automatic-recovery</code> attribute is set to true, this property is ignored.
<code>xaresource-txn-timeout</code>	specific to the XAResource used	Changes the XAResource timeout. In some cases, the XAResource default timeout can cause transactions to be aborted, so it is desirable to change it.

# Java Configuration Elements

Java configuration elements are as follows:

- `java-config`
- `profiler`
- `jvm-options`

## java-config

Specifies Java Virtual Machine (JVM) configuration parameters.

### Subelements

The following table describes subelements for the `java-config` element.

**Table 2-63** `java-config` Subelements

Element	Required	Description
<code>profiler</code>	zero or one	Configures a profiler for use with Sun Java System Application Server.
<code>jvm-options</code>	zero or more	Contains JVM command line options.

**Table 2-63** java-config Subelements (*Continued*)

Element	Required	Description
<a href="#">property</a>	zero or more	Specifies a property or a variable.

**Attributes**

The following table describes attributes for the java-config element.

**Table 2-64** java-config Attributes

Attribute	Default	Description
java-home	none	The path to the directory where the JDK is installed.
debug-enabled	false	(optional) If true, the server starts up in debug mode ready for attachment with a JPDA-based debugger.
debug-options	-Xdebug -Xrunjdpw:transport=dt_socket,server=y,suspend=n	(optional) Specifies JPDA (Java Platform Debugger Architecture) options. A list of debugging options that you can include is available here: <a href="http://java.sun.com/products/jpda/doc/conninv.html#Invocation">http://java.sun.com/products/jpda/doc/conninv.html#Invocation</a>  For more information about debugging, see the <i>Sun Java System Application Server Developer's Guide</i> .
rmic-options	-iiop -poa -alwaysgenerate -keepgenerated -g	(optional) Specifies options passed to the RMI compiler at application deployment time. The -keepgenerated option saves generated source for stubs and ties.  For more information about the rmic command, see the <i>Sun Java System Application Server Developer's Guide to Enterprise JavaBeans Technology</i> .
javac-options	-g	(optional) Specifies options passed to the Java compiler at application deployment time.
classpath-prefix	none	(optional) Specifies a prefix for the system classpath. You should only prefix the system classpath if you wish to override system classes, such as the XML parser classes. Use this attribute with caution.
server-classpath	none	(optional) Specifies the classpath for the environment from which the server was started. This classpath can be accessed using <code>System.getProperty("java.class.path")</code> .

**Table 2-64** java-config Attributes (*Continued*)

Attribute	Default	Description
classpath-suffix	none	(optional) Specifies a suffix for the system classpath.
native-library-path-prefix	none	(optional) Specifies a prefix for the native library path.  The native library path is the automatically constructed concatenation of the Application Server installation relative path for its native shared libraries, the standard JRE native library path, the shell environment setting ( <code>LD_LIBRARY_PATH</code> on UNIX), and any path specified in the <code>profiler</code> element. Since this is synthesized, it does not appear explicitly in the server configuration.
native-library-path-suffix	none	(optional) Specifies a suffix for the native library path.
bytecode-preprocessors		(optional) A comma separated list of class names, each of which must implement the <code>com.sun.appserv.BytecodePreprocessor</code> interface. Each of the specified preprocessor classes is called in the order specified.
env-classpath-ignored	true	(optional) If <code>false</code> , the <code>CLASSPATH</code> environment variable is read and appended to the Sun Java System Application Server classpath. The <code>CLASSPATH</code> environment variable is added after the <code>classpath-suffix</code> , at the very end.  For a development environment, this value should be set to <code>false</code> . For a production environment, this value should be set to <code>true</code> to prevent environment variable side effects.

## profiler

Configures a profiler for use with Sun Java System Application Server. For more information about profilers, see the *Sun Java System Application Server Developer's Guide*.

### Subelements

The following table describes subelements for the `profiler` element.

**Table 2-65** profiler Subelements

Element	Required	Description
<code>jvm-options</code>	zero or more	Contains profiler-specific JVM command line options.

**Table 2-65** profiler Subelements (*Continued*)

Element	Required	Description
property	zero or more	Specifies a property or a variable.

**NOTE** Subelements of a profiler element can occur in any order.

**Attributes**

The following table describes attributes for the profiler element.

**Table 2-66** profiler Attributes

Attribute	Default	Description
name	none	Specifies the name of the profiler.
classpath	none	(optional) Specifies the classpath for the profiler.
native-library-path	none	(optional) Specifies the native library path for the profiler.
enabled	true	(optional) Determines whether the profiler is enabled.

**jvm-options**

Contains JVM command line options, for example:

```
<jvm-options>-Xdebug -Xmx128m</jvm-options>
```

For information about the options you can use, see:

<http://java.sun.com/docs/hotspot/VMOptions.html>

**Subelements**

none

**Attributes**

none



# Resource Elements

Resource elements are as follows:

- `resources`
- `custom-resource`
- `external-jndi-resource`
- `jdbc-resource`
- `mail-resource`
- `jms-resource`
- `persistence-manager-factory-resource`
- `jdbc-connection-pool`

## resources

Contains configured resources, such as database connections, JavaMail™ sessions, and so on.

**NOTE**

You must specify a Java Naming and Directory Interface™ (JNDI) name for each resource. To avoid collisions with names of other enterprise resources in JNDI, and to avoid portability problems, all names in a Sun Java System Application Server application should begin with the string `java:comp/env`.

## Subelements

The following table describes subelements for the `resources` element.

**Table 2-67** `resources` Subelements

Element	Required	Description
<code>custom-resource</code>	zero or more	Defines a custom resource.
<code>external-jndi-resource</code>	zero or more	Defines a resource that resides in an external JNDI repository.
<code>jdbc-resource</code>	zero or more	Defines a JDBC (Java Database Connectivity) resource.
<code>mail-resource</code>	zero or more	Defines a JavaMail resource.
<code>jms-resource</code>	zero or more	Defines a JMS resource.

**Table 2-67** resources Subelements (*Continued*)

Element	Required	Description
<code>persistence-manager-factory-resource</code>	zero or more	Defines a persistence manager factory resource for CMP.
<code>jdbc-connection-pool</code>	zero or more	Defines the properties that are required for creating a JDBC connection pool.

**NOTE** Subelements of a `resources` element can occur in any order.

**Attributes**

none

**custom-resource**

Defines a custom resource, which specifies a custom server-wide resource object factory. Such object factories implement the `javax.naming.spi.ObjectFactory` interface.

**Subelements**

The following table describes subelements for the `custom-resource` element.

**Table 2-68** custom-resource Subelements

Element	Required	Description
<code>description</code>	zero or one	Contains a text description of this element.
<code>property</code>	zero or more	Specifies a property or a variable.

**Attributes**

The following table describes attributes for the `custom-resource` element.

**Table 2-69** custom-resource Attributes

Attribute	Default	Description
<code>jndi-name</code>	none	Specifies the JNDI name for the resource.
<code>res-type</code>	none	Specifies the fully qualified type of the resource.
<code>factory-class</code>	none	Specifies the fully qualified name of the user-written factory class, which implements <code>javax.naming.spi.ObjectFactory</code> .

**Table 2-69** custom-resource Attributes (*Continued*)

Attribute	Default	Description
enabled	true	(optional) Determines whether this resource is enabled at runtime.

## external-jndi-resource

Defines a resource that resides in an external JNDI repository. For example, a generic Java object could be stored in an LDAP server. An external JNDI factory must implement the `javax.naming.spi.InitialContextFactory` interface.

### Subelements

The following table describes subelements for the `external-jndi-resource` element.

**Table 2-70** external-jndi-resource Subelements

Element	Required	Description
<a href="#">description</a>	zero or one	Contains a text description of this element.
<a href="#">property</a>	zero or more	Specifies a property or a variable.

### Attributes

The following table describes attributes for the `external-jndi-resource` element.

**Table 2-71** external-jndi-resource Attributes

Attribute	Default	Description
jndi-name	none	Specifies the JNDI name for the resource.
jndi-lookup-name	none	Specifies the JNDI lookup name for the resource.
res-type	none	Specifies the fully qualified type of the resource.
factory-class	none	Specifies the fully qualified name of the factory class, which implements <code>javax.naming.spi.InitialContextFactory</code> .  For more information about JNDI, see the <i>Sun Java System Application Server Developer's Guide to J2EE Services and APIs</i> .
enabled	true	(optional) Determines whether this resource is enabled at runtime.

# jdbc-resource

Defines a JDBC (`javax.sql.DataSource`) resource.

## Subelements

The following table describes subelements for the `jdbc-resource` element.

**Table 2-72**    `jdbc-resource` Subelements

Element	Required	Description
<a href="#">description</a>	zero or one	Contains a text description of this element.
<a href="#">property</a>	zero or more	Specifies a property or a variable.

## Attributes

The following table describes attributes for the `jdbc-resource` element.

**Table 2-73**    `jdbc-resource` Attributes

Attribute	Default	Description
<code>jndi-name</code>	<code>none</code>	Specifies the JNDI name for the resource.
<code>pool-name</code>	<code>none</code>	Specifies the name of the associated JDBC connection pool, defined in a <a href="#">jdbc-connection-pool</a> element.
<code>enabled</code>	<code>true</code>	(optional) Determines whether this resource is enabled at runtime.

# mail-resource

Defines a JavaMail (`javax.mail.Session`) resource.

## Subelements

The following table describes subelements for the `mail-resource` element.

**Table 2-74**    `mail-resource` Subelements

Element	Required	Description
<a href="#">description</a>	zero or one	Contains a text description of this element.
<a href="#">property</a>	zero or more	Specifies a property or a variable.

## Attributes

The following table describes attributes for the `mail-resource` element.

**Table 2-75** `mail-resource` Attributes

Attribute	Default	Description
<code>jndi-name</code>	<code>none</code>	Specifies the JNDI name for the resource.
<code>store-protocol</code>	<code>imap</code>	(optional) Specifies the storage protocol service, which connects to a mail server, retrieves messages, and saves messages in folder(s). Example values are <code>imap</code> and <code>pop3</code> .
<code>store-protocol-class</code>	<code>com.sun.mail.imap.IMAPStore</code>	(optional) Specifies the service provider implementation class for storage.
<code>transport-protocol</code>	<code>smtp</code>	(optional) Specifies the transport protocol service, which sends messages.
<code>transport-protocol-class</code>	<code>com.sun.mail.smtp.SMTPTransport</code>	(optional) Specifies the service provider implementation class for transport.
<code>host</code>	<code>none</code>	The mail server host name.
<code>user</code>	<code>none</code>	The mail server user name.
<code>from</code>	<code>none</code>	The e-mail address the mail server uses to indicate the message sender.
<code>debug</code>	<code>false</code>	(optional) Determines whether debugging for this resource is enabled.
<code>enabled</code>	<code>true</code>	(optional) Determines whether this resource is enabled at runtime.

## Properties

You can set properties for the `mail-resource` element and then get these properties in a `JavaMail Session` object later. Every property name must start with a `mail-` prefix. Sun Java System Application Server changes the dash (`-`) character to a period (`.`) in the name of the property and saves the property to the `MailConfiguration` and `JavaMail Session` objects. If the name of the property doesn't start with `mail-`, the property is ignored.

For example, if you want to define the property `mail.password` in a `JavaMail Session` object, first edit `server.xml` as follows:

```
...
<mail-resource jndi-name="mail/Session" ...>
  <property name="mail-password" value="adminadmin"/>
</mail-resource>
...
```

After you get the `JavaMail Session` object, you can get the `mail.password` property to retrieve the value `adminadmin`, as follows:

```
String password = session.getProperty("mail.password");
```

## jms-resource

Defines a JMS (Java Message Service) resource.

### Subelements

The following table describes subelements for the `jms-resource` element.

**Table 2-76**    `jms-resource` Subelements

Element	Required	Description
<a href="#">description</a>	zero or one	Contains a text description of this element.
<a href="#">property</a>	zero or more	Specifies a property or a variable.

### Attributes

The following table describes attributes for the `jms-resource` element.

**Table 2-77**    `jms-resource` Attributes

Attribute	Default	Description
<code>jndi-name</code>	<code>none</code>	Specifies the JNDI name for the resource.
<code>res-type</code>	<code>none</code>	Specifies the fully qualified type of the resource, which is one of the following:  <code>javax.jms.Topic</code> <code>javax.jms.Queue</code> <code>javax.jms.TopicConnectionFactory</code> <code>javax.jms.QueueConnectionFactory</code>
<code>enabled</code>	<code>true</code>	(optional) Determines whether this resource is enabled at runtime.

## Properties

The following table describes the most commonly used JMS properties for the `jms-resource` element. For a complete list of the available properties (called *administered object attributes* in Sun Java System Message Queue), see the *Sun Java System Message Queue Administration Guide*.

**Table 2-78** `jms-resource` Properties

Property	Default	Description
<code>imqDestinationName</code>	none	Specifies the JMS physical destination name associated with this JMS resource. You must specify this property for <code>jms-resource</code> elements with the <code>res-type</code> of <code>javax.jms.Topic</code> or <code>javax.jms.Queue</code> .  The <i>Sun Java System Message Queue Administrator's Guide</i> shows a default value for this property, but this does not apply in the Sun Java System Application Server environment.
<code>imqBrokerHostName</code>	the same host name as the Sun Java System Application Server instance (localhost)	Specifies the host name where the JMS service (Sun Java System Message Queue broker) is running. For <code>jms-resource</code> elements with the <code>res-type</code> of <code>javax.jms.TopicConnectionFactory</code> or <code>javax.jms.QueueConnectionFactory</code> .
<code>imqBrokerHostPort</code>	the <code>jms-service</code> element's port attribute	Specifies the port where the JMS service (Sun Java System Message Queue broker) is running. For <code>jms-resource</code> elements with the <code>res-type</code> of <code>javax.jms.TopicConnectionFactory</code> or <code>javax.jms.QueueConnectionFactory</code> .
<code>imqConfiguredClientID</code>	none	Specifies the JMS Client Identifier to be associated with a Connection created using the <code>createQueueConnection</code> and <code>createTopicConnection</code> JMS APIs of the <code>QueueConnectionFactory</code> and <code>TopicConnectionFactory</code> classes, respectively.  For <code>jms-resource</code> elements with the <code>res-type</code> of <code>javax.jms.TopicConnectionFactory</code> or <code>javax.jms.QueueConnectionFactory</code> .  Durable subscription names are unique and only valid within the scope of a client identifier. To create or reactivate a durable subscriber, the connection must have a valid client identifier. The JMS specification ensures that client identifiers are unique and that a given client identifier is allowed to be used by only one active connection at a time.

## persistence-manager-factory-resource

Defines a persistence manager factory resource for container-managed persistence (CMP).

### Subelements

The following table describes subelements for the `persistence-manager-factory-resource` element.

**Table 2-79** `persistence-manager-factory-resource` Subelements

Element	Required	Description
<a href="#">description</a>	zero or one	Contains a text description of this element.
<a href="#">property</a>	zero or more	Specifies a property or a variable.

### Attributes

The following table describes attributes for the `persistence-manager-factory-resource` element.

**Table 2-80** `persistence-manager-factory-resource` Attributes

Attribute	Default	Description
<code>jndi-name</code>	<code>none</code>	Specifies the JNDI name for the resource.
<code>factory-class</code>	<code>com.sun.jdo.spi.persistence.support.sqlstore.impl.PersistenceManagerFactoryImpl</code>	(optional) Specifies the name of the factory class. This attribute supports third party CMP persistence manager factories. Use the name required by the third party CMP implementation. Do not specify this attribute for the built-in CMP implementation.
<code>jdbc-resource-jndi-name</code>	<code>none</code>	(optional) Specifies the <a href="#">jdbc-resource</a> from which database connections are obtained. Must be the <code>jndi-name</code> of an existing <code>jdbc-resource</code> .
<code>enabled</code>	<code>true</code>	(optional) Determines whether this resource is enabled at runtime.

## jdbc-connection-pool

Defines the properties that are required for creating a JDBC connection pool.

<b>TIP</b>	You can create a pool definition and then copy, paste, and edit it to configure multiple JDBC data sources.
------------	---



## Subelements

The following table describes subelements for the `jdbc-connection-pool` element.

**Table 2-81** `jdbc-connection-pool` Subelements

Element	Required	Description
<code>description</code>	zero or one	Contains a text description of this element.
<code>property</code>	zero or more	Specifies a property or a variable.

## Attributes

The following table describes attributes for the `jdbc-connection-pool` element.

**Table 2-82** `jdbc-connection-pool` Attributes

Attribute	Default	Description
<code>name</code>	none	Specifies the name of the connection pool. A <code>jdbc-resource</code> element's <code>pool-name</code> attribute refers to this name.
<code>datasource-classname</code>	none	Specifies the class name of the associated vendor-supplied data source. This class must implement <code>java.sql.DataSource</code> or <code>java.sql.XADataSource</code> or both.
<code>res-type</code>	<code>javax.sql.DataSource</code>	(optional) Specifies the interface the data source class implements. The value of this attribute can be <code>javax.sql.DataSource</code> or <code>javax.sql.XADataSource</code> . If the value is not one of these interfaces, the default is used. An error occurs if this attribute has a legal value and the indicated interface is not implemented by the data source class.
<code>steady-pool-size</code>	8	(optional) Specifies the initial and minimum number of connections maintained in the pool.
<code>max-pool-size</code>	32	(optional) Specifies the maximum number of connections that can be created to satisfy client requests.
<code>max-wait-time-in-millis</code>	60000	(optional) Specifies the amount of time, in milliseconds, that the caller is willing to wait for a connection. If 0, the caller is blocked indefinitely until a resource is available or an error occurs.

**Table 2-82** jdbc-connection-pool Attributes (*Continued*)

Attribute	Default	Description
pool-resize-quantity	2	(optional) Specifies the number of connections to be destroyed if the existing number of connections is above the <code>steady-pool-size</code> (subject to the <code>max-pool-size</code> limit). This is enforced periodically at the <code>idle-time-out-in-seconds</code> interval. An idle connection is one that has not been used for a period of <code>idle-time-out-in-seconds</code> .
idle-timeout-in-seconds	300	(optional) Specifies the maximum time that a connection can remain idle in the pool. After this amount of time, the pool can close this connection.
transaction-isolation-level	default JDBC driver isolation level	<p>(optional) Specifies the transaction isolation level on the pooled database connections. Allowed values are <code>read-uncommitted</code>, <code>read-committed</code>, <code>repeatable-read</code>, or <code>serializable</code>.</p> <p>Applications that change the isolation level on a pooled connection programmatically risk polluting the pool, which can lead to errors. See <code>is-isolation-level-guaranteed</code> for more details.</p>
is-isolation-level-guaranteed	true	(optional) Applicable only when <code>transaction-isolation-level</code> is explicitly set. If true, every connection obtained from the pool is guaranteed to have the desired isolation level. This may impact performance on some JDBC drivers. You can set this attribute to <code>false</code> if you are certain that the hosted applications do not return connections with altered isolation levels.
is-connection-validation-required	false	(optional) Specifies whether connections have to be validated before being given to the application. If a resource's validation fails, it is destroyed, and a new resource is created and returned.
connection-validation-method	auto-commit	<p>(optional) Legal values are as follows:</p> <ul style="list-style-type: none"> <li>• <code>auto-commit</code> (default), which uses <code>Connection.setAutoCommit(Connection.getAutoCommit())</code></li> <li>• <code>meta-data</code>, which uses <code>Connection.getMetaData()</code></li> <li>• <code>table</code>, which performs a query on a table specified in the <code>validation-table-name</code> attribute</li> </ul>

**Table 2-82** jdbc-connection-pool Attributes (*Continued*)

Attribute	Default	Description
validation-table-name	none	(optional) Specifies the table name to be used to perform a query to validate a connection. This parameter is mandatory if and only if connection-validation-type is set to table.
fail-all-connections	false	(optional) If true, closes all connections in the pool if a single validation check fails. This parameter is mandatory if and only if is-connection-validation-required is set to true.

### Properties

Most JDBC 2.0 drivers allow use of standard property lists to specify the user, password, and other resource configuration information. Although properties are optional with respect to Sun Java System Application Server, some properties may be necessary for most databases. For details, see Section 5.3 of JDBC 2.0 Standard Extension API.

When properties are specified, they are passed to the vendor's data source class (specified by the datasource-classname attribute) as is using setName(value) methods.

The user and password properties are used as the default principal if container managed authentication is specified and a default-resource-principal is not found in the application deployment descriptors.

The following table describes some common properties for the jdbc-connection-pool element.

**Table 2-83** jdbc-connection-pool Properties

Property	Description
user	Specifies the user name for this connection pool.
password	Specifies the password for this connection pool.
databaseName	Specifies the database for this connection pool.
serverName	Specifies the database server for this connection pool.
port	Specifies the port on which the database server listens for requests.
networkProtocol	Specifies the communication protocol.
roleName	Specifies the initial SQL role name.
datasourceName	Specifies an underlying XADataSource, or a ConnectionPoolDataSource if connection pooling is done.

**Table 2-83** jdbc-connection-pool Properties (*Continued*)

Property	Description
description	Specifies a text description.
url	Specifies the URL for this connection pool. Although this is not a standard property, it is commonly used.

# Application Elements

Application elements are as follows:

- applications
- lifecycle-module
- j2ee-application
- web-module
- ejb-module
- connector-module

## applications

Contains deployed J2EE applications, J2EE modules, and Lifecycle modules.

### Subelements

The following table describes subelements for the applications element.

**Table 2-84** applications Subelements

Element	Required	Description
lifecycle-module	zero or more	Specifies a deployed lifecycle module.
j2ee-application	zero or more	Specifies a deployed J2EE application.
ejb-module	zero or more	Specifies a deployed EJB module.
web-module	zero or more	Specifies a deployed web module.
connector-module	zero or more	Specifies a deployed connector module.

**NOTE** Subelements of an applications element can occur in any order.

## Attributes

The following table describes attributes for the `applications` element.

**Table 2-85** `applications` Attributes

Attribute	Default	Description
<code>dynamic-reload-enabled</code>	<code>false</code>	(optional) Specifies whether dynamic reloading is enabled. This attribute should be set to <code>true</code> in a development environment and <code>false</code> in a production environment.
<code>dynamic-reload-poll-interval-in-seconds</code>	<code>2</code>	(optional) Specifies the interval at which applications and modules are checked for code changes and dynamically reloaded.

## lifecycle-module

Specifies a deployed lifecycle module. For more information about lifecycle modules, see the *Sun Java System Application Server Developer's Guide*.

## Subelements

The following table describes subelements for the `lifecycle-module` element.

**Table 2-86** `lifecycle-module` Subelements

Element	Required	Description
<a href="#">description</a>	zero or one	Contains a text description of this element.
<a href="#">property</a>	zero or more	Specifies a property or a variable.

## Attributes

The following table describes attributes for the `lifecycle-module` element.

**Table 2-87** `lifecycle-module` Attributes

Attribute	Default	Description
<code>name</code>	<code>none</code>	The name of the lifecycle module.
<code>class-name</code>	<code>none</code>	The fully qualified name of the lifecycle module's class file, which must implement the <code>com.sun.appserv.server.LifecycleListener</code> interface.

**Table 2-87** lifecycle-module Attributes (*Continued*)

Attribute	Default	Description
classpath	value of application-root attribute of server element	(optional) The classpath for the lifecycle module. Specifies where the module is located.
load-order	none	(optional) Determines the order in which lifecycle modules are loaded at startup. Modules with smaller integer values are loaded sooner. Values can range from 101 to the operating system's MAXINT. Values from 1 to 100 are reserved.
is-failure-fatal	false	(optional) Determines whether the server is shut down if the lifecycle module fails.
enabled	true	(optional) Determines whether the lifecycle module is enabled.

## j2ee-application

Specifies a deployed J2EE application.

### Subelements

The following table describes subelements for the j2ee-application element.

**Table 2-88** j2ee-application Subelements

Element	Required	Description
description	zero or one	Contains a text description of this element.

### Attributes

The following table describes attributes for the j2ee-application element.

**Table 2-89** j2ee-application Attributes

Attribute	Default	Description
name	none	The name of the application.
location	none	The location of the application in the Sun Java System Application Server file system.
virtual-servers	all virtual servers	(optional) The virtual servers to which the web modules within this application are deployed.
enabled	true	(optional) Determines whether the application is enabled.

**Table 2-89** j2ee-application Attributes (*Continued*)

Attribute	Default	Description
availability-enabled	Value of same attribute in <a href="#">web-container</a> and <a href="#">ejb-container</a> elements	(optional) Determines whether availability is enabled at the application level. For more information about availability levels, see the <a href="#">availability-service</a> element description.

## ejb-module

Specifies a deployed EJB module.

### Subelements

The following table describes subelements for the `ejb-module` element.

**Table 2-90** ejb-module Subelements

Element	Required	Description
<a href="#">description</a>	zero or one	Contains a text description of this element.

### Attributes

The following table describes attributes for the `ejb-module` element.

**Table 2-91** ejb-module Attributes

Attribute	Default	Description
name	none	The name of the EJB module.
location	none	The location of the EJB module in the Sun Java System Application Server file system.
enabled	true	(optional) Determines whether the EJB module is enabled.
availability-enabled	Value of same attribute in <a href="#">ejb-container</a> and <a href="#">j2ee-application</a> elements	(optional) Determines whether availability is enabled at the EJB module level. For more information about availability levels, see the <a href="#">availability-service</a> element description.

## web-module

Specifies a deployed web module.

### Subelements

The following table describes subelements for the `web-module` element.

**Table 2-92** `web-module` Subelements

Element	Required	Description
<code>description</code>	zero or one	Contains a text description of this element.

### Attributes

The following table describes attributes for the `web-module` element.

**Table 2-93** `web-module` Attributes

Attribute	Default	Description
<code>name</code>	<code>none</code>	The name of the web module.
<code>context-root</code>	<code>empty string</code>	<p>The context root (context path without the <code>/</code> in front) at which the web module is installed (see Section 5.4 of the Java™ Servlet 2.3 specification).</p> <p>If this attribute is an empty string, which is the default value, this web module can be the default web module for the <code>virtual-server</code> elements specified in the <code>virtual-servers</code> attribute.</p> <p>If multiple web modules assigned to a virtual server have an empty context root, one of them is loaded, and the rest generate error messages such as: Virtual server [<i>virtual_server</i>] already has a web module loaded at [/].</p> <p>If you are setting up load balancing, web module context roots must be unique within a cluster. See the <i>Sun Java System Application Server Administrator's Guide</i> for more information about load balancing.</p>
<code>location</code>	<code>none</code>	<p>A fully qualified or relative path to the directory to which the contents of the <code>.war</code> file have been extracted. If relative, it is relative to the following directory:</p> <p><i>instance_dir/applications/j2ee-modules/</i></p>
<code>virtual-servers</code>	<code>all virtual servers</code>	(optional) The <code>virtual-server</code> elements to which the web module is deployed.



**Table 2-93** web-module Attributes (*Continued*)

Attribute	Default	Description
enabled	true	(optional) Determines whether the web module is enabled.
availability-enabled	Value of same attribute in <a href="#">web-container</a> and <a href="#">j2ee-application</a> elements	(optional) Determines whether availability is enabled at the web module level. For more information about availability levels, see the <a href="#">availability-service</a> element description.

## connector-module

Specifies a deployed connector module.

### Subelements

The following table describes subelements for the `connector-module` element.

**Table 2-94** connector-module Subelements

Element	Required	Description
<a href="#">description</a>	zero or one	Contains a text description of this element.

### Attributes

The following table describes attributes for the `connector-module` element.

**Table 2-95** connector-module Attributes

Attribute	Default	Description
name	none	The name of the connector module.
location	none	The location of the connector module in the Sun Java System Application Server file system.
enabled	true	(optional) Determines whether the connector module is enabled.

# User Database Selection

---

**NOTE** The user database applies only to the security of the server itself. It is not related to J2EE application and module security.

---

The `auth-db` element in `server.xml` selects a user database for the parent `virtual-server` element as follows:

- The `auth-db` element's `id` attribute maps to an ACL file's database attribute.
- The `auth-db` element's database attribute maps to a `dbswitch.conf` entry.

This layer between the ACL file and the `dbswitch.conf` file gives the server administrator full control over which databases virtual server administrators and users have access to.

---

**NOTE** The `dbswitch.conf` interface is Unstable. An unstable interface may be experimental or transitional, and hence may change incompatibly, be removed, or be replaced by a more stable interface in the next release.

---

The `dbswitch.conf` file establishes the root of the search tree for LDAP databases as follows:

- The base DN in the LDAP URL in `dbswitch.conf` defines a root object for all further DN specifications. So, for most new installations, it can be empty, because the final base DN is determined in other ways -- either through a DC tree lookup or an explicit `basedn` value in the `auth-db` element.
- A `dbswitch.conf` attribute for LDAP databases, `dcsuffix`, defines the root of the DC tree. This root is relative to the base DN in the LDAP URL. You can use `dcsuffix` if the database is *schema compliant*. Requirements for schema compliance are listed in [“The Sun Java System LDAP Schema” on page 91](#).

A user database is selected for a virtual server as follows:

- If a `virtual-server` has no `auth-db` subelement, user- or group-based ACLs fail.
- When no database attribute is present in a virtual server's ACL definition, the `virtual-server` must have an `auth-db` subelement with an `id` attribute of default. The database attribute of the `auth-db` then points to a database in `dbswitch.conf`. If no database attribute is present, default is used.

- If an LDAP database is schema compliant, the base DN of the access is computed using a DC tree lookup of the `virtual-server` element's `hosts` attribute that matches the client-supplied `Host` header. If no `hosts` attribute matches, the `server-name` attribute of the parent `http-listener` is used. The DC tree lookup is based at the `dcsuffix` DN. The result must contain an `inetDomainBaseDN` attribute that contains the base DN. This base DN is taken as is and is not relative to any of the base DN values.
- If the `basedn` attribute of the `auth-db` element is not present and the database is not schema compliant, the accesses happen relative to the base DN in the `dbswitch.conf` entry, as in previous Application Server versions.

## The Sun Java System LDAP Schema

You can use the `dcsuffix` attribute in the `dbswitch.conf` file if your LDAP database meets the requirements outlined in this section.

The subtree rooted at an ISP entry (for example, `o=isp`) is called the *convergence tree*. It contains all the directory data related to organizations (customers) served by an ISP.

The subtree rooted at `o=internet` is called the *domain component tree* or *dc tree*. It contains a sparse DNS tree with entries for the customer domains served. These entries are links to the appropriate location in the convergence tree where the data for that domain is located.

The directory tree may be single rooted, (for example, `o=root` may have `o=isp` and `o=internet` under it), or have two separate roots, one for the convergence tree and one for the dc tree.

## The Convergence Tree

The top level of the convergence tree must have one organization entry for each customer (or organization), and one for the ISP itself.

Underneath each organization, there must be two `organizationalUnit` entries: `ou=People` and `ou=Groups`. A third, `ou=Devices`, can be present if device data is to be stored for the organization.

Each user entry must have a unique `uid` value within a given organization. The namespace under this subtree can be partitioned into various `ou` entries that aggregate user entries in convenient groups (for example, `ou=eng`, `ou=corp`). User `uid` values must still be unique within the entire `People` subtree.

User entries in the convergence tree are of type `inetOrgPerson`. The `cn`, `sn`, and `uid` attributes must be present. The `uid` attribute must be a valid e-mail name (specifically, it must be a valid local-part as defined in RFC822). The `cn` should contain *name initial sn*. The RDN of the user entry should be the `uid` value. User entries must contain the auxiliary class `inetUser` if they are to be considered enabled for service or valid.

User entries can also contain the auxiliary class `inetSubscriber`, which is used for account management purposes. If an `inetUserStatus` attribute is present in an entry and has a value of `inactive` or `deleted`, the entry is ignored.

Groups are located under the `Groups` subtree and consist of LDAP entries of type `groupOfUniqueNames`.

## The Domain Component (dc)Tree

The `dc` tree contains hierarchical domain entries, each of which is a DNS name component.

Entries that represent the domain name of a customer are overlaid with the LDAP auxiliary class `inetDomain`. For example, the two LDAP entries

`dc=customer1,dc=com,o=Internet,o=root` and

`dc=customer2,dc=com,o=Internet,o=root` contain the `inetDomain` class, but `dc=com,o=Internet,o=root` does not. The latter is present only to provide structure to the tree.

Entries with an `inetDomain` attribute are called virtual domains. These must have the attribute `inetDomainBasedN` filled with the DN of the top level organization entry where the data of this domain is stored in the convergence tree. For example, the virtual domain entry in `dc=cust2,dc=com,o=Internet,o=root` would contain the attribute `inetDomainBasedN` with value `o=Cust2,o=isp,o=root`.

If an `inetDomainStatus` attribute is present in an entry and has a value of `inactive` or `deleted`, the entry is ignored.

## Variables

Some variables are defined in `server.xml` for use in the `obj.conf` file. The following file fragment defines a `docroot` variable:

```
<property name="docroot" value="/server/docs/class2/acme" />
```

A `docroot` variable allows different document root directories to be assigned for different virtual servers. The variable is then used in the `obj.conf` file. For example:

```
NameTrans fn=document-root root="$docroot"
```

Using this `docroot` variable allows you to define different document roots for different virtual servers within the same virtual server class.

---

**NOTE** A variable must be defined in the `server.xml` file at the `http-service`, `virtual-server-class`, or `virtual-server` level. You should define variables with default values at the `http-service` or `virtual-server-class` level and override them at the `virtual-server` level.

---

## Format of a Variable

A variable is found in `obj.conf` when the following regular expression matches:

```
\$[A-Za-z][A-Za-z0-9_]*
```

This expression represents a `$` followed by one or more alphanumeric characters. A delimited version (“`${property}`”) is not supported. To get a regular `$` character, use `$$` to have variable substitution.

## The id Variable

A special variable, `id`, is always available within a `virtual-server` element and refers to the value of the `id` attribute. It is predefined and cannot be overridden. The `id` attribute uniquely identifies a virtual server. For example:

```
<property name=docroot value="/export/$id" />
```

If the `id` attribute of the parent `virtual-server` element is `myserver`, the `docroot` variable is set to the value `/export/myserver`.

## Other Important Variables

The following variables are used in various parts of the Sun Java System Application Server configuration. Unlike the `$id` variable, they are not predefined in the server, and they can be overridden.

### General Variables

The following table lists general `server.xml` variables.

**Table 2-96** General Variables

Variable	Description
docroot	The document root of the virtual server. Typically evaluated as the parameter to the <code>document-root</code> function in the <code>obj.conf</code> file.
accesslog	The access log file for a virtual server.

## send-cgi Variables

The following table lists `server.xml` variables used by the `send-cgi` function in the `obj.conf` file.

**Table 2-97** send-cgi Variables

Variable	Description
user	The value of the <code>user</code> CGI parameter.
group	The value of the <code>group</code> CGI parameter.
chroot	The value of the <code>chroot</code> CGI parameter.
dir	The value of the <code>dir</code> CGI parameter.
nice	The value of the <code>nice</code> CGI parameter.

For more information about the `send-cgi` function, see the *Sun Java System Application Server Developer's Guide to NSAPI*.

## Variable Evaluation

Variables are evaluated when generating specific objsets for individual virtual servers. Evaluation is recursive: variable values can contain other variables. For example:

```
...
<virtual-server-class>
  ...
  <virtual-server ...>
    ...
    <property name=docroot value="$docrootbase/nonjava/$id" />
  </virtual-server>
</virtual-server ...>
```

```

...
    <property name=docroot value="$docrootbase/java/$id" />
</virtual-server>
...
    <property name=docrootbase value="/export" />
</virtual-server-class>
...

```

Variables in subelements override variables in the parent elements. For example, it is possible to set a variable for a class of virtual servers and override it with a definition of the same variable in an individual virtual server.

## Sample server.xml File

When you first install Sun Java System Application Server, the `server.xml` file looks like this:

```

<?xml version="1.0" encoding="UTF-8"?>

<!-- server_7_0_dtd version: 1.1.2.39 -->

<!DOCTYPE server PUBLIC "-//Sun Microsystems Inc.//DTD Sun Java System Application Server
7.0//EN" "file:///C:/Sun/AppServer7/lib/dtds/sun-server_1_1.dtd">

<server name="server1" log-root="C:/Sun/AppServer7/domains/domain1/server1/logs"
application-root="C:/Sun/AppServer7/domains/domain1/server1/applications"
session-store="C:/Sun/AppServer7/domains/domain1/server1/session-store">

    <http-service qos-enabled="false">

        <http-listener id="http-listener-1" address="0.0.0.0" port="1024"
acceptor-threads="1" blocking-enabled="false" security-enabled="false"
default-virtual-server="server1" server-name="goliath" enabled="true"/>

        <mime id="mime1" file="mime.types"/>

        <acl id="acl1"
file="C:/Sun/AppServer7/domains/domain1/server1/config/generated.server1.acl"/>

        <virtual-server-class id="defaultclass" config-file="obj.conf"
default-object="default" accept-language="false" enabled="true">

            <virtual-server id="server1" http-listeners="http-listener-1" hosts="goliath"
mime="mime1" state="on" accept-language="false">

                <auth-db id="default" database="default"/>

                <property name="dir" value=""/>

                <property name="nice" value=""/>

```

```

        <property name="user" value="" />
        <property name="group" value="" />
        <property name="chroot" value="" />
        <property name="docroot"
value="C:/Sun/AppServer7/domains/domain1/server1/docroot"/>
        <property name="accesslog"
value="C:/Sun/AppServer7/domains/domain1/server1/logs/access"/>
    </virtual-server>
</virtual-server-class>
</http-service>
<iiop-service>
    <orb message-fragment-size="1024" steady-thread-pool-size="10"
max-thread-pool-size="200" idle-thread-timeout-in-seconds="300" max-connections="1024"
monitoring-enabled="false"/>
    <iiop-listener id="orb-listener-1" address="goliath" port="3700" enabled="true">
    </iiop-listener>
</iiop-service>
<web-container monitoring-enabled="false" >
</web-container>
    <ejb-container steady-pool-size="32" pool-resize-quantity="16" max-pool-size="64"
cache-resize-quantity="32" max-cache-size="512" pool-idle-timeout-in-seconds="600"
cache-idle-timeout-in-seconds="600" removal-timeout-in-seconds="5400"
victim-selection-policy="nru" commit-option="B" monitoring-enabled="false">
    </ejb-container>
    <mdb-container steady-pool-size="10" pool-resize-quantity="2" max-pool-size="60"
idle-timeout-in-seconds="600" monitoring-enabled="false">
    </mdb-container>
    <jms-service port="7676" admin-user-name="admin" admin-password="admin"
init-timeout-in-seconds="30" enabled="true">
    </jms-service>
    <log-service file="C:/Sun/AppServer7/domains/domain1/server1/logs/server.log"
level="INFO" log-stdout="true" log-stderr="true" echo-log-messages-to-stderr="true"
create-console="true" log-virtual-server-id="false" use-system-logging="false">
    </log-service>

```



```

    <security-service default-realm="file" anonymous-role="ANYONE" audit-enabled="false">
        <auth-realm name="file"
classname="com.iplanet.ias.security.auth.realm.file.FileRealm">
            <property name="file"
value="C:/Sun/AppServer7/domains/domain1/server1/config/keyfile"/>
            <property name="jaas-context" value="fileRealm"/>
        </auth-realm>
        <auth-realm name="ldap"
classname="com.iplanet.ias.security.auth.realm.ldap.LDAPRealm">
            <property name="directory" value="ldap://localhost:389"/>
            <property name="base-dn" value="o=isp"/>
            <property name="jaas-context" value="ldapRealm"/>
        </auth-realm>
        <auth-realm name="certificate"
classname="com.iplanet.ias.security.auth.realm.certificate.CertificateRealm">
        </auth-realm>
        <!--
        <auth-realm name="solaris"
classname="com.iplanet.ias.security.auth.realm.solaris.SolarisRealm">
            <property name="jaas-context" value="solarisRealm"/>
        </auth-realm>
        -->
    </security-service>

    <transaction-service automatic-recovery="false" timeout-in-seconds="0"
tx-log-dir="C:/Sun/AppServer7/domains/domain1/server1/logs" heuristic-decision="rollback"
keypoint-interval="2048" monitoring-enabled="false">
    </transaction-service>

    <java-config java-home="C:/Sun/AppServer7/jdk"
server-classpath="C:/Sun/AppServer7/lib/appserv-rt.jar;C:/Sun/AppServer7/jdk/lib/tools.jar;C:/Sun/AppServer7/lib/appserv-ext.jar;C:/Sun/AppServer7/lib/appserv-cmp.jar;C:/Sun/AppServer7/lib/appserv-ideplugin.jar;C:\Sun\AppServer7\imq\lib\imq.jar;C:\Sun\AppServer7\imq\lib\jaxm-api.jar;C:\Sun\AppServer7\imq\lib\imqadmin.jar;C:\Sun\AppServer7\imq\lib\imqutil.jar;C:/Sun/AppServer7/lib/appserv-admin.jar;C:\Sun\AppServer7\share\lib\jaxrpc-impl.jar;C:\Sun\AppServer7\share\lib\jaxrpc-api.jar;C:\Sun\AppServer7\share\lib\jaxr-impl.jar;C:\Sun\AppServer7\share\lib\jaxr-api.jar;C:\Sun\AppServer7\share\lib\jhall.jar;C:\Sun\AppServ

```

```

er7\share\lib\activation.jar;C:\Sun\AppServer7\share\lib\mail.jar;C:\Sun\AppServer7\share\lib\saaj-api.jar;C:\Sun\AppServer7\share\lib\saaj-impl.jar;C:\Sun\AppServer7\share\lib\commons-logging.jar;C:\Sun\AppServer7\img\lib\fscontext.jar;C:\Sun\AppServer7\img\lib\providerutil.jar;C:\Sun\AppServer7\lib\appserv-jstl.jar"
classpath-suffix="C:\Sun\AppServer7\pointbase\client_tools\lib\pbclient42RE.jar"
env-classpath-ignored="true" debug-options="-Xdebug
-Xrunjdpw:transport=dt_socket,server=y,suspend=n" debug-enabled="false"
javac-options="-g">

    <jvm-options>-Dorg.xml.sax.parser=org.xml.sax.helpers.XMLReaderAdapter
</jvm-options>

    <jvm-options>-Dorg.xml.sax.driver=org.apache.crimson.parser.XMLReaderImpl
</jvm-options>

<jvm-options>-Djava.security.policy=C:/Sun/AppServer7/domains/domain1/server1/config/server.policy</jvm-options>

<jvm-options>-Djava.security.auth.login.config=C:/Sun/AppServer7/domains/domain1/server1/config/login.conf</jvm-options>

    <jvm-options>-Dcom.sun.jdo.api.persistence.model.multipleClassLoaders=reload
</jvm-options>

<jvm-options>-Djava.util.logging.manager=com.ipplanet.ias.server.logging.ServerLogManager</jvm-options>

    <jvm-options>-Dcom.sun.aas.configRoot=C:/Sun/AppServer7/config</jvm-options>
    <jvm-options>-Dcom.sun.aas.imgLib=C:\Sun\AppServer7\img\lib</jvm-options>
    <jvm-options>-Dcom.sun.aas.imgBin=C:\Sun\AppServer7\img\bin</jvm-options>

<jvm-options>-Dcom.sun.aas.webServicesLib=C:\Sun\AppServer7\share\lib</jvm-options>
    <jvm-options>-Dsun.rmi.dgc.server.gcInterval=300000</jvm-options>
<jvm-options> -Xrs -Xms128m -Xmx256m </jvm-options>
</java-config>

<resources>
</resources>

    <applications dynamic-reload-enabled="false"
dynamic-reload-poll-interval-in-seconds="2">
</applications>

```

```
</server>
```

Sample server.xml File

# Syntax and Use of init.conf

When the Sun Java System Application Server starts up, it looks in a file called `init.conf` in the `instance_dir/config` directory to establish a set of global variable settings that affect the server instance's behavior and configuration. Sun Java System Application Server executes all the directives defined in `init.conf`.

Except for the `Init` functions, the directives in `init.conf` specify a variable and a value, for example:

```
TempDir /tmp
```

The order of the directives is not important.

---

<b>NOTE</b>	Directives noted with boolean values have the following equivalent values: <code>on/yes/true</code> and <code>off/no/false</code> .
-------------	---

---

---

<b>NOTE</b>	When you edit the <code>init.conf</code> file, you must restart the server for the changes to take effect.
-------------	--

---

---

<b>NOTE</b>	The <code>init.conf</code> interface is Unstable. An unstable interface may be experimental or transitional, and hence may change incompatibly, be removed, or be replaced by a more stable interface in the next release.
-------------	--

---

This chapter lists the global settings that can be specified in `init.conf` in Sun Java System Application Server 7.

The categories are:

- [Init Functions](#)
- [Server Information](#)
- [DNS Lookup](#)
- [Threads, Processes and Connections](#)
- [Native Thread Pools](#)
- [CGI](#)
- [Error Logging](#)
- [ACL](#)
- [Security](#)
- [Chunked Encoding](#)
- [Miscellaneous](#)

For an alphabetical list of directives, see [Appendix C, “Alphabetical List of Directives in `init.conf`.”](#)

---

<b>NOTE</b>	Much of the functionality of the file cache is controlled by a configuration file called <code>nsfc.conf</code> . For information about <code>nsfc.conf</code> , see <a href="#">“<code>nsfc.conf</code>” on page 131</a> and the <i>Sun Java System Application Server Performance Tuning Guide</i> .
-------------	--

---

## Init Functions

The `Init` functions load and initialize server modules and plug-ins, and they initialize log files. For more information about these functions, see the *Sun Java System Application Server Developer’s Guide to NSAPI*.

# Server Information

This sub-section lists the directives in `init.conf` that specify information about the server. They are:

- [NetSiteRoot](#)
- [TempDir](#)
- [TempDirSecurity](#)
- [User \(UNIX only\)](#)

## NetSiteRoot

Specifies the absolute pathname to the top-level directory under which the server's `bin` and `lib` directories can be found. There is no default value; the file must specify a value.

### Syntax

`NetSiteRoot path`

## TempDir

Specifies the directory on the local volume that the server uses for its temporary files. On UNIX, this directory must be owned by, and writable by, the user the server runs as. See also the directives [User \(UNIX only\)](#) and [TempDirSecurity](#).

### Syntax

`TempDir path`

### Default

`/tmp (UNIX)`

## TempDirSecurity

Determines whether the server checks if the `TempDir` directory is secure. On UNIX, specifying `TempDirSecurity off` allows the server to use `/tmp` as a temporary directory.

---

**CAUTION** Specifying `TempDirSecurity off` or using `/tmp` as a temporary directory on UNIX is highly discouraged. Using `/tmp` as a temporary directory opens a number of potential security risks.

---

**Syntax**

`TempDirSecurity [on|off]`

**Default**

on

**User (UNIX only)**

The `User` directive specifies the UNIX user account for the server. If the server is started by the superuser or root user, the server binds to the Port you specify and then switches its user ID to the user account specified with the `User` directive. This directive is ignored if the server isn't started as `root`. The user account you specify should have *read* permission to the server's root and subdirectories. The user account should have write access to the `logs` directory and execute permissions to any CGI programs. The user account should not have write access to the configuration files. This ensures that in the unlikely event that someone compromises the server, they won't be able to change configuration files and gain broader access to your machine. Use of the `nobody` user may not provide adequate security.

**Syntax**

`User name`

`name` is the 8-character (or less) login name for the UNIX user account.

**Default**

If there is no `User` directive, the server runs with the user account it was started with.

**Examples**

`User http`

`User server`

`User nobody`

## DNS Lookup

This section lists the directives in `init.conf` that affect DNS lookup. The directives are:

- [AsyncDNS](#)
- [DNS](#)



## AsyncDNS

Specifies whether asynchronous DNS is used. The `DNS` directive must be set to `on` for this directive to take effect. The value is either `on` or `off`. If DNS is enabled, enabling asynchronous DNS improves server performance.

### Default

The default is `off`.

## DNS

The `DNS` directive specifies whether the server performs DNS lookups on clients that access the server. When a client connects to your server, the server knows the client's IP address but not its host name (for example, it knows the client as 198.95.251.30, rather than its host name `www.a.com`). The server will resolve the client's IP address into a host name for operations like access control, CGI, error reporting, and access logging.

If your server responds to many requests per day, you might want (or need) to stop host name resolution; doing so can reduce the load on the DNS or NIS server.

### Syntax

`DNS [on|off]`

### Default

DNS host name resolution is `off` as a default.

### Example

`DNS on`

# Threads, Processes and Connections

In Sun Java System Application Server 7, acceptor threads on a listen socket accept connections and put them onto a connection queue. Session threads then pick up connections from the queue and service the requests. The session threads post more session threads if required at the end of the request. The policy for adding new threads is based on the connection queue state:

- Each time a new connection is returned, the number of connections waiting in the queue (the backlog of connections) is compared to the number of session threads already created. If it is greater than the number of threads, more threads are scheduled to be added the next time a request completes.

- The previous backlog is tracked, so that if it is seen to be increasing over time, and if the increase is greater than the `ThreadIncrement` value, and the number of session threads minus the backlog is less than the `ThreadIncrement` value, then another `ThreadIncrement` number of threads are scheduled to be added.
- The process of adding new session threads is strictly limited by the `RqThrottle` value.
- To avoid creating too many threads when the backlog increases suddenly (such as the startup of benchmark loads), the decision whether more threads are needed is made only once every 16 or 32 times a connection is made based on how many session threads already exist.

This subsection lists the directives in `init.conf` that affect the number and timeout of threads, processes, and connections. They are:

- [ConnQueueSize](#)
- [HeaderBufferSize](#)
- [IOTimeout](#)
- [KeepAliveThreads](#)
- [KeepAliveTimeout](#)
- [ListenQ](#)
- [MaxKeepAliveConnections](#)
- [PostThreadsEarly](#)
- [RcvBufSize](#)
- [RqThrottle](#)
- [RqThrottleMin](#)
- [SndBufSize](#)
- [StackSize](#)
- [StrictHttpHeaders](#)
- [TerminateTimeout](#)
- [ThreadIncrement](#)

Also see the section “[Native Thread Pools](#)” on page 110 for directives for controlling the pool of native threads.

## ConnQueueSize

Specifies the number of outstanding (yet to be serviced) connections that the application server can have. This value should always be greater than the operating system limit for the maximum number of open file descriptors per process.

### Default

The default value is 5000.

## HeaderBufferSize

The size (in bytes) of the buffer used by each of the request processing threads for reading the request data from the client. The maximum number of request processing threads is controlled by the [RqThrottle](#) setting.

### Default

The default value is 8192 (8 KB).

## IOTimeout

Specifies the number of seconds the server waits for data to arrive from the client. If data does not arrive before the timeout expires then the connection is closed. By setting it to less than the default 30 seconds, you can free up threads sooner. However, you may also disconnect users with slower connections.

### Syntax

IOTimeout *seconds*

### Default

30 seconds for servers that don't use hardware encryption devices and 300 seconds for those that do.

## KeepAliveThreads

This directive determines the number of threads in the keep-alive subsystem. This number should be a small multiple of the number of processors on the system. (for example, a 2 CPU system should have 2 or 4 keep alive threads). The maximum number of keep-alive connections allowed ([MaxKeepAliveConnections](#)) should also be taken into consideration when choosing a value for this setting.

### Default

1

## **KeepAliveTimeout**

This directive determines the maximum time that the server holds open an HTTP Keep-Alive connection or a persistent connection between the client and the server. The Keep-Alive feature for earlier versions of the server allows the client/server connection to stay open while the server processes the client request. The default connection is a persistent connection that remains open until the server closes it or the connection has been open for longer than the time allowed by `KeepAliveTimeout`.

The timeout countdown starts when the connection is handed over to the keep-alive subsystem. If there is no activity on the connection when the timeout expires, the connection is closed.

### **Default**

The default value is 30 seconds. The maximum value is 300 seconds (5 minutes).

## **ListenQ**

Specifies the maximum number of pending connections on a listen socket. Connections that time out on a listen socket whose backlog queue is full will fail.

### **Default**

The default value is 128.

## **MaxKeepAliveConnections**

Specifies the maximum number of Keep-Alive and persistent connections that the server can have open simultaneously. Values range from 0 to 32768.

### **Default**

256

## **PostThreadsEarly**

If this directive is set to 1 (on), the server checks the whether the minimum number of threads are available at a listen socket after accepting a connection but before sending the response to the request. Use this directive when the server will be handling requests that take a long time to handle, such as those that do long database connections.

### **Default**

0 (off)

## RcvBufSize

Specifies the size (in bytes) of the receive buffer used by sockets. Allowed values are determined by the operating system.

### Default

The default value is determined by the operating system. Typical defaults are 4096 (4K), 8192 (8K).

## RqThrottle

Specifies the maximum number of request processing threads that the server can handle simultaneously. Each request runs in its own thread.

There is additional discussion of this and other server configuration and performance tuning issues in the *Sun Java System Application Server Performance Tuning Guide*.

### Default

128

## RqThrottleMin

Specifies the number of request processing threads that are created when the server is started. As the load on the server increases, more request processing threads are created (up to a maximum of `RqThrottle` threads).

### Default

48

## SndBufSize

Specifies the size (in bytes) of the send buffer used by sockets.

### Default

The default value is determined by the operating system. Typical defaults are 4096 (4K), 8192 (8K).

## StackSize

Determines the maximum stack size for each request handling thread.

### Default

The most favorable machine-specific stack size.

## StrictHttpHeaders

Controls strict HTTP header checking. If strict HTTP header checking is on, the server rejects connections that include inappropriately duplicated headers.

### Syntax

StrictHttpHeaders [on|off]

### Default

off

## TerminateTimeout

Specifies the time that the server waits for all existing connections to terminate before it shuts down.

### Default

30 seconds

## ThreadIncrement

The number of additional or new request processing threads created to handle an increase in the load on the server, for example when the number of pending connections (in the request processing queue) exceeds the number of idle request processing threads.

When a server starts up, it creates `RqThrottleMin` number of request processing threads. As the load increases, it creates `ThreadIncrement` additional request processing threads until `RqThrottle` request processing threads have been created.

### Default

The default value is 10.

# Native Thread Pools

This section lists the directives for controlling the size of the native thread pool. This thread pool consists entirely of native, OS-level threads. The native pool on UNIX is normally not engaged, as all threads are OS-level threads. Using native pools on UNIX may introduce a small performance overhead as they'll require an additional context switch; however, they can be used to localize the `stickyAttach` effect or for other purposes, such as resource control and management or to emulate single-threaded behavior for plug-ins.

You can specify `stickyAttach` as a property of the `java-config` element in the `server.xml` file as follows:

```
<java-config>
...
  <property name="stickyAttach" value="1" >
</java-config>
```

The directives are:

- [NativePoolStackSize](#)
- [NativePoolMaxThreads](#)
- [NativePoolMinThreads](#)
- [NativePoolQueueSize](#)

## NativePoolStackSize

Determines the stack size of each thread in the native thread pool.

### Default

0 (represents an operating-system-specific default value)

## NativePoolMaxThreads

Determines the maximum number of threads in the native thread pool.

### Default

128

## NativePoolMinThreads

Determines the minimum number of threads in the native thread pool.

### Default

1

## NativePoolQueueSize

Determines the number of threads that can wait in the queue for the thread pool. If all threads in the pool are busy, then the next request-handling thread that needs to use a thread in the native pool must wait in the queue. If the queue is full, the next request-handling thread that tries to get in the queue is rejected, with the result that it returns a busy response to the client. It is then free to handle another incoming request instead of being tied up waiting in the queue.

**Default**

0 (represents an unlimited queue size)

# CGI

This section lists the directives in `init.conf` that affect requests for CGI programs. The directives are:

- [CGIExpirationTimeout](#)
- [CGIStubIdleTimeout](#)
- [MaxCGIStubs](#)
- [MinCGIStubs](#)

## CGIExpirationTimeout

This directive specifies the maximum time in seconds that CGI processes are allowed to run before being killed.

The value of `CGIExpirationTimeout` should not be set too low - 300 seconds (5 minutes) would be a good value for most interactive CGIs; but if you have CGIs that are expected to take longer without misbehaving, then you should set it to the maximum duration you expect a CGI program to run normally. A value of 0 disables CGI expiration, which means that there is no time limit for CGI processes.

**Default**

0 (unlimited)

## CGIStubIdleTimeout

This directive causes the server to kill any CGIStub processes that have been idle for the number of seconds set by this directive. Once the number of processes is at `MinCGIStubs`, the server does not kill any more processes.

**Default**

30



## MaxCGIStubs

Controls the maximum number of CGIStub processes the server can spawn. This is the maximum concurrent CGIStub processes in execution, not the maximum number of pending requests. The default value should be adequate for most systems. Setting this too high may actually reduce throughput.

### Default

10

## MinCGIStubs

Controls the number of processes that are started by default. The first CGIStub process is not started until a CGI program has been accessed. Note that if you have an `init-cgi` directive in the `init.conf` file, the minimum number of CGIStub processes are spawned at startup. The value must be less than the `MaxCGIStubs` value.

### Default

2

# Error Logging

This section lists the directives in that affect error logging. They are:

- [ErrorLogDateFormat](#)
- [LogFlushInterval](#)
- [PidLog](#)

## ErrorLogDateFormat

The `ErrorLogDateFormat` directive specifies the date format that the server log uses.

### Syntax

`ErrorLogDateFormat` *format*

The *format* can be any format listed in [Appendix A, “Time Formats.”](#)

### Default

`%d/%b/%Y:%H:%M:%S`

## LogFlushInterval

This directive determines the log flush interval, in seconds, of the log flush thread.

### Default

30

## PidLog

`PidLog` specifies a file in which to record the process ID (pid) of the base server process. Some of the server support programs assume that this log is in the server root, in `logs/pid`.

To shut down your server, kill the base server process listed in the pid log file by using a `-TERM` signal. To tell your server to reread its configuration files and reopen its log files, use `kill` with the `-HUP` signal.

If the `PidLog` file isn't writable by the user account that the server uses, the server does not log its process ID anywhere. The server won't start if it can't log the process ID.

### Syntax

`PidLog file`

The *file* is the full path name and file name where the process ID is stored.

### Default

There is no default.

### Examples

```
PidLog /var/ns-server/logs/pid
```

```
PidLog /tmp/ns-server.pid
```

# ACL

This section lists the directives in `init.conf` relevant to access control lists (ACLs). They are:

- [ACLCacheLifetime](#)
- [ACLUserCacheSize](#)
- [ACLGroupCacheSize](#)

## ACLCacheLifetime

`ACLCacheLifetime` determines the number of seconds before cache entries expire. Each time an entry in the cache is referenced, its age is calculated and checked against `ACLCacheLifetime`. The entry is not used if its age is greater than or equal to the `ACLCacheLifetime`. If this value is set to 0, the cache is turned off.

If you use a large number for this value, you may need to restart the Sun Java System Application Server when you make changes to the LDAP entries. For example, if this value is set to 120 seconds, the Sun Java System Application Server might be out of sync with the LDAP server for as long as two minutes. If your LDAP is not likely to change often, use a large number.

### Default

120

## ACLUserCacheSize

`ACLUserCacheSize` determines the number of users in the User Cache.

### Default

200

## ACLGroupCacheSize

`ACLGroupCacheSize` determines how many group IDs can be cached for a single UID/cache entry.

### Default

4

# Security

This section lists the directives in `init.conf` that affect server access and security issues for Sun Java System Application Server. They are:

- [Security](#)
- [SSLCacheEntries](#)
- [SSLClientAuthDataLimit](#)
- [SSLClientAuthTimeout](#)
- [SSLSessionTimeout](#)

- [SSL3SessionTimeout](#)

## Security

The Security directive globally enables or disables SSL by making certificates available to the server instance. It must be `on` for virtual servers to use SSL. If set to `on`, the user is prompted for:

- The password to the trust database, which contains the server's private key(s)
- The PINs required by any installed cryptographic hardware

---

**NOTE** When you create a secure listen socket through the Administration interface, security is automatically turned on globally in `init.conf`. When you create a secure listen socket manually in `server.xml`, security must be turned on by editing `init.conf`.

---

For more information about enabling SSL for individual virtual servers, see [Chapter 2, “Server Configuration Files.”](#)

### Syntax

`Security [on|off]`

### Default

`off`

### Example

`Security off`

## SSLCacheEntries

Specifies the number of SSL sessions that can be cached. There is no upper limit.

### Syntax

`SSLCacheEntries number`

If the *number* is 0, the default value, which is 10000, is used.

## SSLClientAuthDataLimit

Specifies the maximum amount of application data, in bytes, that is buffered during the client certificate handshake phase.

**Default**

The default value is 1048576 (1 MB).

**SSLClientAuthTimeout**

Specifies the number of seconds after which the client certificate handshake phase times out.

**Default**

60

**SSLSessionTimeout**

The `SSLSessionTimeout` directive controls SSL2 session caching.

**Syntax**

`SSLSessionTimeout` *seconds*

The *seconds* value is the number of seconds until a cached SSL2 session becomes invalid. If the `SSLSessionTimeout` directive is specified, the value of seconds is silently constrained to be between 5 and 100 seconds.

**Default**

The default value is 100.

**SSL3SessionTimeout**

The `SSL3SessionTimeout` directive controls SSL3 session caching.

**Syntax**

`SSL3SessionTimeout` *seconds*

The *seconds* value is the number of seconds until a cached SSL3 session becomes invalid. The default value is 86400 (24 hours). If the `SSL3SessionTimeout` directive is specified, the value of seconds is silently constrained to be between 5 and 86400 seconds.

## Chunked Encoding

This section lists directives that control chunked encoding.

- [UseOutputStreamSize](#)
- [ChunkedRequestBufferSize](#)

- [ChunkedRequestTimeout](#)

These directives have equivalent Service SAF parameters in `obj.conf`. The `obj.conf` parameters override these directives. For more information, see the *Sun Java System Application Server NSAPI Developer's Guide*.

## UseOutputStreamSize

The `UseOutputStreamSize` directive determines the default output stream buffer size for the `net_read` and `netbuf_grab` NSAPI functions. For details about these functions, see the *Sun Java System Application Server NSAPI Developer's Guide*.

---

<b>NOTE</b>	The <code>UseOutputStreamSize</code> parameter can be set to zero in the <code>obj.conf</code> file to disable output stream buffering. For the <code>init.conf</code> file, setting <code>UseOutputStreamSize</code> to zero has no effect.
-------------	--

---

### Syntax

`UseOutputStreamSize size`

The *size* value is the number of bytes.

### Default

The default value is 8192 (8 KB).

## ChunkedRequestBufferSize

The `ChunkedRequestBufferSize` directive determines the default buffer size for “un-chunking” request data.

### Syntax

`ChunkedRequestBufferSize size`

The *size* value is the number of bytes.

### Default

The default value is 8192.

## ChunkedRequestTimeout

The `ChunkedRequestTimeout` directive determines the default timeout for “un-chunking” request data.

**Syntax**

`ChunkedRequestTimeout` *seconds*

The *seconds* value is the number of seconds.

**Default**

The default value is 60 (1 minute).

## Miscellaneous

This section lists miscellaneous other directives in `init.conf`.

- [ChildRestartCallback](#)
- [HTTPVersion](#)
- [MaxRqHeaders](#)
- [ReentrantTimeFunctions](#) (Solaris only)
- [Umask](#) (UNIX only)

### ChildRestartCallback

This directive forces the callback of NSAPI functions that were registered using the `daemon_atrestart` NSAPI function when the server is restarting or shutting down. Values are `on`, `off`, `yes`, `no`, `true`, or `false`. For details about `daemon_atrestart`, see the *Sun Java System Application Server NSAPI Developer's Guide*.

**Default**

`no`

### HTTPVersion

The current HTTP version used by the server in the form *m.n*, where *m* is the major version number and *n* the minor version number.

**Default**

The default value is `1.1`.

### MaxRqHeaders

Specifies the maximum number of header lines in a request. Values range from 0 to 512.

**Default**

64

**ReentrantTimeFunctions (Solaris only)**

The `ReentrantTimeFunctions` directive specifies whether the server should use its own reentrant time formatting implementation instead of the implementation provided by the operating system. Using the server's reentrant implementation may provide a small performance improvement on computers with a very large number of CPUs.

The default value, `off`, instructs the server to use the operating system's implementation.

**Default**

`off`

**Umask (UNIX only)**

This directive specifies the umask value used by the NSAPI functions `system_fopenWA` and `system_fopenRW` to open files in different modes. Valid values for this directive are standard UNIX umask values.

For more information on these functions, see `system_fopenWA` and `system_fopenRW` in the *Sun Java System Application Server NSAPI Developer's Guide*.



# MIME Types

This chapter discusses the MIME types file. The sections are:

- [Introduction](#)
- [Determining the MIME Type](#)
- [How the Type Affects the Response](#)
- [What Does the Client Do with the MIME Type?](#)
- [Syntax of the MIME Types File](#)
- [Sample MIME Types File](#)

The `mime.types` file interacts with the `obj.conf` file. For more information about `obj.conf`, see the *Sun Java System Application Server Developer's Guide to NSAPI*.

---

**NOTE** The `mime.types` interface is Unstable. An unstable interface may be experimental or transitional, and hence may change incompatibly, be removed, or be replaced by a more stable interface in the next release.

---

## Introduction

The MIME types file in the `instance_dir/config` directory contains mappings between MIME (Multipurpose Internet Mail Extensions) types and file extensions. For example, the MIME types file maps the extensions `.html` and `.htm` to the type `text/html`:

```
type=text/html exts=htm,html
```

When the Sun Java System Application Server receives a web request for a resource from a client, it uses the MIME type mappings to determine what kind of resource is being requested.

MIME types are defined by three attributes: language (`lang`), encoding (`enc`), and content-type (`type`). At least one of these attributes must be present for each type. The most commonly used attribute is `type`. The server frequently considers the `type` when deciding how to generate the response to the client. (The `enc` and `lang` attributes are rarely used.)

The default MIME types file is called `mime.types`.

## Determining the MIME Type

During the `ObjectType` step in the request handling process, the server determines the MIME type attributes of the resource requested by the client. Several different functions can be used to determine the MIME type, but the most commonly used one is `type-by-extension`. This function tells the server to look up the MIME type according to the requested resource's file extension in the MIME types table.

The directive in `obj.conf` that tells the server to look up the MIME type according to the extension is:

```
ObjectType fn=type-by-extension
```

If the server uses a different function, such as `force-type`, to determine the `type`, then the MIME types table is not used for that particular request.

## How the Type Affects the Response

The server considers the value of the `type` attribute when deciding which `Service` directive in `obj.conf` to use to generate the response to the client.

By default, if the `type` does not start with `magnus-internal/`, the server just sends the requested file to the client. The directive in `obj.conf` that contains this instruction is:

```
Service method=(GET|HEAD|POST) type=~magnus-internal/* fn=send-file
```

By convention, all values of `type` that require the server to do something other than just send the requested resource to the client start with `magnus-internal/`.

For example, if the requested resource's file extension is `.map`, the type is mapped to `magnus-internal/imagemap`. If the extension is `.cgi`, `.exe`, or `.bat`, the type is set to `magnus-internal/cgi`:

```
type=magnus-internal/imagemap      exts=map
type=magnus-internal/cgi           exts=cgi,exe,bat
```

If the type starts with `magnus-internal/`, the server executes whichever `Service` directive in `obj.conf` matches the specified type. For example, if the type is `magnus-internal/imagemap`, the server uses the `imagemap` function to generate the response to the client, as indicated by the following directive:

```
Service method=(GET|HEAD) type=magnus-internal/imagemap fn=imagemap
```

## What Does the Client Do with the MIME Type?

The `Service` function generates the data and sends it to the client that made the request. The server sends headers to the client along with the data. These headers include whichever MIME type attributes are known (which is usually `type`).

When the client receives the data, it uses the MIME type to decide what to do with the data. For browser clients, the data is usually displayed in the browser window.

If the requested resource cannot be displayed in a browser but needs to be handled by another application, its `type` starts with `application/`, for example `application/octet-stream` (for `.bin` file extensions) or `application/x-maker` (for `.fm` file extensions). The client has its own set of user-editable mappings that tells it which application to use to handle which types of data.

For example, if the type is `application/x-maker`, the client usually handles it by opening Adobe FrameMaker to display the file.

## Syntax of the MIME Types File

The first line in the MIME types file identifies the file format and must read:

```
#--Sun Microsystems MIME Information
```

Other non-comment lines have the following format:

```
type=type/subtype exts=[file extensions]
```

- `type/subtype` is the type and subtype.
- `exts` are the file extensions associated with this type.

# Sample MIME Types File

Here is an example of a MIME types file:

```

#--Sun Microsystems MIME Information
# Do not delete the above line. It is used to identify the file type.
type=application/octet-stream      exts=bin,exe
type=application/oda                exts=oda
type=application/pdf                exts=pdf
type=application/postscript         exts=ai,eps,ps
type=application/rtf                exts=rtf
type=application/x-mif              exts=mif,fm
type=application/x-gtar             exts=gtar
type=application/x-shar             exts=shar
type=application/x-tar              exts=tar
type=application/mac-binhex40       exts=hqx

type=audio/basic                    exts=au,snd
type=audio/x-aiff                   exts=aif,aiff,aifc
type=audio/x-wav                    exts=wav

type=image/gif                      exts=gif
type=image/ief                      exts=ief
type=image/jpeg                     exts=jpeg,jpg,jpe
type=image/tiff                     exts=tiff,tif
type=image/x-rgb                    exts=rgb
type=image/x-xbitmap                exts=xbm
type=image/x-xpixmap                exts=xpm
type=image/x-xwindowdump            exts=xwd

type=text/html                      exts=htm,html
type=text/plain                     exts=txt
type=text/richtext                   exts=rtx
type=text/tab-separated-values       exts=tsv
type=text/x-setext                  exts=etx

type=video/mpeg                     exts=mpeg,mpg,mpe
type=video/quicktime                exts=qt,mov
type=video/x-msvideo                exts=avi

enc=x-gzip                          exts=gz
enc=x-compress                      exts=z
enc=x-uencode                       exts=uu,uue

type=magnus-internal/imagemap       exts=map
type=magnus-internal/parsed-html    exts=shtml
type=magnus-internal/cgi             exts=cgi,exe,bat
type=magnus-internal/jsp            exts=jsp

```

# Other Configuration Files

This chapter summarizes the **Purpose**, **Location**, and **Contents** or **Syntax** of important configuration files not discussed in other chapters, then briefly describes all directives or parameters allowed in each file (if any) in a table. Cross references are listed after **See Also** headings when other chapters or manuals describe some of the directives or parameters in more detail. Configuration files that should never be modified are not listed in this chapter.

The following configuration files are described in alphabetical order:

- [dbswitch.conf](#)
- [Deployment Descriptors](#)
- [generated.instance.acl](#)
- [loadbalancer.xml](#)
- [nsfc.conf](#)
- [password.conf](#)
- [server.policy](#)

## dbswitch.conf

### **Purpose**

Specifies the LDAP directory that Sun Java System Application Server uses.

---

### **NOTE**

The `dbswitch.conf` interface is Unstable. An unstable interface may be experimental or transitional, and hence may change incompatibly, be removed, or be replaced by a more stable interface in the next release.

---

**Location**

*instance\_dir*/config

**Syntax**

directory *name* *LDAP\_URL*  
*name:property1* [*value1*]  
*name:property2* [*value2*]  
...

The default contents of this file are as follows:

```
directory default null:///none
```

Edit the file as follows for anonymous binding over SSL:

```
directory default ldaps://directory.sun.com:636:/dc%3Dcom
```

Edit the file as follows for anonymous binding *not* over SSL:

```
directory default ldap://directory.sun.com:389:/dc%3Dcom
```

**See Also**

[“User Database Selection” on page 90](#)

The following table describes properties in the `dbswitch.conf` file.

**Table 5-1** dbswitch.conf Properties

Property	Allowed Values	Default Value	Description
<code>nsessions</code>	A positive integer	8	The number of LDAP connections for the database.
<code>dyngroups</code>	<code>off</code> , <code>on</code> , <code>recursive</code>	<code>on</code>	Determines how dynamic groups are handled. If <code>off</code> , dynamic groups are not supported. If <code>on</code> , dynamic groups are supported. If <code>recursive</code> , dynamic groups can contain other groups.
<code>binddn</code>	A valid DN		The DN used for connecting to the database. If both <code>binddn</code> and <code>bindpw</code> are not present, binding is anonymous.
<code>bindpw</code>			The password used for connecting to the database. If both <code>binddn</code> and <code>bindpw</code> are not present, binding is anonymous.

**Table 5-1** `dbswitch.conf` Properties (*Continued*)

Property	Allowed Values	Default Value	Description
<code>dcsuffix</code>	A valid DN (relative to the LDAP URL)	<code>none</code>	<p>If present, the default value of the base DN for the request's virtual server is determined by a DC tree lookup, starting at the <code>dcsuffix</code> DN, of the virtual server's <code>hosts</code> attribute that matches the client-supplied <code>Host</code> header. If no <code>hosts</code> attribute matches, the <code>server-name</code> attribute of the parent <code>http-listener</code> is used.</p> <p>If not present, the default value of the base DN is the <code>base DN</code> value in the LDAP URL.</p> <p>The <code>basedn</code> attribute of an <code>auth-db</code> element in the <code>server.xml</code> file overrides this value.</p>
<code>digestauth</code>	<code>off</code> , <code>on</code>	<code>off</code>	Specifies whether the database can do digest authentication. If <code>on</code> , a special Directory Server plug-in is required. For information about how to install this plug-in, see the <i>Sun Java System Application Server Administrator's Guide</i> .

If an LDAP database is schema compliant, the base DN of the access is computed using a DC tree lookup of the `virtual-server` element's `hosts` attribute that matches the client-supplied `Host` header. If no `hosts` attribute matches, the `server-name` attribute of the parent `http-listener` is used. The result must contain an `inetDomainBaseDN` attribute that contains the base DN. This base DN is taken as is and is not relative to any of the base DN values.

## Deployment Descriptors

### Purpose

Configures features specific to the Sun Java System Application Server for deployed modules and applications.

### Location

The `META-INF` or `WEB-INF` directory of a module or application.

**See Also**

The following table shows where to find more information about Sun Java System Application Server deployment descriptors.

**Table 5-2** Sun Java System Application Server Descriptors

Deployment Descriptor	Where to Find More Information
sun-application.xml	<i>Sun Java System Application Server Developer's Guide</i>
sun-web.xml	<i>Sun Java System Application Server Developer's Guide to Web Applications</i>
sun-ejb-jar.xml and sun-cmp-mapping.xml	<i>Sun Java System Application Server Developer's Guide to Enterprise JavaBeans Technology</i>
sun-application-client.xml and sun-acc.xml	<i>Sun Java System Application Server Developer's Guide to Clients</i>
sun-ra.xml	<i>Sun Java System J2EE CA Service Provider Implementation Administrator's Guide</i>

# generated.instance.acl

**Purpose**

Sets permissions for access to the server instance. This is the default ACL file; you can create and use others.

<b>NOTE</b>	The ACL file interface is Unstable. An unstable interface may be experimental or transitional, and hence may change incompatibly, be removed, or be replaced by a more stable interface in the next release.
-------------	--

**Location**

*instance\_dir/config*

**See Also**

*Sun Java System Application Server Administrator's Guide to Security*



# loadbalancer.xml

## Purpose

Configures clusters and load balancing.

## NOTE

The `loadbalancer.xml` interface is Unstable. An unstable interface may be experimental or transitional, and hence may change incompatibly, be removed, or be replaced by a more stable interface in the next release.

## Location

The `loadbalancer.xml` file is located in the front-end web server.

## Syntax

Most of the file has the following basic XML syntax, with nested elements:

```
<ELEMENT attribute="value" attribute="value" ... >
  <SUBELEMENT attribute="value" attribute="value" ... />
</ELEMENT>
```

In the “[loadbalancer.xml Elements and Attributes](#)” table, elements are in bold to distinguish them from attributes.

## See Also

*Sun Java System Application Server Administrator's Guide*

Sun Java System Web Server documentation set

The following tables describe elements, attributes, and properties in the `loadbalancer.xml` file.

**Table 5-3** `loadbalancer.xml` Elements and Attributes

Element/Attribute	Subelements or Default Values	Description
<b>loadbalancer</b>	<b>cluster</b> , <b>property</b>	Defines a load balancer.
<b>cluster</b>	<b>instance</b> , <b>web-module</b> , <b>health-checker</b>	Defines a cluster of application server instances.
name	none	Specifies the name of the cluster. Within a load balancer, cluster names must be unique.
<b>instance</b>	none	Defines an application server instance.

**Table 5-3** loadbalancer.xml Elements and Attributes (*Continued*)

Element/Attribute	Subelements or Default Values	Description
name	none	Specifies the name of the server instance. Within a cluster, instance names must be unique.
enabled	true	(optional) Specifies whether instance is active (enabled) for requests to be load balanced to it.
disable-timeout-in-minutes	31	(optional) Specifies the quiescing timeout interval, at which the load balancer disables the instance and closes all its open connections.
listeners	none	Specifies a space-delimited list of the URLs of the instance's listeners. Within a cluster, listener URLs must be unique.
<b>web-module</b>	none	Defines a web module.
context-root	none	Specifies the context root of the web module. Within a cluster, context roots must be unique.
enabled	true	(optional) Specifies whether web module is active (enabled) for requests to be load balanced to it.
disable-timeout-in-minutes	31	(optional) Specifies the quiescing timeout interval, at which the load balancer disables the web module and closes all its open connections.
<b>health-checker</b>	none	Configures the cluster's health checker.
url	/	(optional) Specifies the URL to ping to determine the health state of a listener.
interval-in-seconds	30	(optional) Specifies the interval at which health checks of instances occur.
timeout-in-seconds	10	(optional) Specifies the timeout interval within which a response must be obtained for a listener to be considered healthy.
<b>property</b>	<b>description</b>	Defines a property.
name	none	Specifies the name of the property.
value	none	Specifies the value of the property.
<b>description</b>		Defines a text description of a property.

**Table 5-4** loadbalancer.xml Properties

Property Name	Default Value	Description
response-timeout-in-seconds	60	Specifies the timeout interval within which a response must be obtained for a request that is load balanced, or the instance is considered unhealthy.
reload-poll-interval-in-seconds	0	Specifies the interval at which the load balancer checks whether the loadbalancer.xml time stamp has changed. If it has changed, the load balancer reloads the file. If 0, polling is disabled.
https-routing	false	Specifies whether the load balancer routes incoming HTTPS requests to the instance as HTTPS requests.
require-monitor-daemon	false	Specifies whether load balancer monitoring is enabled.

## nsfc.conf

### Purpose

Sets file cache parameters. This file is present only if file cache parameters have been changed from their defaults.

### NOTE

The nsfc.conf interface is Unstable. An unstable interface may be experimental or transitional, and hence may change incompatibly, be removed, or be replaced by a more stable interface in the next release.

### Location

*instance\_dir/config*

### Syntax

*parameter=value*

### See Also

*Sun Java System Application Server Performance Tuning Guide*

The following table describes parameters in the nsfc.conf file.

**Table 5-5** nsfc.conf Parameters

Parameter	Allowed Values	Default Value	Description
FileCacheEnable	on, off	on	Enables the file cache.
CacheFileContent	on, off	on	Enables caching of file contents as well as file information for files smaller than MediumFileSizeLimit (smaller than SmallFileSizeLimit if TransmitFiles is on).
MaxAge	Number of seconds	30	The maximum age of a valid cache entry. This setting controls how long cached information is used once a file has been cached. An entry older than MaxAge is replaced by a new entry for the same file.
MediumFileSizeLimit	Limited by available memory	537600 (525K)	(UNIX only) Maximum size of a file that can be cached as a memory-mapped file (if TransmitFiles is off).
MediumFileSpace	Limited by available memory	10485760 (10 M)	Total size of all files that are cached as memory-mapped files (if TransmitFiles is off).
SmallFileSizeLimit	Limited by available memory	2048 (2K)	(UNIX only) Maximum size of a file that can be read into memory.
SmallFileSpace	Limited by available memory	1048576 (UNIX, 1 M)	Total size of all files that are read into memory.
TransmitFiles	on, off	off (UNIX)	Enables use of the TransmitFile system call. Not supported on IRIX, Compaq, or Linux.
MaxFiles		1024	Maximum number of files in the file cache.
HashInitSize	Limited by available memory	0	Initial number of hash buckets. If 0, the number of hash buckets is dynamically determined as 2 * MaxFiles + 1.
TempDir	A path	system_temp/ instance	Specifies a temporary directory for the file cache if CopyFiles is on.

# password.conf

## Purpose

By default, the application server prompts the administrator for the SSL key database password before starting up. If you want the application server to be able to restart unattended, you need to save the password in a `password.conf` file. Be sure that your system is adequately protected so that this file and the key databases are not compromised.

## Location

*instance\_dir/config*

This file is not present by default. You must create it if you need it.

## Syntax

*PKCS#11\_module\_name:password*

If you are using the internal PKCS#11 software encryption module that comes with the server, type the following:

*internal:password*

If you are using a different PKCS#11 module, for example for hardware encryption or hardware accelerators, you will need to specify the name of the PKCS#11 module, followed by the password, for example:

*internal:password*

## See Also

*Sun Java System Application Server Administrator's Guide*

# server.policy

## Purpose

Controls what access applications have to resources. This is the standard J2SE policy file.

## Location

*instance\_dir/config*

## Syntax

```
grant [codeBase "path"] {
    permission permission_class "package", "permission_type";
    ...
};
```

server.policy

**See Also**

*Sun Java System Application Server Developer's Guide*

<http://java.sun.com/docs/books/tutorial/security1.2/tour2/index.html>

# Time Formats

This appendix describes the format strings used for dates and times in the server log. You use these formats in the `ErrorLogDateFormat` directive in `init.conf`.

The following table describes the format strings for dates and times.

**Table A-1** Time Formats

Symbol	Meaning
%a	Abbreviated weekday name (3 chars)
%d	Day of month as decimal number (01-31)
%S	Second as decimal number (00-59)
%M	Minute as decimal number (00-59)
%H	Hour in 24-hour format (00-23)
%Y	Year with century, as decimal number, up to 2099
%b	Abbreviated month name (3 chars)
%h	Abbreviated month name (3 chars)
%T	Time "HH:MM:SS"
%X	Time "HH:MM:SS"
%A	Full weekday name
%B	Full month name
%C	"%a %b %e %H:%M:%S %Y"
%c	Date & time "%m/%d/%y %H:%M:%S"
%D	Date "%m/%d/%y"
%e	Day of month as decimal number (1-31) without leading zeros
%l	Hour in 12-hour format (01-12)

1.

**Table A-1** Time Formats (*Continued*)

Symbol	Meaning
%j	Day of year as decimal number (001-366)
%k	Hour in 24-hour format (0-23) without leading zeros
%l	Hour in 12-hour format (1-12) without leading zeros
%m	Month as decimal number (01-12)
%n	line feed
%p	A.M./P.M. indicator for 12-hour clock
%R	Time "%H: %M"
%r	Time "%I: %M: %S %p"
%t	tab
%U	Week of year as decimal number, with Sunday as first day of week (00-51)
%w	Weekday as decimal number (0-6; Sunday is 0)
%W	Week of year as decimal number, with Monday as first day of week (00-51)
%x	Date "%m/%d/%Y"
%y	Year without century, as decimal number (00-99)
%%	Percent sign



# Alphabetical List of Server Configuration Elements

## A

[acl](#) 35  
[admin-service](#) 30  
[applications](#) 84  
[auth-db](#) 42  
[auth-realm](#) 66  
[availability-service](#) 52

## C

[connector-module](#) 89  
[custom-resource](#) 74

## D

[description](#) 29

## E

[ejb-container](#) 46

## H

[ejb-module 87](#)  
[external-jndi-resource 75](#)

## H

[http-listener 32](#)  
[http-qos 41](#)  
[http-service 30](#)

## I

[iiop-cluster 53](#)  
[iiop-endpoint 54](#)  
[iiop-listener 44](#)  
[iiop-server-instance 54](#)  
[iiop-service 42](#)

## J

[j2ee-application 86](#)  
[java-config 69](#)  
[jdbc-connection-pool 80](#)  
[jdbc-resource 76](#)  
[jms-resource 78](#)  
[jms-service 61](#)  
[jvm-options 72](#)

**L**

[lifecycle-module](#) 85

[log-service](#) 63

**M**

[mail-resource](#) 76

[manager-properties](#) 57

[mdb-container](#) 50

[mime](#) 35

**O**

[orb](#) 43

**P**

[persistence-manager-factory-resource](#) 80

[persistence-store](#) 55

[profiler](#) 71

[property](#) 28

**R**

[resources](#) 73

**S**

[security-service](#) 64

## T

- [server](#) 27
- [server-instance](#) 30
- [session-config](#) 55
- [session-manager](#) 56
- [session-properties](#) 60
- [ssl](#) 33
- [ssl-client-config](#) 44
- [store-properties](#) 58

## T

- [transaction-service](#) 68

## V

- [virtual-server](#) 37
- [virtual-server-class](#) 36

## W

- [web-container](#) 46
- [web-module](#) 88

# Alphabetical List of Directives in init.conf

## A

[ACLCacheLifetime](#) 115  
[ACLGroupCacheSize](#) 115  
[ACLUserCacheSize](#) 115  
[AsyncDNS](#) 105

## C

[CGIExpirationTimeout](#) 112  
[CGIStubIdleTimeout](#) 112  
[ChildRestartCallback](#) 119  
[ChunkedRequestBufferSize](#) 118  
[ChunkedRequestTimeout](#) 118  
[ConnQueueSize](#) 107

## D

[DNS](#) 105

## E

## E

[ErrorLogDateFormat](#) 113

## H

[HeaderBufferSize](#) 107

[HTTPVersion](#) 119

## I

[IOTimeout](#) 107

## K

[KeepAliveThreads](#) 107

[KeepAliveTimeout](#) 108

## L

[ListenQ](#) 108

[LogFlushInterval](#) 114

## M

[MaxCGIStubs](#) 113

[MaxKeepAliveConnections](#) 108

[MaxRqHeaders](#) 119

[MinCGIStubs](#) 113

**N**

[NativePoolMaxThreads 111](#)

[NativePoolMinThreads 111](#)

[NativePoolQueueSize 111](#)

[NativePoolStackSize 111](#)

[NetSiteRoot 103](#)

**P**

[PidLog 114](#)

[PostThreadsEarly 108](#)

**R**

[RcvBufSize 109](#)

[ReentrantTimeFunctions \(Solaris only\) 120](#)

[RqThrottle 109](#)

[RqThrottleMin 109](#)

**S**

[Security 116](#)

[SndBufSize 109](#)

[SSL3SessionTimeout 117](#)

[SSLCacheEntries 116](#)

[SSLClientAuthDataLimit 116](#)

[SSLClientAuthTimeout 117](#)

[SSLSessionTimeout 117](#)

[StackSize 109](#)

## T

[StrictHttpHeaders 110](#)

## T

[TempDir 103](#)

[TempDirSecurity 103](#)

[TerminateTimeout 110](#)

[ThreadIncrement 110](#)

## U

[Umask \(UNIX only\) 120](#)

[UseOutputStreamSize 118](#)

[User \(UNIX only\) 104](#)



# Index

## A

- accept-language attribute [37, 40](#)
- acceptor-threads attribute [32](#)
- access log file [63](#)
- accesslog variable [94](#)
- acl element [35](#)
- ACL file [35, 90](#)
  - default [128](#)
  - init.conf directives for [114](#)
- ACLCacheLifetime init.conf directive [115](#)
- ACLGroupCacheSize init.conf directive [115](#)
- acls attribute [40](#)
- ACLUserCacheSize init.conf directive [115](#)
- address attribute [32, 45](#)
- admin-password attribute [62](#)
- admin-service element [30](#)
- admin-user-name attribute [62](#)
- alphabetical reference
  - init.conf directives [141](#)
  - server.xml elements [137](#)
- anonymous-role attribute [65](#)
- append-version property [63](#)
- application-root attribute [28](#)
- applications
  - configuration of [86](#)
  - location [28](#)
- applications element [84](#)
- assign-groups property [67](#)
- AsyncDNS init.conf directive [105](#)
- attributes, about [26](#)

- audit-enabled attribute [65](#)
- auth-db element [42](#)
- auth-realm element [66](#)
- automatic-recovery attribute [68](#)
- availability-enabled attribute [46, 49, 52, 87, 89](#)
- availability-service element [52](#)

## B

- bandwidth-limit attribute [41](#)
- basedn attribute [42](#)
- base-dn property [67](#)
- binddn property [126](#)
- bindpw property [126](#)
- blocking-enabled attribute [33](#)
- bytecode-preprocessors attribute [71](#)

## C

- CacheFileContent parameter [132](#)
- cache-idle-timeout-in-seconds attribute [48](#)
- cache-resize-quantity attribute [48](#)
- certificates [67](#)
  - settings in init.conf [115](#)
- certmaps attribute [42](#)
- cert-nickname attribute [34](#)
- CGI

- settings in init.conf 112
- variables for 94
- CGIExpirationTimeout init.conf directive 112
- CGIStubIdleTimeout init.conf directive 112
- ChildRestartCallback init.conf directive 119
- chroot variable 94
- chunked encoding 117
- ChunkedRequestBufferSize init.conf directive 118
- ChunkedRequestTimeout init.conf directive 118
- class-name attribute 85
- classname attribute 66
- classpath attribute 72, 86
- classpath-prefix attribute 70
- classpath-suffix attribute 71
- client-auth-enabled attribute 34
- cluster element 129
- cluster, changing the configuration of 21
- CMP 80
- cmt-max-runtime-exceptions property 51
- commit-option attribute 49
- config directory, location 17
- config-file attribute 37, 39
- configuration files 17
  - manually editing 20
- configuration, dynamic 19
- connection-limit attribute 41
- connections, settings in init.conf 105
- connection-validation-method attribute 82
- connector-module element 89
- ConnQueueSize init.conf directive 107
- container-managed persistence 80
- context-root attribute 88, 130
- convergence tree
  - auxiliary class inetSubscriber 92
  - in LDAP schema 91
  - organization of 91
  - user entries are called inetOrgPerson 92
- custom-resource element 74

## D

- database attribute 42
- database selection 42, 90
- databaseName property 83
- datasource-classname attribute 81
- datasourceName property 83
- dbswitch.conf file 90, 91, 125
- dcsuffix property 127
- debug-enabled attribute 70
- debug-options attribute 70
- default virtual server
  - for an http-listener element 33
  - for the entire server 37
- default-object attribute 37, 39
- default-principal attribute 65
- default-principal-password attribute 65
- default-realm attribute 65
- default-virtual-server attribute 33
- default-web-module attribute 39
- deployment descriptors 127
- description element 29, 130
- description property 84
- digestauth property 127
- dir variable 94
- directory property 59, 67
- disable-distributed-transaction-logging property 69
- disable-timeout-in-minutes attribute 130
- DNS init.conf directive 105
- docroot variable 94
- domain component tree (dc) 92
- Domain Name Service *see* DNS
- dynamic reconfiguration 19
- dyngroups property 126

## E

- echo-log-messages-to-stderr attribute 64
- ejb-container element 46
- ejb-module element 87
- enableCookies property 60

- enabled attribute [130](#)
- enableURLRewriting property [61](#)
- enc parameter [122](#)
- encoding, chunked [117](#)
- enforce-bandwidth-limit attribute [41](#)
- enforce-connection-limit attribute [41](#)
- env-classpath-ignored attribute [71](#)
- ErrorLogDateFormat init.conf directive [113](#)
- external-jndi-resource element [75](#)

## F

- factory-class attribute [74, 75, 80](#)
- fail-all-connections attribute [83](#)
- family attribute [32](#)
- file attribute [64](#)
  - for acl element [36](#)
  - for mime element [35](#)
- file name extensions, MIME types [121](#)
- file property [67](#)
- FileCacheEnable parameter [132](#)
- files, mapping types of [121](#)
- formats, time [135](#)

## G

- group variable [94](#)
- group-base-dn property [67](#)
- group-search-filter property [67](#)
- group-target property [67](#)

## H

- HashInitSize parameter [132](#)
- HeaderBufferSize init.conf directive [107](#)
- health-checker element [130](#)
- heuristic-decision attribute [68](#)

- host attribute [54](#)
- hosts attribute [40](#)
  - checking against subject pattern [38](#)
- HTTP listen socket [32](#)
- HTTP requests, processing [18](#)
- http-listener element [32](#)
- http-listeners attribute [39](#)
- http-qos element [41](#)
- http-service element [30](#)
- https-routing property [131](#)
- HTTPVersion init.conf directive [119](#)
- HUP signal, PidLog and [114](#)

## I

- id attribute
  - for acl element [36](#)
  - for auth-db element [42](#)
  - for http-listener element [32, 45](#)
  - for iiop-endpoint element [54](#)
  - for mime element [35](#)
  - for virtual-server element [39](#)
  - for virtual-server-class element [37](#)
- id variable [93](#)
- idLengthBytes property [61](#)
- idle-thread-timeout-in-seconds attribute [44](#)
- idle-timeout-in-seconds attribute [50, 82](#)
- IIOP listen socket [44](#)
- iiop-cluster element [53](#)
- iiop-endpoint element [54](#)
- iiop-listener element [44](#)
  - definition in sun-server\_1\_1.dtd file [25](#)
- iiop-server-instance element [54](#)
- iiop-service element [42](#)
- imqBrokerHostName property [79](#)
- imqBrokerHostPort property [79](#)
- imqConfiguredClientID property [79](#)
- imqDestinationName property [79](#)
- inetOrgPerson in convergence tree [92](#)
- Init functions [102](#)
- init.conf file [18, 101](#)

- alphabetical list of directives [141](#)
- initialization settings [18, 101](#)
- init-timeout-in-seconds attribute [62](#)
- instance element [129](#)
- instance-name property [62](#)
- instance-name-suffix property [62](#)
- interval-in-seconds attribute [130](#)
- IOTimeout init.conf directive [107](#)
- is-connection-validation-required attribute [82](#)
- is-failure-fatal attribute [86](#)
- is-isolation-level-guaranteed attribute [82](#)

## J

- j2ee-application element [86](#)
- jaas-context property [67](#)
- Java Database Connectivity *see* JDBC
- Java Message Service *see* JMS
- Java Naming and Directory Interface *see* JNDI
- Java Platform Debugger Architecture *see* JPDA
- Java Transaction Service *see* JTS
- Java Virtual Machine *see* JVM
- java-config element [69](#)
- javac-options attribute [70](#)
- java-home attribute [70](#)
- JavaMail [76](#)
- JDBC connection pool [80](#)
- JDBC resource [76](#)
- jdbc-connection-pool element [80](#)
- jdbc-resource element [76](#)
- jdbc-resource-jndi-name attribute [80](#)
- JMS [61, 78](#)
- jms-resource element [78](#)
- jms-service element [61](#)
- JNDI [75](#)
- jndi-lookup-name attribute [75](#)
- jndi-name attribute
  - for custom-resource element [74](#)
  - for external-jndi-resource element [75](#)
  - for jdbc-resource element [76](#)

- for jms-resource element [78](#)
- for mail-resource element [77](#)
- for persistence-manager-factory-resource element [80](#)
- JPDA debugging options [70](#)
- JTS [68](#)
- JVM [69](#)
  - adding options to the server [72](#)
- jvm-options element [72](#)

## K

- KeepAliveThreads init.conf directive [107](#)
- KeepAliveTimeout init.conf directive [108](#)
- keypoint-interval attribute [68](#)

## L

- lang parameter [122](#)
- LDAP [90](#)
  - specifying which certificate mapping to use [42](#)
  - Sun Java System schema [91](#)
- level attribute [64](#)
- lifecycle-module element [85](#)
- listen socket [32, 44](#)
- listeners attribute [130](#)
- ListenQ init.conf directive [108](#)
- load balancing [88](#)
- loadbalancer element [129](#)
- loadbalancer.xml file [129](#)
- load-order attribute [86](#)
- locale attribute [28](#)
- location attribute [86, 87, 88, 89](#)
- log-file attribute [40](#)
- LogFlushInterval init.conf directive [114](#)
- logging
  - settings in init.conf [113](#)
  - settings in server.xml [63](#)
- log-level attribute [64](#)
- log-root attribute [28](#)

log-service element [63](#)  
 log-stderr attribute [64](#)  
 log-stdout attribute [64](#)  
 log-virtual-server-id attribute [64](#)

## M

mail-resource element [76](#)  
 manager-properties element [57](#)  
 MaxAge parameter [132](#)  
 max-cache-size attribute [48](#)  
 MaxCGIStubs init.conf directive [113](#)  
 max-connections attribute [44](#)  
 MaxFiles parameter [132](#)  
 MaxKeepAliveConnections init.conf directive [108](#)  
 max-pool-size attribute [47](#), [50](#), [81](#)  
 MaxRqHeaders init.conf directive [119](#)  
 maxSessions property [57](#)  
 max-thread-pool-size attribute [43](#)  
 max-wait-time-in-millis attribute [81](#)  
 mdb-container element [50](#)  
 MediumFileSizeLimit parameter [132](#)  
 MediumFileSpace parameter [132](#)  
 message-driven beans [50](#)  
 message-fragment-size attribute [43](#)  
 mime attribute [40](#)  
 mime element [35](#)  
 MIME types [121](#)  
 mime.types file  
   default [122](#)  
   overview of [19](#)  
   sample of [124](#)  
   server.xml element referencing [35](#)  
   syntax [123](#)  
 MinCGIStubs init.conf directive [113](#)  
 monitoring-enabled attribute [44](#), [46](#), [49](#), [50](#), [68](#)

## N

name attribute [129](#), [130](#)  
   for auth-realm element [66](#)  
   for connector-module element [89](#)  
   for ejb-module element [87](#)  
   for iiop-server-instance element [54](#)  
   for j2ee-application element [86](#)  
   for jdbc-connection-pool element [81](#)  
   for lifecycle-module element [85](#)  
   for profiler element [72](#)  
   for property element [29](#)  
   for server element [28](#)  
   for web-module element [88](#)  
 native library path, configuring [71](#)  
 native thread pools, settings in init.conf [110](#)  
 native-library-path attribute [72](#)  
 native-library-path-prefix attribute [71](#)  
 native-library-path-suffix attribute [71](#)  
 NativePoolMaxThreads init.conf directive [111](#)  
 NativePoolMinThreads init.conf directive [111](#)  
 NativePoolQueueSize init.conf directive [111](#)  
 NativePoolStackSize init.conf directive [111](#)  
 networkProtocol property [83](#)  
 nice variable [94](#)  
 nsessions property [126](#)  
 nsfc.conf file [131](#)

## O

obj.conf file [18](#), [39](#)  
   and virtual servers [18](#)  
 oracle-xa-recovery-workaround property [69](#)  
 orb element [43](#)

## P

password property [83](#)  
 password.conf file [133](#)  
 persistenceFrequency property [58](#)

- persistence-manager-factory-resource element 80
- persistenceScope property 59
- persistence-store element 55
- persistence-type attribute 56
- PidLog init.conf directive 114
- pool-idle-timeout-in-seconds attribute 48
- pool-name attribute 76
- pool-resize-quantity attribute 47, 50, 82
- port attribute 32, 45, 54, 62
- port property 83
- PostThreadsEarly init.conf directive 108
- processes, settings in init.conf 105
- profiler element 71
- properties, about 28
- property element 28, 130

## Q

- quality of service 41

## R

- RcvBufSize init.conf directive 109
- reapIntervalSeconds property 57
- reconnect-delay-in-seconds property 51
- reconnect-enabled property 51
- reconnect-max-retries property 51
- ReentrantTimeFunctions init.conf directive 120
- reload-poll-interval-in-seconds property 131
- removal-timeout-in-seconds attribute 49
- request-handling process 18
- request-response process 18
- requirement rules 25
- require-monitor-data property 131
- resource adapters 89
- resources element 73
- response-timeout-in-seconds property 131
- res-type attribute 74, 75, 78, 81

- rmic-options attribute 70
- roleName property 83
- rpm 16
- RqThrottle init.conf directive 109
- RqThrottleMin init.conf directive 109

## S

- search-bind-dn property 67
- search-bind-password property 67
- search-filter property 67
- Secure Socket Layer *see* SSL
- security 64
  - of listen sockets 32, 44
  - settings in init.conf 115
- Security init.conf directive 116
- security-enabled attribute 33
- security-service element 64
- send-cgi function, variables for 94
- server
  - basic operation 17
  - HUP signal 114
  - init.conf directives that configure 103
  - initialization directives in init.conf 101
  - killing process of 114
  - modifying 17
  - TERM signal 114
- server element 27
- server log file 63
  - location 28, 64
- server.policy file 133
- server.xml file 18
  - alphabetical element list 137
  - elements in 26
  - example of 95
  - location 23
- server-classpath attribute 70
- server-instance element 30
- server-name attribute 33
- serverName property 83
- session managers 56
- session-config element 55

- sessionFilename property 58
- session-manager element 56
- session-properties element 60
- session-store attribute 28
- session-timeout element 60
- showrev 16
- single sign-on 40
- SmallFileSizeLimit parameter 132
- SmallFileSpace parameter 132
- SndBufSize init.conf directive 109
- SSL
  - configuring 33
  - settings in init.conf 115
- ssl element 33
- ssl2-ciphers attribute 34
- ssl2-enabled attribute 34
- SSL3 client authentication 34
- ssl3-enabled attribute 34
- SSL3SessionTimeout init.conf directive 117
- ssl3-tls-ciphers attribute 34
- SSLCacheEntries init.conf directive 116
- SSLClientAuthDataLimit init.conf directive 116
- SSLClientAuthTimeout init.conf directive 117
- ssl-client-config element 44
- SSLSessionTimeout init.conf directive 117
- sso-enabled property 40
- sso-max-inactive-seconds property 41
- sso-reap-interval-seconds property 41
- StackSize init.conf directive 109
- start-args attribute 62
- state attribute 40
- stderr output 64
- stdout output 64
- steady-pool-size attribute 47, 50, 81
- steady-thread-pool-size attribute 43
- store-pool-jndi-name property 55
- store-properties element 58
- store-protocol attribute 77
- store-protocol-class attribute 77
- StrictHttpHeaders init.conf directive 110
- subelements, about 25
- Sun customer support 15

- Sun Java System Application Server *see* server
- Sun Java System Message Queue 62
- sun-acc.xml file 128
- sun-application.xml file 128
- sun-application-client.xml file 128
- sun-cmp-mapping.xml file 128
- sun-ejb-jar.xml file 128
- sun-ra.xml file 128
- sun-server\_1\_1.dtd file 24
- sun-web.xml file 128
- sybase-xa-recovery-workaround property 69

## T

- TempDir init.conf directive 103
- TempDir parameter 132
- TempDirSecurity init.conf directive 103
- TERM signal 114
- TerminateTimeout init.conf directive 110
- thread pools, settings in init.conf 110
- ThreadIncrement init.conf directive 110
- threads, settings in init.conf 105
- time formats 135
- timeout-in-seconds attribute 68, 130
- timeoutSeconds property 60
- tls-enabled attribute 34
- tls-rollback-enabled attribute 34
- transaction log file 63, 68
- transaction-isolation-level attribute 82
- transaction-service element 68
- TransmitFiles parameter 132
- transport-protocol attribute 77
- transport-protocol-class attribute 77
- tx-log-dir attribute 68
- type parameter 122
- type-by-extension function 122

## U

- Umask init.conf directive [120](#)
- UNIX user account, specifying [104](#)
- url attribute [130](#)
- url property [84](#)
- UseOutputStreamSize init.conf directive [118](#)
- user database selection [42](#), [90](#)
- User init.conf directive [104](#)
- user property [83](#)
- user variable [94](#)
- use-system-logging attribute [64](#)

## V

- validation-table-name attribute [83](#)
- value attribute [29](#), [130](#)
- variables [92](#)
  - evaluation [94](#)
  - format of [93](#)
  - referencing in obj.conf [92](#)
  - substitution, where allowed [93](#)
  - used in the interfaces [93](#)
- victim-selection-policy attribute [49](#)
- virtual servers
  - and the obj.conf file [18](#)
  - log file for [63](#)
- virtual-server element [37](#)
- virtual-server-class element [36](#)
- virtual-servers attribute [86](#), [88](#)

## W

- web module, default [39](#)
- web.xml file
  - session-timeout element [60](#)
- web-container element [46](#)
- web-module element [88](#), [130](#)

## X

- xaresource-txn-timeout property [69](#)