

Administrator's Configuration File Reference

SunTM ONE Application Server

Version 7

816-7155-10
September 2002

Copyright © 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

THIS SOFTWARE 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 and the Sun ONE logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

UNIX is a registered trademark in the U.S. and 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.

Copyright © 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

CE LOGICIEL 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, ÉCRITE ET PRÉALABLE DE SUN MICROSYSTEMS, INC. Droits du gouvernement américain, utilisateurs gouvernementaux - logiciel commercial. Les utilisateurs gouvernementaux sont soumis au contrat de licence standard de Sun Microsystems, Inc., ainsi qu'aux dispositions en vigueur de la FAR [(Federal Acquisition Regulations) et des suppléments à celles-ci. Distribué par des licences qui en restreignent l'utilisation.

Cette distribution peut comprendre des composants développés par des tiers.

Sun, Sun Microsystems, le logo Sun, Java et le logo Sun ONE sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

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

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

Contents

About This Book	9
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
Product Support	15
Chapter 1 Basics of Server Operation	17
Configuration Files	17
server.xml	18
init.conf	18
obj.conf	18
mime.types	19
Dynamic Reconfiguration	19
Manually Editing Configuration Files	20
Chapter 2 Server Configuration Files	23
The sun-server_1_0.dtd File	24
Subelements	24
Data	25
Attributes	25
Elements in the server.xml File	26
General Elements	26
server	27
property	28
description	29
admin-service	30
server-instance	30

Listener Service Elements	30
http-service	30
http-listener	32
ssl	34
mime	35
acl	36
virtual-server-class	37
virtual-server	38
http-qos	43
auth-db	43
iiop-service	44
orb	45
ssl-client-config	46
iiop-listener	47
Container Elements	48
web-container	48
ejb-container	49
mdb-container	53
J2EE Service Elements	55
jms-service	55
log-service	57
security-service	59
auth-realm	60
transaction-service	62
Java Configuration Elements	64
java-config	65
profiler	67
jvm-options	68
Resource Elements	68
resources	69
custom-resource	70
external-jndi-resource	71
jdbc-resource	72
mail-resource	72
jms-resource	74
persistence-manager-factory-resource	76
jdbc-connection-pool	77
Application Elements	81
applications	81
lifecycle-module	82
j2ee-application	83
ejb-module	84
web-module	85

connector-module	86
User Database Selection	87
The Sun ONE LDAP Schema	88
The Convergence Tree	89
The Domain Component (dc)Tree	90
Variables	90
Format of a Variable	91
The id Variable	91
Other Important Variables	91
Variable Evaluation	92
Sample server.xml File	93
 Chapter 3 Syntax and Use of init.conf	97
Init Functions	98
Server Information	99
NetSiteRoot	99
TempDir	99
TempDirSecurity	99
User (UNIX only)	100
DNS Lookup	101
AsyncDNS	101
DNS	101
Threads, Processes and Connections	102
ConnQueueSize	103
HeaderBufferSize	103
IOTimeout	103
KeepAliveThreads	104
KeepAliveTimeout	104
KernelThreads (Windows only)	104
ListenQ	105
MaxKeepAliveConnections	105
PostThreadsEarly	105
RcvBufSize	105
RqThrottle	105
RqThrottleMin	106
SndBufSize	106
StackSize	106
StrictHttpHeaders	106
TerminateTimeout	107
ThreadIncrement	107
Native Thread Pools	107
NativePoolStackSize	108
NativePoolMaxThreads	108

NativePoolMinThreads	108
NativePoolQueueSize	108
CGI	109
CGIExpirationTimeout	109
CGIStubIdleTimeout	109
MaxCGIStubs	110
MinCGIStubs	110
WincgiTimeout (Windows only)	110
Error Logging	110
ErrorLogDateFormat	110
LogFlushInterval	111
PidLog	111
ACL	112
ACLCacheLifetime	112
ACLUserCacheSize	112
ACLGroupCacheSize	112
Security	113
Security	113
SSLCacheEntries	114
SSLClientAuthDataLimit	114
SSLClientAuthTimeout	114
SSLSessionTimeout	114
SSL3SessionTimeout	114
Chunked Encoding	115
UseOutputStreamSize	115
ChunkedRequestBufferSize	115
ChunkedRequestTimeout	116
Miscellaneous	116
ChildRestartCallback	116
HTTPVersion	117
MaxRqHeaders	117
ReentrantTimeFunctions (Solaris only)	117
Umask (UNIX only)	117
Chapter 4 MIME Types	119
Introduction	119
Determining the MIME Type	120
How the Type Affects the Response	120
What Does the Client Do with the MIME Type?	121
Syntax of the MIME Types File	121
Sample MIME Types File	122

Chapter 5 Other Configuration Files	123
dbswitch.conf	123
Deployment Descriptors	125
generated. <i>instance</i> .acl	126
nsfc.conf	127
password.conf	128
server.policy	129
 Appendix A Time Formats	 131
 Appendix B Alphabetical List of Server Configuration Elements	 133
 Appendix C Alphabetical List of Directives in init.conf	 137
 Index	 141

About This Book

The book discusses the purpose and use of the configuration files for Sun™ Open Net Environment (Sun ONE) Application Server 7, 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
- Product Support

Who Should Use This Guide

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

This guide assumes you are familiar with the following topics:

- J2EE specification
- HTTP
- HTML
- XML
- Java programming
- Java APIs as defined in servlet, JSP, EJB, and JDBC specifications

- Relational database concepts

Using the Documentation

The Sun ONE Application Server manuals are available as online files in Portable Document Format (PDF) and Hypertext Markup Language (HTML) formats, at:

<http://docs.sun.com/>

The following table lists tasks and concepts described in the Sun ONE Application Server manuals. The left column lists the tasks and concepts, and the right column lists the corresponding manuals.

Sun ONE Application Server Documentation Roadmap

For information about	See the following
Late-breaking information about the software and the documentation	<i>Release Notes</i>
Supported platforms and environments	<i>Platform Summary</i>
Introduction to the application server, including new features, evaluation installation information, and architectural overview.	<i>Getting Started Guide</i>
Installing Sun ONE Application Server and its various components (sample applications, Administration interface, Sun ONE Message Queue).	<i>Installation Guide</i>
Creating and implementing J2EE applications that follow the open Java standards model on the Sun ONE Application Server 7. Includes general information about application design, developer tools, security, assembly, deployment, debugging, and creating lifecycle modules.	<i>Developer's Guide</i>
Creating and implementing J2EE applications that follow the open Java standards model for web applications on the Sun ONE Application Server 7. Discusses web application programming concepts and tasks, and provides sample code, implementation tips, and reference material.	<i>Developer's Guide to Web Applications</i>
Creating and implementing J2EE applications that follow the open Java standards model for enterprise beans on the Sun ONE Application Server 7. Discusses EJB programming concepts and tasks, and provides sample code, implementation tips, and reference material.	<i>Developer's Guide to Enterprise JavaBeans Technology</i>

Sun ONE Application Server Documentation Roadmap (Continued)

For information about	See the following
Creating clients that access J2EE applications on the Sun ONE Application Server 7	<i>Developer's Guide to Clients</i>
Creating web services	<i>Developer's Guide to Web Services</i>
J2EE features such as JDBC, JNDI, JTS, JMS, JavaMail, resources, and connectors	<i>Developer's Guide to J2EE Features and Services</i>
Creating custom NSAPI plugins	<i>Developer's Guide to NSAPI</i>
Performing the following administration tasks:	<i>Administrator's Guide</i>
• Using the Administration interface and the command line interface • Configuring server preferences • Using administrative domains • Using server instances • Monitoring and logging server activity • Configuring the web server plugin • Configuring the Java Messaging Service • Using J2EE features • Configuring support for CORBA-based clients • Configuring database connectivity • Configuring transaction management • Configuring the web container • Deploying applications • Managing virtual servers	
Editing server configuration files	<i>Administrator's Configuration File Reference</i>
Configuring and administering security for the Sun ONE Application Server 7 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>

Sun ONE Application Server Documentation Roadmap (Continued)

For information about	See the following
Configuring and administering service provider implementation for J2EE CA connectors for the Sun ONE Application Server 7. Includes information about the Administration Tool, DTDs and provides sample XML files.	<i>J2EE CA Service Provider Implementation Administrator's Guide</i>
Migrating your applications to the new Sun ONE Application Server 7 programming model from the Netscape Application Server version 2.1, including a sample migration of an Online Bank application provided with Sun ONE Application Server	<i>Migration Guide</i>
Using Sun ONE Message Queue.	The Sun ONE Message Queue documentation at: http://docs.sun.com/?p=/coll/S1_MessageQueue_30

How This Guide Is Organized

This book has the following chapters and appendices:

- Chapter 1, “Basics of Server Operation”
This chapter introduces the major configuration files that control the Sun ONE 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 ONE Application Server during initialization.
- Chapter 4, “MIME Types”
This appendix 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 appendices 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 8 non-package-based Evaluation installations:
user's home directory/sun/appserver7
- Solaris unbundled, non-evaluation installations:
/opt/SUNWappserver7
- Windows, all installations:
C:\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 8 and 9 package-based installation and the Solaris 9 bundled installation, the application server files are spread across several root directories. These directories are described in this section.

- **For Solaris 9 bundled installations**, this guide uses the following document conventions to correspond to the various default installation directories provided:
 - *install_dir* refers to */usr/appserver/*, 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/appserver/domains*, which is the default location for any domains that are created.

- *install_config_dir* refers to `/etc/appserver/config`, which contains installation-wide configuration information such as licenses and the master list of administrative domains configured for this installation.
- **For Solaris 8 and 9 package-based, non-evaluation, unbundled installations,** 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.

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
- Detailed steps on the methods you have used to reproduce the problem
- Any error logs or core dumps

Basics of Server Operation

The configuration and behavior of Sun ONE Application Server is determined by a set of configuration files. When you use the Administration interface, you change the settings in these configuration files. You can also manually edit these files.

This chapter has the following sections:

- Configuration Files
- Dynamic Reconfiguration
- Manually Editing Configuration Files

Configuration Files

The configuration and operation of the Sun ONE 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. These files are:

- `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, see Chapter 5, “Other Configuration Files.”

server.xml

This file contains most of the server configuration. A schema file, `sun-server_1_0.dtd`, defines its format and content.

For more information about how the server uses `sun-server_1_0.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.

See Chapter 3, “Syntax and Use of `init.conf`” for a list of all the variables and `Init` directives that can be set in `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 plugins and CGI programs. 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 ONE 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`. Editing one of these files directly or through the Administration interface changes the configuration of a virtual server.

The `obj.conf` file is essential to the operation of the Sun ONE Application Server. When you make changes to the server through the Administration interface, the system automatically updates `obj.conf`.

The file `obj.conf` contains a series of instructions (directives) that tell the Sun ONE Application Server what to do at each stage in the request-response process.

For information about how the server uses `obj.conf`, see the *Sun ONE 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

You do not have to restart the server for changes to these files to take effect:

- `obj.conf`
- `mime.types`
- ACL files
- `server.xml` (applications and resources subelements only)

However, with the exception of deployment, you must apply the changes for those changes to take effect. 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 that 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

Manually editing configuration files is not recommended. Here are some cautionary guidelines for manually editing configuration files:

- Edit only the files in the *instance_dir/config* directory. Do not copy files to the backup directory or remove the timestamp files (which prevent overwriting of manual changes).
- If you have made both manual changes and changes through the Administration interface 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.

- Use the `--keepmanualchanges` or `--discardmanualchanges` option of the `asadmin reconfig` command to make the same choice at the command line. For example:

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

To avoid this dilemma, always apply changes immediately after making them, and don't mix manual and Administration interface changes.

- Do not manually edit a file and then restart the server. Use the `--keepmanualchanges` option of the `asadmin reconfig` command before restarting the server. See the example above.
- Whenever the configuration in `server.xml` is modified using the Administration interface, it is always validated against the `sun-server_1_0.dtd` file. Additional validation is done that pertains to checking the file's cross-references. Make sure your configuration is complete and correct before applying it.
- When you modify applications or resources subelements of `server.xml` (for example, you deploy a J2EE application or register a JDBC Connection Pool), the Administration Server attempts to apply the changes to a running server instance. Make sure your configuration is complete and correct before applying it.
- 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.

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_0.dtd`, determines the format and content of the `server.xml` file.

This chapter describes `server.xml` and `sun-server_1_0.dtd` in these sections:

- The `sun-server_1_0.dtd` File
- Elements in the `server.xml` File
- General Elements
- Listener Service Elements
- Container Elements
- J2EE Service Elements
- Java Configuration Elements
- Resource Elements
- Application Elements
- User Database Selection
- The Sun ONE 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_0.dtd File

The sun-server_1_0.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_0.dtd file is located in the *install_dir/lib/dtds* directory.

NOTE Do not edit the sun-server_1_0.dtd file; its contents change only with new versions of Sun ONE Application Server.

NOTE The sun-server_1_0.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. The left column lists the subelement ending character, and the right column lists the corresponding requirement rule.

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 iio-listener      id          CDATA      #REQUIRED
                             address     CDATA      #REQUIRED
                             port        CDATA      "3700"
                             enabled     %boolean;  "true">
```

An `iio-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 ONE 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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

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 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 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 Java Transaction Service.
<code>java-config</code>	only one	Contains the JVM configuration.
<code>resources</code>	only one	Contains configured resources.
<code>applications</code>	only one	Contains deployed J2EE applications, J2EE modules, and Lifecycle modules.
<code>property</code>	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `server` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

server attributes

Attribute	Default	Description
name	none	Specifies the name of the server instance.
locale	operating system default	(optional) Specifies the server instance language.
log-root	<i>instance_dir</i> /logs	(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 log-service description for details about logs.
application-root	<i>instance_dir</i> /applications	(optional) Specifies the absolute path where deployed applications reside for this server instance.
session-store	<i>instance_dir</i> /session-store	(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 90.

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

- Optional with respect to Sun ONE Application Server
- Needed by a system or object that Sun ONE 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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

property subelements

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

Attributes

The following table describes attributes for the property element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

property attributes

Attribute	Default	Description
name	none	Specifies the name of the property or variable.
value	none	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 ONE Application Server Performance Tuning, Sizing, and Scaling Guide*.

Subelements

The following table describes subelements for the `http-service` element. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

http-service subelements

Element	Required	Description
http-listener	zero or more	Defines an HTTP listen socket.
mime	zero or more	Defines MIME types.
acl	zero or more	References an ACL file.
virtual-server-class	zero or more	Defines a virtual server class.
http-qos	zero or one	Defines quality of service parameters.
property	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. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

http-service attributes

Attribute	Default	Description
qos-metrics-interval-in-seconds	30	(optional) Specifies the interval during which traffic is measured.
qos-recompute-time-interval-in-millis	100	(optional) Specifies the period in which the bandwidth gets recomputed for all server entities.
qos-enabled	true	(optional) Enables quality of service features, which let you set limits on server entities or view server statistics for bandwidth and connections. Allowed values are <code>yes</code> , <code>no</code> , <code>on</code> , <code>off</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .

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 <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 Security directive.
-------------	--

Subelements

The following table describes subelements for the `http-listener` element. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

http-listener subelements

Element	Required	Description
<code>ssl</code>	zero or one	Defines SSL parameters.

Attributes

The following table describes attributes for the `http-listener` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

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 recommended.

http-listener attributes

Attribute	Default	Description
family	inet	(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.
acceptor-threads	1	(optional) Number of acceptor threads for the listener. The recommended value is the number of processors in the machine. Legal values are 1 - 1024.
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. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .
security-enabled	false	(optional) Determines whether the listener runs SSL. Legal values are <code>on</code>, <code>off</code>, <code>yes</code>, <code>no</code>, <code>1</code>, <code>0</code>, <code>true</code>, <code>false</code>. 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 <code>on</code> or <code>security-enabled</code> in <code>server.xml</code> does not work. For more information, see Chapter 3, “Syntax and Use of <code>init.conf</code>.”
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. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .

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. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

SSL attributes

Attribute	Default	Description
cert-nickname	none	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.
ssl2-enabled	false	(optional) Determines whether SSL2 is enabled. Legal values are on, off, yes, no, 1, 0, true, false. 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.
ssl2-ciphers	none	(optional) A comma-separated list of the SSL2 ciphers used, with the prefix + to enable or - to disable, for example +rc4. Allowed values are rc4, rc4export, rc2, rc2export, idea, des, desede3.

ssl attributes

Attribute	Default	Description
ssl3-enabled	true	(optional) Determines whether SSL3 is enabled. Legal values are on, off, yes, no, 1, 0, true, false. The default is true. 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.
ssl3-tls-ciphers	none	(optional) A comma-separated list of the SSL3 ciphers used, with the prefix + to enable or - to disable, for example +rsa_des_sha. Allowed SSL3 values are rsa_rc4_128_md5, rsa_3des_sha, rsa_des_sha, rsa_rc4_40_md5, rsa_rc2_40_md5, rsa_null_md5. Allowed TLS values are rsa_des_56_sha, rsa_rc4_56_sha.
tls-enabled	true	(optional) Determines whether TLS is enabled. Legal values are on, off, yes, no, 1, 0, true, false.
tls-rollback-enabled	true	(optional) Determines whether TLS rollback is enabled. Legal values are on, off, yes, no, 1, 0, true, false. TLS rollback should be enabled for Microsoft Internet Explorer 5.0 and 5.5. For more information, see the <i>Sun ONE Application Server Administrator's Guide</i> .
client-auth-enabled	false	(optional) Determines whether SSL3 client authentication is performed on every request, independent of ACL-based access control. Legal values are on, off, yes, no, 1, 0, true, false.

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.

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.

Subelements
none

Attributes
The following table describes attributes for the `mime` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

`mime` attributes

Attribute	Default	Description
<code>id</code>	<code>none</code>	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.
<code>file</code>	<code>none</code>	The name of a MIME types file. For information about the format of this file, see Chapter 4, “MIME Types.”

acl
References an ACL file.

NOTE 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. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

acl attributes

Attribute	Default	Description
id	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.
file	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 ONE 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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

virtual-server-class subelements

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

Attributes

The following table describes attributes for the `virtual-server-class` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

virtual-server-class attributes

Attribute	Default	Description
id	none	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.

virtual-server-class attributes

Attribute	Default	Description
config-file	none	(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.
default-object	default	(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. 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. 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.
accept-language	false	(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. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> . Can be overridden in a <code>virtual-server</code> element.
enabled	true	(optional) Determines whether the virtual server class is active. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .

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 ONE 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 ONE 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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

`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. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

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	(optional) The default <code>web-module</code> for this virtual server, which responds to all requests that cannot be resolved to other <code>web</code> modules deployed to this virtual server. 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 <code>web</code> module is used.
config-file	<i>virtual_server_name</i> -obj.conf	(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.

virtual-server attributes

Attribute	Default	Description
default-object	default	(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. 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>. 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.
hosts	none	A comma-separated list of values allowed in the <code>Host</code> request header to select the current virtual server. Each <code>virtual-server</code> that is configured to the same <code>http-listener</code> must have a unique <code>hosts</code> value for that listener.
mime	none	The id of the <code>mime</code> element used by the virtual server.
state	on	(optional) Determines whether a <code>virtual-server</code> is active (<code>on</code>) or inactive (<code>off</code>, <code>disabled</code>). The default is <code>on</code> (active). When inactive, a <code>virtual-server</code> does not service requests. If a <code>virtual-server</code> is disabled, only the global server administrator can turn it on.
acls	none	(optional) One or more id attributes of <code>acl</code> elements, separated by commas. Specifies the ACL file(s) used by the virtual server.

virtual-server attributes

Attribute	Default	Description
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. Legal values are on, off, yes, no, 1, 0, true, false. 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 server 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 log-service description for details about logs.

Properties

The following table describes properties for the virtual-server element. The left column lists the property name, the middle column indicates the default value if the property is omitted, and the right column describes what the property does.

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. Legal values are on, off, yes, no, 1, 0, true, false.
sso-max-inactive-seconds	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-seconds	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 ONE Application Server Performance Tuning, Sizing, and Scaling Guide*.

Subelements

none

Attributes

The following table describes attributes for the `http-qos` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

http-qos attributes

Attribute	Default	Description
<code>bandwidth-limit</code>	none	(required if <code>enforce-bandwidth-limit</code> is <code>true</code>) The maximum bandwidth limit for the server, <code>virtual-server-class</code> , or <code>virtual-server</code> in bytes per second.
<code>enforce-bandwidth-limit</code>	false	(optional) Specifies whether the bandwidth limit should be enforced or not. Allowed values are <code>yes</code> , <code>no</code> , <code>on</code> , <code>off</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .
<code>connection-limit</code>	none	(required if <code>enforce-connection-limit</code> is <code>true</code>) The maximum number of concurrent connections for the server, <code>virtual-server-class</code> , or <code>virtual-server</code> .
<code>enforce-connection-limit</code>	false	(optional) Specifies whether the connection limit should be enforced or not. Allowed values are <code>yes</code> , <code>no</code> , <code>on</code> , <code>off</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .

auth-db

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

See “User Database Selection,” on page 87 for more information about how a user database is selected for a given virtual server.

NOTE	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 ONE Application Server Administrator's Guide to Security</i> .
-------------	---

Subelements

none

Attributes

The following table describes attributes for the `auth-db` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

`auth-db` attributes

Attribute	Default	Description
<code>id</code>	none	The user database name in the virtual server's ACL file. A user database name cannot begin with a number.
<code>database</code>	none	The user database name in the <code>dbswitch.conf</code> file.
<code>basedn</code>	none	(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>	none	(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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

iiop-service subelements

Element	Required	Description
orb	only one	Configures the ORB.
ssl-client-config	zero or one	Defines SSL parameters for the ORB.
iiop-listener	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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

orb subelements

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

Attributes

The following table describes attributes for the `orb` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

orb attributes

Attribute	Default	Description
message-fragment-size	1024	(optional) GIOPv1.2 messages larger than this number of bytes are fragmented.

orb attributes

Attribute	Default	Description
steady-thread-pool-size	10	(optional) The minimum number of worker threads in the ORB.
max-thread-pool-size	200	(optional) The maximum number of worker threads in the ORB.
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 log-service 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 log-service element.
monitoring-enabled	false	(optional) Determines whether monitoring of the ORB is enabled. Legal values are on, off, yes, no, 1, 0, true, false.

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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

ssl-client-config subelements

Element	Required	Description
ssl	only one	Defines SSL parameters.

Attributes

none

iiop-listener

Defines an IIOP listen socket.

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

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 `iiop-listener` element. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

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. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

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>	<code>3700</code> (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.

iiop-listener attributes

Attribute	Default	Description
enabled	true	(optional) Determines whether the listener is active. Legal values are on, off, yes, no, 1, 0, true, false.

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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

web-container subelements

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

Attributes

The following table describes attributes for the web-container element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

web-container attributes

Attribute	Default	Description
monitoring-enabled	false	(optional) Determines whether monitoring of the web container is enabled. Legal values are on, off, yes, no, 1, 0, true, false.

web-container attributes

Attribute	Default	Description
log-level	Value of level attribute of log-service 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 log-service element. ServletContext.log messages are logged at the INFO level by default.

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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

ejb-container subelements

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

Attributes

The following table describes attributes for the `ejb-container` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

ejb-container attributes

Attribute	Default	Description
steady-pool-size	32	(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>. Bean instances are removed from the pool and returned after use. The pool is replenished or cleaned up periodically to maintain this size. Applies to stateless session beans and entity beans.
pool-resize-quantity	16	(optional) Specifies the number of beans to be: - created if a request arrives when the pool has no available beans (subject to the <code>max-pool-size</code> limit) - removed when the <code>pool-idle-timeout-in-seconds</code> timer expires and a cleaner thread removes any unused instances 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>. Applies to stateless session beans and entity beans.
max-pool-size	64	(optional) Specifies the maximum number of beans that can be created to satisfy client requests. A value of 0 indicates an unbounded pool. Applies to stateless session beans and entity beans.

ejb-container attributes

Attribute	Default	Description
cache-resize-quantity	16	(optional) Specifies the number of beans to be: - created if a request arrives when the pool has no available beans (subject to the max-cache-size limit) - passivated when the cache-idle-timeout-in-seconds timer expires and a cleaner thread removes any unused instances, or when the cache size exceeds max-cache-size. Must be greater than 1 and less than max-cache-size. Applies to stateful session beans and entity beans.
max-cache-size	512	(optional) Specifies the maximum number of beans in the cache. A value of 0 indicates an unbounded cache. Applies to stateful session beans and entity beans.
pool-idle-timeout-in-seconds	600	(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. Applies to stateless session beans and entity beans.
cache-idle-timeout-in-seconds	600	(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. Applies to stateful session beans and entity beans.

ejb-container attributes

Attribute	Default	Description
removal-timeout-in-seconds	5400	(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. If removal-timeout-in-seconds is less than or equal to cache-idle-timeout-in-seconds, beans are removed immediately without being passivated. The session-store attribute of the server element determines the location of the session store. Applies to stateful session beans and entity beans.
victim-selection-policy	nru	(optional) Specifies how entity and stateful session beans are selected for passivation. Allowed values are fifo, lru, and nru: • fifo selects the oldest instance. • lru selects the least recently accessed instance. • nru selects a not recently used instance.
commit-option	B	(optional) Determines which commit option is used for entity beans. Legal values are B or C.
log-level	Value of level attribute of log-service 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 log-service element.
monitoring-enabled	false	(optional) Determines whether monitoring of the EJB container is enabled. Legal values are on, off, yes, no, 1, 0, true, false.

mdb-container

Configures the message-driven bean (MDB) container.

Subelements

The following table describes subelements for the `mdb-container` element. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

mdb-container subelements

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

Attributes

The following table describes attributes for the `mdb-container` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

mdb-container attributes

Attribute	Default	Description
steady-pool-size	10	(optional) Specifies the initial and minimum number of beans maintained in the pool.
pool-resize-quantity	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> .
max-pool-size	60	(optional) Specifies the maximum number of beans that can be created to satisfy client requests.
idle-timeout-in-seconds	600	(optional) Specifies the maximum time that a bean can remain idle in the pool. After this amount of time, the bean is destroyed.

mdb-container attributes

Attribute	Default	Description
log-level	Value of level attribute of log-service 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 log-service element.
monitoring-enabled	false	(optional) Determines whether monitoring of the message-driven bean (MDB) container is enabled. Legal values are on, off, yes, no, 1, 0, true, false.

Properties

The following table describes properties for the mdb-container element. The left column lists the property name, the middle column indicates the default value if the property is omitted, and the right column describes what the property does.

mdb-container properties

Property	Default	Description
cmt-max-runtime-exceptions	1	Specifies the maximum number of RuntimeException occurrences allowed from a message-driven bean's <code>onMessage()</code> method when container-managed transactions are used. Deprecated.
reconnect-enabled	true	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.
reconnect-delay-in-seconds	60	Specifies the delay between reconnect attempts.
reconnect-max-retries	60	Specifies the maximum number of reconnect attempts.

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 ONE Application Server.

Subelements

The following table describes subelements for the `jms-service` element. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

`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. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

`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.

jms-service attributes

Attribute	Default	Description
init-timeout-in-seconds	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.
start-args	none	(optional) Specifies the string of arguments supplied for startup of the corresponding JMS instance.
log-level	Value of level attribute of log-service 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 log-service element.
enabled	true	(optional) If set to true, the Sun ONE Application Server instance is responsible for starting up and shutting down the JMS provider. If set to false, the Sun ONE 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 ONE Application Server). Legal values are on, off, yes, no, 1, 0, true, false.

Properties

The following table describes properties for the jms-service element. The left column lists the property name, the middle column indicates the default value if the property is omitted, and the right column describes what the property does.

jms-service properties

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

jms-service properties

Property	Default	Description
append-version	false	If true, appends the major and minor version numbers, preceded by underscore characters (_), to the full Sun ONE 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 ONE 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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

log-service subelements

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

Attributes

The following table describes attributes for the `log-service` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

`log-service` attributes

Attribute	Default	Description
<code>file</code>	<code>server.log</code> in the directory specified by the <code>log-root</code> attribute of the <code>server</code> 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 <code>log-root</code> attribute of the <code>server</code> element. If you specify a relative path, it is relative to the <code>log-root</code> attribute of the <code>server</code> element. If no <code>log-root</code> value is specified, it is relative to <code>instance_dir/config</code> .
<code>level</code>	<code>INFO</code>	(optional) Controls the default type of messages logged by other elements to the server log. Many other elements can override this default in their <code>log-level</code> attributes. Allowed values are, from highest to lowest: <code>FINEST</code> , <code>FINER</code> , <code>FINE</code> , <code>CONFIG</code> , <code>INFO</code> , <code>WARNING</code> , <code>SEVERE</code> , <code>ALERT</code> , <code>FATAL</code> . Each value logs all messages for all lower values; for example, <code>FINEST</code> logs all messages, and <code>FATAL</code> logs only <code>FATAL</code> messages. The default value is <code>SEVERE</code> , which logs all <code>SEVERE</code> , <code>ALERT</code> , and <code>FATAL</code> messages.
<code>log-stdout</code>	<code>true</code>	(optional) If <code>true</code> , redirects <code>stdout</code> output to the server log. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .
<code>log-stderr</code>	<code>true</code>	(optional) If <code>true</code> , redirects <code>stderr</code> output to the server log. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .
<code>echo-log-messages-to-stderr</code>	<code>true</code>	(optional) If <code>true</code> , sends log messages to <code>stderr</code> in addition to the server log. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .
<code>create-console</code>	<code>false</code>	(optional, Windows only) If <code>true</code> , creates a Windows console for <code>stderr</code> output. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .
<code>log-virtual-server-id</code>	<code>false</code>	(optional) If <code>true</code> , virtual server IDs are displayed in the virtual server logs. This is useful if multiple <code>virtual-server</code> elements share the same log file. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .

log-service attributes

Attribute	Default	Description
use-system-logging	false	(optional) If true, uses the UNIX syslog service or Windows Event Logging to produce and manage logs. Legal values are on, off, yes, no, 1, 0, true, false.

security-service

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

Subelements

The following table describes subelements for the security-service element. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

security-service subelements

Element	Required	Description
auth-realm	one or more	Defines a realm for authentication.
property	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the security-service element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

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.

security-service attributes

Attribute	Default	Description
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. Legal values are on, off, yes, no, 1, 0, true, false. Audit information consists of: • Authentication success and failure events • Servlet and EJB access grants and denials
log-level	Value of level attribute of log-service 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 log-service 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 ONE Application Server Developer's Guide*.

Here is an example of the default file realm:

```
<auth-realm name="file"
  classname="com.ipplanet.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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

auth-realm subelements

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

Attributes

The following table describes attributes for the auth-realm element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

auth-realm attributes

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

Properties

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

The following table describes properties for the auth-realm element. The left column lists the property name, the middle column indicates the standard realms that use the property, and the right column describes what the property does.

auth-realm properties

Property	Realms	Description
jaas-context	file, ldap, solaris	Specifies the JAAS (Java Authentication and Authorization Service) context.

auth-realm properties

Property	Realms	Description
file	file	Specifies the file that stores user names. The default is <i>instance_dir/config/keyfile</i> .
assign-groups	certificate	(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.
directory	ldap	Specifies the LDAP URL to your server.
base-dn	ldap	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.
search-filter	ldap	(optional) Specifies the search filter to use to find the user. The default is <code>uid=%s</code> (%s expands to the subject name).
group-base-dn	ldap	(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.
group-search-filter	ldap	(optional) Specifies the search filter to find group memberships for the user. The default is <code>uniquemember=%d</code> (%d expands to the user element DN).
group-target	ldap	(optional) Specifies the LDAP attribute name that contains group name entries. The default is CN.
search-bind-dn	ldap	(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.
search-bind-password	ldap	(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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

transaction-service subelements

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

Attributes

The following table describes attributes for the transaction-service element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

transaction-service attributes

Attribute	Default	Description
automatic-recovery	false	(optional) If true, the server instance attempts transaction recovery during startup. Legal values are on, off, yes, no, 1, 0, true, false.
timeout-in-seconds	0	(optional) Specifies the amount of time after which the transaction is aborted. If set to 0, the transaction never times out.
tx-log-dir	directory specified by the log-root attribute of the server 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 log-service description for details about logs.
heuristic-decision	rollback	(optional) During recovery, if the outcome of a transaction cannot be determined from the logs, this property determines the outcome. Allowed values are rollback and commit.
keypoint-interval	2048	(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, 4096) results in a larger transaction log file, but fewer keypoint operations and potentially better performance. A smaller value (for example, 100) results in smaller log files, but slightly reduced performance due to the greater frequency of keypoint operations.
log-level	Value of level attribute of log-service 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 log-service element.

transaction-service attributes

Attribute	Default	Description
monitoring-enabled	false	(optional) Determines whether monitoring of the JTS is enabled. Legal values are on, off, yes, no, 1, 0, true, false. The default is false.

Properties

The following table describes properties for the transaction-service element. The left column lists the property name, the middle column indicates the default value if the property is omitted, and the right column describes what the property does.

transaction-service properties

Property	Default	Description
oracle-xa-recovery-workaround	false	If true, the Oracle XA Resource workaround is used in transaction recovery. Legal values are on, off, yes, no, 1, 0, true, false.
sybase-xa-recovery-workaround	false	If true, the Sybase XA Resource workaround is used in transaction recovery. Legal values are on, off, yes, no, 1, 0, true, false.
disable-distributed-transaction-logging	false	If true, disables transaction logging, which may improve performance. Legal values are on, off, yes, no, 1, 0, true, false. If the automatic-recovery attribute is set to true, this property is ignored.
xaresource-txn-timeout	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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

java-config subelements

Element	Required	Description
profiler	zero or one	Configures a profiler for use with Sun ONE Application Server.
jvm-options	zero or more	Contains JVM command line options.
property	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `java-config` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

java-config attributes

Attribute	Default	Description
java-home	none	The path to the directory where the JDK is installed.
debug-enabled	false	(optional) If <code>true</code> , the server starts up in debug mode ready for attachment with a JPDA-based debugger. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .
debug-options	-Xdebug -Xrunjdw: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: http://java.sun.com/products/jpda/doc/conninv.html#Invocation For more information about debugging, see the <i>Sun ONE Application Server Developer's Guide</i> .

java-config attributes

Attribute	Default	Description
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 ONE 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> .
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.

java-config attributes

Attribute	Default	Description
env-classpath-ignored	true	(optional) If <code>false</code>, the <code>CLASSPATH</code> environment variable is read and appended to the Sun ONE 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. Legal values are <code>on</code>, <code>off</code>, <code>yes</code>, <code>no</code>, <code>1</code>, <code>0</code>, <code>true</code>, <code>false</code>.

profiler

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

Subelements

The following table describes subelements for the `profiler` element. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

profiler subelements

Element	Required	Description
jvm-options	zero or more	Contains profiler-specific JVM command line options.
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. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

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. Legal values are on, off, yes, no, 1, 0, true, false.

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 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 ONE Application Server application should begin with the string java:comp/env.

Subelements

The following table describes subelements for the resources element. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

resources subelements		
Element	Required	Description
custom-resource	zero or more	Defines a custom resource.
external-jndi-resource	zero or more	Defines a resource that resides in an external JNDI repository.
jdbc-resource	zero or more	Defines a JDBC (Java Database Connectivity) resource.
mail-resource	zero or more	Defines a JavaMail resource.
jms-resource	zero or more	Defines a JMS resource.
persistence-manager-factory-resource	zero or more	Defines a persistence manager factory resource for CMP.
jdbc-connection-pool	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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

custom-resource subelements

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

Attributes

The following table describes attributes for the `custom-resource` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

custom-resource attributes

Attribute	Default	Description
jndi-name	none	Specifies the JNDI 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 user-written factory class, which implements <code>javax.naming.spi.ObjectFactory</code> .
enabled	true	(optional) Determines whether this resource is enabled at runtime. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .

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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

external-jndi-resource subelements

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

Attributes

The following table describes attributes for the `external-jndi-resource` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

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 ONE Application Server Developer's Guide to J2EE Features and Services</i> .
enabled	true	(optional) Determines whether this resource is enabled at runtime. Legal values are on, off, yes, no, 1, 0, true, false.

jdbc-resource

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

Subelements

The following table describes subelements for the `jdbc-resource` element. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

`jdbc-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 `jdbc-resource` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

`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 <code>jdbc-connection-pool</code> element.
<code>enabled</code>	<code>true</code>	(optional) Determines whether this resource is enabled at runtime. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .

mail-resource

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

Subelements

The following table describes subelements for the `mail-resource` element. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

mail-resource subelements

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

Attributes

The following table describes attributes for the mail-resource element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

mail-resource attributes

Attribute	Default	Description
jndi-name	none	Specifies the JNDI name for the resource.
store-protocol	imap	(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> .
store-protocol-class	<code>com.sun.mail.imap.IMAPStore</code>	(optional) Specifies the service provider implementation class for storage.
transport-protocol	smtp	(optional) Specifies the transport protocol service, which sends messages.
transport-protocol-class	<code>com.sun.mail.smtp.SMTPTransport</code>	(optional) Specifies the service provider implementation class for transport.
host	none	The mail server host name.
user	none	The mail server user name.
from	none	The e-mail address the mail server uses to indicate the message sender.
debug	false	(optional) Determines whether debugging for this resource is enabled. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .
enabled	true	(optional) Determines whether this resource is enabled at runtime. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .

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 ONE Application Server the 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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

jms-resource subelements		
Element	Required	Description
description	zero or one	Contains a text description of this element.
property	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the jms-resource element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

jms-resource attributes

Attribute	Default	Description
jndi-name	none	Specifies the JNDI name for the resource.
res-type	none	Specifies the fully qualified type of the resource, which is one of the following: javax.jms.Topic javax.jms.Queue javax.jms.TopicConnectionFactory javax.jms.QueueConnectionFactory
enabled	true	(optional) Determines whether this resource is enabled at runtime. Legal values are on, off, yes, no, 1, 0, true, false.

Properties

The following table describes properties for the jms-resource element. The left column lists the property name, the middle column indicates the default value if the property is omitted, and the right column describes what the property does.

jms-resource properties

Property	Default	Description
imqDestinationName	none	Specifies the JMS physical destination name associated with this JMS resource. You must specify this property for jms-resource elements with the res-type of javax.jms.Topic or javax.jms.Queue. The <i>Sun ONE Message Queue Administrator's Guide</i> shows a default value for this property, but this does not apply in the Sun ONE Application Server environment.
imqBrokerHostName	the same host name as the Sun ONE Application Server instance (localhost)	Specifies the host name where the JMS service (Sun ONE Message Queue broker) is running. For jms-resource elements with the res-type of javax.jms.TopicConnectionFactory or javax.jms.QueueConnectionFactory.
imqBrokerHostPort	the jms-service element's port attribute	Specifies the port where the JMS service (Sun ONE Message Queue broker) is running. For jms-resource elements with the res-type of javax.jms.TopicConnectionFactory or javax.jms.QueueConnectionFactory.

jms-resource properties

Property	Default	Description
imqConfiguredClientID	none	Specifies the JMS Client Identifier to be associated with a Connection created using the createQueueConnection and createTopicConnection JMS APIs of the QueueConnectionFactory and TopicConnectionFactory classes, respectively. For <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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

persistence-manager-factory-resource subelements

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

Attributes

The following table describes attributes for the persistence-manager-factory-resource element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

persistence-manager-factory-resource attributes

Attribute	Default	Description
jndi-name	none	Specifies the JNDI name for the resource.
factory-class	com.sun.jdo.spi.persistence.support.sqlstore.impl.PersistenceManagerFactoryImpl	(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.
jdbc-resource-jndi-name	none	(optional) Specifies the jdbc-resource from which database connections are obtained. Must be the jndi-name of an existing jdbc-resource.
enabled	true	(optional) Determines whether this resource is enabled at runtime. Legal values are on, off, yes, no, 1, 0, true, false.

jdbc-connection-pool

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

TIP 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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

jdbc-connection-pool subelements

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

Attributes

The following table describes attributes for the `jdbc-connection-pool` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

`jdbc-connection-pool` attributes

Attribute	Default	Description
<code>name</code>	<code>none</code>	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>	<code>none</code>	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>	<code>8</code>	(optional) Specifies the initial and minimum number of connections maintained in the pool.
<code>max-pool-size</code>	<code>32</code>	(optional) Specifies the maximum number of connections that can be created to satisfy client requests.
<code>max-wait-time-in-millis</code>	<code>60000</code>	(optional) Specifies the amount of time, in milliseconds, that the caller is willing to wait for a connection. If <code>0</code> , the caller is blocked indefinitely until a resource is available or an error occurs.
<code>pool-resize-quantity</code>	<code>2</code>	(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> .

jdbc-connection-pool attributes

Attribute	Default	Description
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	(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>. 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.
is-isolation-level-guaranteed	true	(optional) Applicable only when <code>transaction-isolation-level</code> is explicitly set. If <code>true</code> , 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. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .
connection-validation-method	auto-commit	(optional) Legal values are as follows: <code>auto-commit</code> (default), which uses <code>Connection.setAutoCommit(Connection.getAutoCommit())</code> <code>meta-data</code>, which uses <code>Connection.getMetaData()</code> <code>table</code>, which performs a query on a table specified in the <code>validation-table-name</code> attribute

jdbc-connection-pool attributes

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. Legal values are on, off, yes, no, 1, 0, true, false.

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 ONE 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. The left column lists the property name, and the right column describes what the property does.

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.

`jdbc-connection-pool` properties

Property	Description
<code>networkProtocol</code>	Specifies the communication protocol.
<code>roleName</code>	Specifies the initial SQL role name.
<code>datasourceName</code>	Specifies an underlying <code>XADataSource</code> , or a <code>ConnectionPoolDataSource</code> if connection pooling is done.
<code>description</code>	Specifies a text description.
<code>url</code>	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. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

`applications` subelements

Element	Required	Description
<code>lifecycle-module</code>	zero or more	Specifies a deployed lifecycle module.
<code>j2ee-application</code>	zero or more	Specifies a deployed J2EE application.
<code>ejb-module</code>	zero or more	Specifies a deployed EJB module.

applications subelements

Element	Required	Description
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. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

applications attributes

Attribute	Default	Description
dynamic-reload-enabled	false	(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. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .
dynamic-reload-poll-interval-in-seconds	2	(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 ONE Application Server Developer's Guide*.

Subelements

The following table describes subelements for the lifecycle-module element. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

lifecycle-module subelements

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

lifecycle-module subelements

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

Attributes

The following table describes attributes for the lifecycle-module element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

lifecycle-module attributes

Attribute	Default	Description
name	none	The name of the lifecycle module.
class-name	none	The fully qualified name of the lifecycle module's class file, which must implement the <code>com.sun.appserv.server.LifecycleListener</code> interface.
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. Legal values are on, off, yes, no, 1, 0, true, false.
enabled	true	(optional) Determines whether the lifecycle module is enabled. Legal values are on, off, yes, no, 1, 0, true, false.

j2ee-application

Specifies a deployed J2EE application.

Subelements

The following table describes subelements for the j2ee-application element. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

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. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

j2ee-application attributes

Attribute	Default	Description
name	none	The name of the application.
location	none	The location of the application in the Sun ONE 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. Legal values are on, off, yes, no, 1, 0, true, false.

ejb-module

Specifies a deployed EJB module.

Subelements

The following table describes subelements for the ejb-module element. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

ejb-module subelements

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

Attributes

The following table describes attributes for the `ejb-module` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

`ejb-module` attributes

Attribute	Default	Description
<code>name</code>	<code>none</code>	The name of the EJB module.
<code>location</code>	<code>none</code>	The location of the EJB module in the Sun ONE Application Server file system.
<code>enabled</code>	<code>true</code>	(optional) Determines whether the EJB module is enabled. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .

web-module

Specifies a deployed web module.

Subelements

The following table describes subelements for the `web-module` element. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

`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. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

`web-module` attributes

Attribute	Default	Description
<code>name</code>	<code>none</code>	The name of the web module.

web-module attributes		
Attribute	Default	Description
context-root	empty string	The context root (context path without the / in front) at which the web module is installed (see Section 5.4 of the Servlet 2.3 specification). If this attribute is an empty string, which is the default value, this web module can be the default web module for the virtual-server elements specified in the virtual-servers attribute. 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 [virtual_server] already has a web module loaded at [/].
location	none	A fully qualified or relative path to the directory to which the contents of the .war file have been extracted. If relative, it is relative to the following directory: instance_dir/applications/j2ee-modules/
virtual-servers	all virtual servers	(optional) The virtual-server elements to which the web module is deployed.
enabled	true	(optional) Determines whether the web module is enabled. Legal values are on, off, yes, no, 1, 0, true, false.

connector-module

Specifies a deployed connector module.

Subelements

The following table describes subelements for the connector-module element. The left column lists the subelement name, the middle column indicates the requirement rule, and the right column describes what the element does.

connector-module subelements		
Element	Required	Description
description	zero or one	Contains a text description of this element.

Attributes

The following table describes attributes for the `connector-module` element. The left column lists the attribute name, the middle column indicates the default value, and the right column describes what the attribute does.

connector-module attributes

Attribute	Default	Description
<code>name</code>	<code>none</code>	The name of the connector module.
<code>location</code>	<code>none</code>	The location of the connector module in the Sun ONE Application Server file system.
<code>enabled</code>	<code>true</code>	(optional) Determines whether the connector module is enabled. Legal values are <code>on</code> , <code>off</code> , <code>yes</code> , <code>no</code> , <code>1</code> , <code>0</code> , <code>true</code> , <code>false</code> .

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 ONE LDAP Schema,” on page 88.

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 ONE 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, which is recommended (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). It is recommended that the `cn` contain *name initial sn*. It is recommended that the RDN of the user entry 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. Defining variables with default values at the `http-service` or `virtual-server-class` level and overriding them at the `virtual-server` level is recommended.

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 ONE 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. The left column lists variables, and the right column lists descriptions of those variables.

General variables

Variable	Description
<code>docroot</code>	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.
<code>accesslog</code>	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. The left column lists variables, and the right column lists descriptions of those variables.

send-cgi variables

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

For more information about the `send-cgi` function, see the *Sun ONE 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 the Windows version of Sun ONE 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 ONE Application Server 7.0//EN" "file:///C:/Sun/AppServer7/lib/dtds/sun-server_1_0.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.ipplanet.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.ja
r;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:/S
un/AppServer7/share/lib/jaxrpc-impl.jar;C:/Sun/AppServer7/share/lib/jaxrpc-api.ja
r;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/AppServer7/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-lo
gging.jar;C:/Sun/AppServer7/imq/lib/fscontext.jar;C:/Sun/AppServer7/imq/lib/provi

```

```

derutil.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
-Xrunjdwp: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/con
fig/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.ServerLog
Manager</jvm-options>

<jvm-options>-Dcom.sun.aas.configRoot=C:/Sun/AppServer7\config</jvm-options>
    <jvm-options>-Dcom.sun.aas.imqLib=C:\Sun\AppServer7\imq\lib</jvm-options>
    <jvm-options>-Dcom.sun.aas.imqBin=C:\Sun\AppServer7\imq\bin</jvm-options>

<jvm-options>-Dcom.sun.aas.webServicesLib=C:\Sun\AppServer7\share\lib</jvm-option
s>
    <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>

```

Syntax and Use of init.conf

When the Sun ONE 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 ONE 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.

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

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

NOTE	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 ONE 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`.”

NOTE	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 “ <code>nsfc.conf</code> ,” on page 127 and the <i>Sun ONE Application Server Performance Tuning, Sizing, and Scaling Guide</i> .
-------------	--

Init Functions

The Init functions load and initialize server modules and plugins, and they initialize log files. For more information about these functions, see the *Sun ONE 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)

`TEMP` (environment variable for Windows)

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. Although you can use the `nobody` user, it isn't recommended.

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 `on` as a default.

Example

`DNS on`

Threads, Processes and Connections

In Sun ONE 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`
- `KernelThreads` (Windows only)
- `ListenQ`
- `MaxKeepAliveConnections`
- `PostThreadsEarly`
- `RcvBufSize`
- `RqThrottle`

- RqThrottleMin
- SndBufSize
- StackSize
- StrictHttpHeaders
- TerminateTimeout
- ThreadIncrement

Also see the section “Native Thread Pools,” on page 107 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. It is recommended that this value 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. It is recommended that this number 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).

KernelThreads (Windows only)

Sun ONE Application Server can support both kernel-level and user-level threads whenever the operating system supports kernel-level threads. Local threads are scheduled by NSPR within the process whereas kernel threads are scheduled by the host operating system. Usually, the standard debugger and compiler are intended for use with kernel-level threads. By setting KernelThreads to 1 (on), you ensure that the server uses only kernel-level threads, not user-level threads. By setting KernelThreads to 0 (off), you ensure that the server uses only user-level threads, which may improve performance.

Default

The default is 0 (off).

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 platform-specific: 4096 (AIX), 200 (Windows), 128 (all others).

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 ONE Application Server Performance Tuning, Sizing, and Scaling 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>
```

On Windows, the default native pool is always being used and Sun ONE Application Server uses fibers (user-scheduled threads) for initial request processing. Using custom additional pools on Windows introduces no additional overhead.

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`
- `WincgiTimeout` (Windows only)

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.

Note that on Windows platforms `init-cgi` time-out does not work, so you must use `CGIExpirationTimeout`.

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

WinCGITimeout (Windows only)

WinCGI processes that take longer than this value are terminated when this timeout (in seconds) expires.

Default

60

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 ONE Application Server when you make changes to the LDAP entries. For example, if this value is set to 120 seconds, the Sun ONE 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 ONE 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 ONE 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 ONE Application Server NSAPI Developer's Guide*.

NOTE The `UseOutputStreamSize` parameter can be set to zero in the `obj.conf` file to disable output stream buffering. For the `init.conf` file, setting `UseOutputStreamSize` 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 ONE 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. This is the recommended value.

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 ONE Application Server NSAPI Developer's Guide*.

MIME Types

This appendix 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 ONE Application Server Developer's Guide to NSAPI*.

NOTE	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.
-------------	---

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 ONE 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. When the server sends the data to the client, it also sends headers. 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 usual thing is to display the data 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-uuencode                      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
- nsfc.conf
- password.conf
- server.policy

dbswitch.conf

Purpose

Specifies the LDAP directory that Sun ONE 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 87

The following table describes properties in the `dbswitch.conf` file. The left column lists the property names. The second column from the left lists allowed values. The third column from the left lists default values. The right column lists property descriptions.

dbswitch.conf

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

dbswitch.conf

Property	Allowed Values	Default Value	Description
dcsuffix	A valid DN (relative to the LDAP URL)	none	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 dcsuffix DN, of the virtual server'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. If not present, the default value of the base DN is the base DN value in the LDAP URL. The basedn attribute of an auth-db element in the server.xml file overrides this value.
digestauth	off, on	off	Specifies whether the database can do digest authentication. If on, a special Directory Server plugin is required. For information about how to install this plugin, see the <i>Sun ONE 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 ONE 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 ONE Application Server deployment descriptors. The left column lists the deployment descriptors, and the right column lists where to find more information about those descriptors.

Sun ONE Application Server Descriptors

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

NOTE

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 ONE Application Server Administrator's Guide to Security

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 ONE Application Server Performance Tuning, Sizing, and Scaling Guide

The following table describes parameters in the `nsfc.conf` file. The left column lists the parameter names. The second column from the left lists allowed values. The third column from the left lists default values. The right column lists parameter descriptions.

nsfc.conf

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 <code>MediumFileSizeLimit</code> (smaller than <code>SmallFileSizeLimit</code> if <code>TransmitFiles</code> 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 <code>MaxAge</code> is replaced by a new entry for the same file.

nsfc.conf

Parameter	Allowed Values	Default Value	Description
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), 0 (Windows)	Total size of all files that are read into memory.
TransmitFiles	on, off	on (Windows), 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.
CopyFiles	on, off	on	(Windows only) Prevents sharing violations by copying files to a temporary directory.
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 ONE 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";
    ...
};
```

See Also

Sun ONE Application Server Developer's Guide

<http://java.sun.com/docs/books/tutorial/security1.2/tour2/index.html>

server.policy

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. The left column lists time format symbols, and the right column explains the meanings of the symbols.

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"

Time formats

Symbol	Meaning
%e	Day of month as decimal number (1-31) without leading zeros
%I	Hour in 12-hour format (01-12)
%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 36
- admin-service 30
- applications 81
- auth-db 43
- auth-realm 60

C

- connector-module 86
- custom-resource 70

D

- description 29

E

- ejb-container 49
- ejb-module 84

external-jndi-resource 71

H

http-listener 32

http-qos 43

http-service 30

I

iiop-listener 47

iiop-service 44

J

j2ee-application 83

java-config 65

jdbc-connection-pool 77

jdbc-resource 72

jms-resource 74

jms-service 55

jvm-options 68

L

lifecycle-module 82

log-service 57

M

mail-resource 72
mdb-container 53
mime 35

O

orb 45

P

persistence-manager-factory-resource 76
profiler 67
property 28

R

resources 69

S

security-service 59
server 27
server-instance 30
ssl 34
ssl-client-config 46

T

transaction-service 62

V

virtual-server 38

virtual-server-class 37

W

web-container 48

web-module 85

Alphabetical List of Directives in init.conf

A

ACLCacheLifetime 112
ACLGroupCacheSize 112
ACLUserCacheSize 112
AsyncDNS 101

C

CGIExpirationTimeout 109
CGIStubIdleTimeout 109
ChildRestartCallback 116
ChunkedRequestBufferSize 115
ChunkedRequestTimeout 116
ConnQueueSize 103

D

DNS 101

E

ErrorLogDateFormat 110

H

HeaderBufferSize 103

HTTPVersion 117

I

IOTimeout 103

K

KeepAliveThreads 104

KeepAliveTimeout 104

KernelThreads (Windows only) 104

L

ListenQ 105

LogFlushInterval 111

M

MaxCGIStubs 110

MaxKeepAliveConnections 105

MaxRqHeaders 117

MinCGIStubs 110

N

NativePoolMaxThreads 108
NativePoolMinThreads 108
NativePoolQueueSize 108
NativePoolStackSize 108
NetSiteRoot 99

P

PidLog 111
PostThreadsEarly 105

R

RcvBufSize 105
ReentrantTimeFunctions (Solaris only) 117
RqThrottle 105
RqThrottleMin 106

S

Security 113
SndBufSize 106
SSL3SessionTimeout 114
SSLCacheEntries 114
SSLClientAuthDataLimit 114
SSLClientAuthTimeout 114
SSLSessionTimeout 114
StackSize 106

StrictHttpHeaders 106

T

TempDir 99

TempDirSecurity 99

TerminateTimeout 107

ThreadIncrement 107

U

Umask (UNIX only) 117

UseOutputStreamSize 115

User (UNIX only) 100

W

WincgiTimeout (Windows only) 110

Index

A

- accept-language attribute 38, 42
- acceptor-threads attribute 33
- access log file 57
- accesslog variable 91
- acl element 36
- ACL file 36, 87
 - default 126
 - init.conf directives for 112
- ACLCacheLifetime init.conf directive 112
- ACLGroupCacheSize init.conf directive 112
- acls attribute 41
- ACLUserCacheSize init.conf directive 112
- address attribute 32, 47
- admin-password attribute 55
- admin-service element 30
- admin-user-name attribute 55
- alphabetical reference
 - init.conf directives 137
 - server.xml elements 133
- anonymous-role attribute 60
- append-version property 57
- application-root attribute 28
- applications
 - configuration of 83
 - location 28
- applications element 81
- assign-groups property 62
- AsyncDNS init.conf directive 101
- attributes, about 25

- audit-enabled attribute 60
- auth-db element 43
- auth-realm element 60
- automatic-recovery attribute 63

B

- bandwidth-limit attribute 43
- basedn attribute 44
- base-dn property 62
- binddn property 124
- bindpw property 124
- blocking-enabled attribute 33
- bytecode-preprocessors attribute 66

C

- CacheFileContent parameter 127
- cache-idle-timeout-in-seconds attribute 51
- cache-resize-quantity attribute 51
- certificates 62
 - settings in init.conf 113
- certmaps attribute 44
- cert-nickname attribute 34
- CGI
 - settings in init.conf 109
 - variables for 92

- CGIExpirationTimeout init.conf directive 109
- CGIStubIdleTimeout init.conf directive 109
- ChildRestartCallback init.conf directive 116
- chroot variable 92
- chunked encoding 115
- ChunkedRequestBufferSize init.conf directive 115
- ChunkedRequestTimeout init.conf directive 116
- class-name attribute 83
- classname attribute 61
- classpath attribute 68, 83
- classpath-prefix attribute 66
- classpath-suffix attribute 66
- client-auth-enabled attribute 35
- CMP 76
- cmt-max-runtime-exceptions property 54
- commit-option attribute 52
- config directory, location 17
- config-file attribute 38, 40
- configuration files 17
- configuration, dynamic 19
- connection-limit attribute 43
- connections, settings in init.conf 102
- connection-validation-method attribute 79
- connector-module element 86
- ConnQueueSize init.conf directive 103
- console, Windows 58
- container-managed persistence 76
- context-root attribute 86
- convergence tree
 - auxiliary class inetSubscriber 89
 - in LDAP schema 88
 - organization of 89
 - user entries are called inetOrgPerson 89
- CopyFiles parameter 128
- create-console attribute 58
- custom-resource element 70

D

- database attribute 44

- database selection 43, 87
- databaseName property 80
- datasource-classname attribute 78
- datasourceName property 81
- dbswitch.conf file 87, 88, 123
- dcsuffix property 125
- debug-enabled attribute 65
- debug-options attribute 65
- default virtual server
 - for an http-listener element 33
 - for the entire server 38
- default-object attribute 38, 41
- default-principal attribute 59
- default-principal-password attribute 60
- default-realm attribute 59
- default-virtual-server attribute 33
- default-web-module attribute 40
- deployment descriptors 125
- description element 29
- description property 81
- digestauth property 125
- dir variable 92
- directory property 62
- disable-distributed-transaction-logging property 64
- DNS init.conf directive 101
- docroot variable 91
- domain component tree (dc) 90
- Domain Name Service *see* DNS
- dynamic reconfiguration 19
- dyngroups property 124

E

- echo-log-messages-to-stderr attribute 58
- ejb-container element 49
- ejb-module element 84
- enc parameter 120
- encoding, chunked 115
- enforce-bandwidth-limit attribute 43
- enforce-connection-limit attribute 43

- env-classpath-ignored attribute 67
- ErrorLogDateFormat init.conf directive 110
- external-jndi-resource element 71

F

- factory-class attribute 70, 71, 77
- fail-all-connections attribute 80
- family attribute 33
- file attribute 58
 - for acl element 37
 - for mime element 36
- file name extensions, MIME types 119
- file property 62
- FileCacheEnable parameter 127
- files, mapping types of 119
- formats, time 131

G

- group variable 92
- group-base-dn property 62
- group-search-filter property 62
- group-target property 62

H

- HashInitSize parameter 128
- HeaderBufferSize init.conf directive 103
- heuristic-decision attribute 63
- hosts attribute 41
 - checking against subject pattern 39
- HTTP listen socket 32
- HTTP requests, processing 18
- http-listener element 32
- http-listeners attribute 40
- http-qos element 43

- http-service element 30
- HTTPVersion init.conf directive 117
- HUP signal
 - PidLog and 111

I

- id attribute
 - for acl element 37
 - for auth-db element 44
 - for http-listener element 32, 47
 - for mime element 36
 - for virtual-server element 40
 - for virtual-server-class element 37
- id variable 91
- idle-thread-timeout-in-seconds attribute 46
- idle-timeout-in-seconds attribute 53, 79
- IIOP listen socket 47
- iiop-listener element 47
 - definition in sun-server_1_0.dtd file 24
- iiop-service element 44
- imqBrokerHostName property 75
- imqBrokerHostPort property 75
- imqConfiguredClientID property 76
- imqDestinationName property 75
- inetOrgPerson in convergence tree 89
- Init functions 98
- init.conf file 18, 97
 - alphabetical list of directives 137
- initialization settings 18, 97
- init-timeout-in-seconds attribute 56
- instance-name property 56
- instance-name-suffix property 56
- IOTimeout init.conf directive 103
- is-connection-validation-required attribute 79
- is-failure-fatal attribute 83
- is-isolation-level-guaranteed attribute 79

J

- j2ee-application element 83
- jaas-context property 61
- 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 65
- javac-options attribute 66
- java-home attribute 65
- JavaMail 72
- JDBC connection pool 77
- JDBC resource 72
- jdbc-connection-pool element 77
- jdbc-resource element 72
- jdbc-resource-jndi-name attribute 77
- JMS 55, 74
- jms-resource element 74
- jms-service element 55
- JNDI 71
- jndi-lookup-name attribute 71
- jndi-name attribute
 - for custom-resource element 70
 - for external-jndi-resource element 71
 - for jdbc-resource element 72
 - for jms-resource element 75
 - for mail-resource element 73
 - for persistence-manager-factory-resource element 77
- JPDA debugging options 65
- JTS 62
- JVM 65
 - adding options to the server 68
- jvm-options element 68

K

- KeepAliveThreads init.conf directive 104

- KeepAliveTimeout init.conf directive 104
- KernelThreads init.conf directive 104
- keypoint-interval attribute 63

L

- lang parameter 120
- LDAP 88
 - specifying which certificate mapping to use 44
 - Sun ONE schema 88
- level attribute 58
- lifecycle-module element 82
- listen socket 32, 47
- ListenQ init.conf directive 105
- load-order attribute 83
- locale attribute 28
- location attribute 84, 85, 86, 87
- log-file attribute 42
- LogFlushInterval init.conf directive 111
- logging
 - settings in init.conf 110
 - settings in server.xml 57
- log-level attribute 58
- log-root attribute 28
- log-service element 57
- log-stderr attribute 58
- log-stdout attribute 58
- log-virtual-server-id attribute 58

M

- mail-resource element 72
- MaxAge parameter 127
- max-cache-size attribute 51
- MaxCGIStubs init.conf directive 110
- max-connections attribute 46
- MaxFiles parameter 128
- MaxKeepAliveConnections init.conf directive 105
- max-pool-size attribute 50, 53, 78

- MaxRqHeaders init.conf directive 117
- max-thread-pool-size attribute 46
- max-wait-time-in-millis attribute 78
- mdb-container element 53
- MediumFileSizeLimit parameter 128
- MediumFileSpace parameter 128
- message-driven beans 53
- message-fragment-size attribute 45
- mime attribute 41
- mime element 35
- MIME types 119
- mime.types file
 - default 120
 - overview of 19
 - sample of 122
 - server.xml element referencing 36
 - syntax 121
- MinCGIStubs init.conf directive 110
- monitoring-enabled attribute 46, 48, 52, 54, 64

N

- name attribute
 - for auth-realm element 61
 - for connector-module element 87
 - for ejb-module element 85
 - for j2ee-application element 84
 - for jdbc-connection-pool element 78
 - for lifecycle-module element 83
 - for profiler element 68
 - for property element 29
 - for server element 28
 - for web-module element 85
- native library path, configuring 66
- native thread pools, settings in init.conf 107
- native-library-path attribute 68
- native-library-path-prefix attribute 66
- native-library-path-suffix attribute 66
- NativePoolMaxThreads init.conf directive 108
- NativePoolMinThreads init.conf directive 108
- NativePoolQueueSize init.conf directive 108
- NativePoolStackSize init.conf directive 108

- networkProtocol property 81
- nice variable 92
- nsessions property 124
- nsfc.conf file 127

O

- obj.conf file 18, 40
 - and virtual servers 18
- oracle-xa-recovery-workaround property 64
- orb element 45

P

- password property 80
- password.conf file 128
- persistence-manager-factory-resource element 76
- PidLog init.conf directive 111
- pool-idle-timeout-in-seconds attribute 51
- pool-name attribute 72
- pool-resize-quantity attribute 50, 53, 78
- port attribute 32, 47, 55
- port property 80
- PostThreadsEarly init.conf directive 105
- processes, settings in init.conf 102
- profiler element 67
- properties, about 28
- property element 28

Q

- quality of service 43

R

- RcvBufSize init.conf directive 105
- reconnect-delay-in-seconds property 54
- reconnect-enabled property 54
- reconnect-max-retries property 54
- ReentrantTimeFunctions init.conf directive 117
- removal-timeout-in-seconds attribute 52
- request-handling process 19
- request-response process 19
- requirement rules 24
- resource adapters 86
- resources element 69
- res-type attribute 70, 71, 75, 78
- rmic-options attribute 66
- roleName property 81
- RqThrottle init.conf directive 105
- RqThrottleMin init.conf directive 106

S

- search-bind-dn property 62
- search-bind-password property 62
- search-filter property 62
- Secure Socket Layer *see* SSL
- security 59
 - of listen sockets 32, 47
 - settings in init.conf 113
- Security init.conf directive 113
- security-enabled attribute 33
- security-service element 59
- send-cgi function, variables for 92
- server
 - basic operation 17
 - HUP signal 111
 - init.conf directives that configure 99
 - initialization directives in init.conf 97
 - killing process of 111
 - modifying 17
 - TERM signal 111
- server element 27
- server log file 57

- location 28, 58
- server.policy file 129
- server.xml file 18
 - elements in 26
 - alphabetical list 133
 - example of 93
 - location 23
- server-classpath attribute 66
- server-instance element 30
- server-name attribute 33
- serverName property 80
- session-store attribute 28
- single sign-on 42
- SmallFileSizeLimit parameter 128
- SmallFileSpace parameter 128
- SndBufSize init.conf directive 106
- SSL
 - configuring 34
 - settings in init.conf 113
- ssl element 34
- ssl2-ciphers attribute 34
- ssl2-enabled attribute 34
- SSL3 client authentication 35
- ssl3-enabled attribute 35
- SSL3SessionTimeout init.conf directive 114
- ssl3-tls-ciphers attribute 35
- SSLCacheEntries init.conf directive 114
- SSLClientAuthDataLimit init.conf directive 114
- SSLClientAuthTimeout init.conf directive 114
- ssl-client-config element 46
- SSLSessionTimeout init.conf directive 114
- sso-enabled property 42
- sso-max-inactive-seconds property 42
- sso-reap-interval-seconds property 42
- StackSize init.conf directive 106
- start-args attribute 56
- state attribute 41
- stderr output 58
- stdout output 58
- steady-pool-size attribute 50, 53, 78
- steady-thread-pool-size attribute 46
- store-protocol attribute 73

- store-protocol-class attribute 73
- StrictHttpHeaders init.conf directive 106
- subelements, about 24
- Sun customer support 15
- Sun ONE Application Server *see* server
- Sun ONE Message Queue 56
- sun-acc.xml file 126
- sun-application.xml file 126
- sun-application-client.xml file 126
- sun-cmp-mapping.xml file 126
- sun-ejb-jar.xml file 126
- sun-ra.xml file 126
- sun-server_1_0.dtd file 24
- sun-web.xml file 126
- sybase-xa-recovery-workaround property 64

T

- TempDir init.conf directive 99
- TempDir parameter 128
- TempDirSecurity init.conf directive 99
- TERM signal 111
- TerminateTimeout init.conf directive 107
- thread pools, settings in init.conf 107
- ThreadIncrement init.conf directive 107
- threads, settings in init.conf 102
- time formats 131
- timeout-in-seconds attribute 63
- tls-enabled attribute 35
- tls-rollback-enabled attribute 35
- transaction log file 57, 63
- transaction-isolation-level attribute 79
- transaction-service element 62
- TransmitFiles parameter 128
- transport-protocol attribute 73
- transport-protocol-class attribute 73
- tx-log-dir attribute 63
- type parameter 120
- type-by-extension function 120

U

- Umask init.conf directive 117
- UNIX user account, specifying 100
- url property 81
- UseOutputStreamSize init.conf directive 115
- user database selection 43, 87
- User init.conf directive 100
- user property 80
- user variable 92
- use-system-logging attribute 59

V

- validation-table-name attribute 80
- value attribute 29
- variables 90
 - evaluation 92
 - format of 91
 - referencing in obj.conf 90
 - substitution, where allowed 90
 - used in the interfaces 91
- victim-selection-policy attribute 52
- virtual servers
 - and the obj.conf file 18
 - log file for 57
- virtual-server element 38
- virtual-server-class element 37
- virtual-servers attribute 84, 86

W

- web module, default 40
- web-container element 48
- web-module element 85
- WincgiTimeout init.conf directive 110
- Windows console 58

X

xaresource-txn-timeout property 64