Sun GlassFish Enterprise Server v3 Prelude Administration Reference

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

Part No: 820–4507–10
October 2008

081028@21288

Contents

Preface

This *Administration Reference* provides information about the Sun GlassFish™ Enterprise Server configuration file, `domain.xml`. This file contains most of the Enterprise Server configuration.

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

The following topics are addressed here:

Enterprise Server Documentation Set

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

TABLE P–1 Books in the Enterprise Server Documentation Set

Book Title	Description
Release Notes	Provides late-breaking information about the software and the documentation. Includes a comprehensive, table-based summary of the supported hardware, operating system, Java™ Development Kit (JDK™), and database drivers.

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

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

Related Documentation

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

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

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

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

Typographic Conventions

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

TABLE P–2 Typographic Conventions

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

Symbol Conventions

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

TABLE P–3 Symbol Conventions

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

Default Paths and File Names

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

TABLE P–4 Default Paths and File Names

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

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

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

Documentation, Support, and Training

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

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

Searching Sun Product Documentation

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

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

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

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

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

Third-Party Web Site References

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

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

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Send Comments. In the online form, provide the full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the part number of this book is 820-4507.

◆ ◆ ◆ CHAPTER 1

The domain.xml File

This chapter describes the `domain.xml` configuration file in these sections:

Note – For GlassFish v3 Prelude, EJB modules are not supported unless the optional EJB container add-on component is downloaded from the Update Tool. Only stateless session beans with local interfaces and entity beans that use the Java Persistence API are supported. Stateful, message-driven, and EJB 2.0 and 2.1 entity beans are not supported. Remote interfaces and remote business interfaces for any of the bean types are not supported.

Web services are not supported unless the optional Metro (JSR 109) add-on component is downloaded from the Update Tool. Without the Metro add-on component, a servlet or EJB module cannot be a web service endpoint.

Global (XA) transactions are not supported unless the optional JTS and Object Management Group (OMG) add-on components are downloaded from the Update Tool. Without the these components, only local transactions are supported. Transaction recovery is not implemented for GlassFish v3 Prelude, even if the JTS and OMG add-on components are installed. Therefore, all transaction service attributes and properties pertaining to transaction recovery or transaction logs are not implemented.

JRuby applications are not supported unless the optional JRuby add-on component is downloaded from the Update Tool.

For information about the Update Tool, see the *Sun GlassFish Enterprise Server v3 Prelude Installation Guide*.

Note – GlassFish v3 Prelude does not support clustering, load balancing features, or Sun GlassFish Message Queue software. Elements related to these features are ignored.

Note – Subelements must be defined in the order in which they are listed under each **Subelements** heading in this chapter unless otherwise noted.

About the domain.xml File

The `domain.xml` file contains most of the Sun Java™ System Enterprise Server configuration. The `domain.xml` file is located in the domain configuration directory, which is typically *domain-dir*`/config`.

There is no DTD file or schema that determines the format of the `domain.xml` file. The server's configuration is dictated by the Java interfaces with appropriate annotations and `domain.xml` serves as a seed for that configuration. At any point in time, the overall structure of an XML element present in `domain.xml` is clearer when one sees the Java interface in the package `com.sun.enterprise.config.serverbeans`. Thus these Java interfaces take the place of the DTD file in previous releases.

This file is further described in the following sections:

- "Default Values" on page 14
- "Variables" on page 15
- "Element Referencing" on page 16
- "Element Hierarchy" on page 16

Note – Settings in the Enterprise Server deployment descriptors override corresponding settings in the `domain.xml` file unless otherwise stated. For more information about the Enterprise Server deployment descriptors, see the *Sun GlassFish Enterprise Server v3 Prelude Application Deployment Guide*.

Default Values

In this manual, the term *default* is used in its broader sense, and not in the specific way it is used in the XML 1.0 standard. A default value is an initial value or the value used if no value is present in the XML file. A default value can be any of the following:

- A value supplied by the XML parser when no value is found in the `domain.xml` file. The relevant element or attribute is optional.

- A value supplied by the Enterprise Server when no value is found in the `domain.xml` file and the XML parser doesn't provide a value. The relevant element or attribute is optional.
- An initial value supplied when the `domain.xml` file is created. The relevant element or attribute might or might not be optional.

Note – Removal of initial values in `domain.xml` is not recommended because attributes may be left without values or values may revert to internal defaults. Explicitly changing values is recommended.

Variables

Variables and variable references are needed for two reasons:

- Parts of the Enterprise Server share much configuration information but differ in specific details.
- Parts of the configuration come from the system environment but must still be captured in the configuration.

Variable references appear in the `domain.xml` file as strings that begin with the characters `${` and end with the character `}`. For example, the string `${com.sun.enterprise.myVar}` is a reference to the variable `com.sun.enterprise.myVar`.

Variables are defined both outside of and within `domain.xml`. Predefined variables that exist outside of `domain.xml` are defined as Java System Properties. Within `domain.xml`, a variable is defined using the "system-property" on page 87 element or the "jvm-options" on page 61 element.

The `system-property` element's `name` attribute is the name of a variable; its `value` attribute is the definition of the variable. For example, the following `system-property` element defines a `port-number` variable with the value `6500`:

```
<system-property name="port-number" value="6500"/>
```

Multiple `system-property` subelements are permitted within "server" on page 80, "config" on page 27, and "domain" on page 30 elements.

A variable defined in the `jvm-options` element is a Java System Property with the `-D` flag. For example, the following `jvm-options` element defines a `port-number` variable with the value `5500`:

```
<jvm-option>-Dport-number=5500</jvm-option>
```

Multiple definitions for the same variable are permitted. The Enterprise Server determines the actual value of a variable by searching for its first definition in a strict hierarchy of the elements within `domain.xml`. The hierarchy is as follows:

```
server → config → jvm-options → domain → System
```

Implicit in this hierarchy is the notion of reference and containment. A variable referenced in a `server` element is only looked up:

- In the `config` element that references that specific `server`
- In the `jvm-options` subelements of the `config` element referenced by that `server`

Element Referencing

One element *references* another when an attribute of the referencing element has the same value as an attribute of the referenced element. For example, the "application-ref" on page 22 element references a module that is deployed to its parent "server" on page 80 element. The `application-ref` element's `ref` attribute has the same value as the `name` attribute of an "application" on page 20 element.

The referencing `application-ref` element might look like this:

```
<application-ref ref="MyServlet"/>
```

The referenced "application" on page 20 element might look like this:

```
<application name="MyServlet" location="myservletdir"/>
```

Element Hierarchy

The element hierarchy for the `domain.xml` file is as follows. To make the hierarchy more readable, elements having "property" on page 73 as their last or only subelement are marked with a `P`, and the `property` subelements are not shown. Parent/child relationships between elements are shown, but not cardinality. For those details, see the element descriptions.

```
domain      P
.    system-applications
.    .    application      P
.    .   .    engine      P
.    applications
.    .    application      P
.    .   .    engine      P
.    resources
.    .    jdbc-resource      P
.    .    jdbc-connection-pool      P
.    configs
.    .    config      P
.    .   .    http-service      P
.    .   .    .    access-log
```

```
. . . . request-processing
. . . . keep-alive
. . . . connection-pool
. . . . http-protocol
. . . . http-file-cache
. . . . http-listener      P
. . . . . ssl
. . . . virtual-server      P
. . . . . http-access-log
. . . iiop-service
. . . . orb      P
. . . . ssl-client-config
. . . . . ssl
. . . . iiop-listener      P
. . . . . ssl
. . . admin-service      P
. . . . jmx-connector      P
. . . . . ssl
. . . . das-config      P
. . . web-container      P
. . . . session-config
. . . . . session-manager
. . . . . . manager-properties      P
. . . . . . store-properties      P
. . . . . session-properties      P
. . . ejb-container      P
. . . . ejb-timer-service      P
. . . mdb-container      P
. . . jms-service      P
. . . . jms-host      P
. . . log-service      P
. . . . module-log-levels      P
. . . security-service      P
. . . . auth-realm      P
. . . . jacc-provider      P
. . . . audit-module      P
. . . . message-security-config
. . . . . provider-config      P
. . . . . . request-policy
. . . . . . response-policy
. . . transaction-service      P
. . . monitoring-service      P
. . . . module-monitoring-levels      P
. . . java-config      P
. . . . profiler      P
. . . . . jvm-options
. . . . jvm-options
. . . thread-pools
```

```
.    .    .    .    thread-pool      P
.    .    .    system-property
.    servers
.    .    server      P
.    .    .    application-ref
.    .    .    resource-ref
.    .    .    system-property
.    system-property
```

Alphabetical List of Elements

A

access-log

Defines access log settings for each "http-access-log" on page 37 subelement of each "virtual-server" on page 92.

Superelements

"http-service" on page 44

Subelements

none

Attributes

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

TABLE 1–1 `access-log` Attributes

Attribute	Default	Description
`format`	`%client.name% %auth-user-name% %datetime% %request% %status% %response.length%`	(optional) Specifies the format of the access log. For a complete list of token values you can use in the format, see the online help for the Access Log tab of the HTTP Service page in the Administration Console.
`rotation-policy`	`time`	(optional) Specifies the condition that triggers log rotation. The only legal value is `time`, which rotates log files at the `rotation-interval-in-minutes` interval.
`rotation-interval -in-minutes`	`15`	(optional) Specifies the time interval between log rotations if `rotation-policy` is set to `time`.
`rotation-suffix`	`yyyy-MM-dd`	(optional) Specifies the format of the timestamp appended to the access log name when log rotation occurs. For supported formats, see http://java.sun.com/j2se/1.5.0/docs/api/java/text/SimpleDateFormat.html. The following value is supported for backward compatibility. It results in the same format as the default. `%YYYY;%MM;%DD;-%hh;h%mm;m%ss;s`
`rotation-enabled`	`true`	(optional) If `true`, enables log rotation.

admin-service

Contains configuration for JMX connectors, the domain admin server (DAS), and related properties.

Superelements

"config" on page 27

Subelements

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

TABLE 1–2 `admin-service` Subelements

Element	Required	Description
"jmx-connector" on page 60	zero or more	Configures a JSR 160/255 compliant remote JMX connector, which responds to JConsole port 8686.
"das-config" on page 29	only one	Defines a domain administration server configuration.
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–3 `admin-service` Attributes

Attribute	Default	Description
`type`	`das-and-server`	Specifies whether the server instance is a regular instance (`server`), a domain administration server (`das`), or a combination (`das-and-server`). modifying this value is not recommended.
`system-jmx-connector-name`	none	Specifies the name of the internal "jmx-connector" on page 60.

application

Specifies a system application or a Java EE module.

An "engine" on page 35 runs a sniffer during deployment, which is responsible for identifying a type of deployment artifact (such as a WAR file) and setting up the associated container (such as the web container). Multiple engines, each with its own sniffer, can be associated with a given application.

The `application` element replaces the `web-module` and `ejb-module` elements of previous releases.

Note – For GlassFish v3 Prelude, EJB modules are not supported unless the optional EJB container add-on component is downloaded from the Update Tool. Only stateless session beans with local interfaces and entity beans that use the Java Persistence API are supported. Stateful, message-driven, and EJB 2.0 and 2.1 entity beans are not supported. Remote interfaces and remote business interfaces for any of the bean types are not supported.

Web services are not supported unless the optional Metro (JSR 109) add-on component is downloaded from the Update Tool. Without the Metro component, a servlet or EJB module cannot be a web service endpoint.

JRuby applications are not supported unless the optional JRuby add-on component is downloaded from the Update Tool.

For information about the Update Tool, see the *Sun GlassFish Enterprise Server v3 Prelude Installation Guide*.

Superelements

"system-applications" on page 86, "applications" on page 23

Subelements

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

TABLE 1–4 `application` Subelements

Element	Required	Description
"engine" on page 35	one or more	Configures an engine.
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–5 `application` Attributes

Attribute	Default	Description
`name`	none	The name of the application.
`description`	none	(optional) Specifies a text description of this element.

TABLE 1–5 `application` Attributes *(Continued)*

Attribute	Default	Description
`location`	none	The location of the application in the Enterprise Server file system. If a relative path is specified, it is relative to the *domain-dir*`/applications/` directory. **Note –** Deployment directories may change between Enterprise Server releases.
`object-type`	`system-admin`	(optional) Defines the type of the resource. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`enabled`	`true`	(optional) Determines whether the application is enabled.
`context-root`	none	The context root at which the application is deployed. The context root can be the empty string or just `/`. The context root can start with the `/` character, but doesn't have to.
`directory-deployed`	`false`	(optional) Specifies whether the application has been deployed as a directory.

application-ref

References a module deployed to the server.

Superelements

"server" on page 80

Subelements

none

Attributes

The following table describes attributes for the `application-ref` element.

TABLE 1–6 `application-ref` Attributes

Attribute	Default	Description
`enabled`	`true`	(optional) Determines whether the application or module is enabled.

TABLE 1–6 `application-ref` Attributes *(Continued)*

Attribute	Default	Description
`virtual-servers`	all virtual servers	(optional) In a comma-separated list, references `id` attributes of the "virtual-server" on page 92 elements to which the web "application" on page 20 is deployed. If you deploy a web application and don't specify any assigned virtual servers, the web application is assigned to all currently defined virtual servers. If you then create additional virtual servers and want to assign existing web applications to them, you must redeploy the web applications. For more information about deployment, see the *Sun GlassFish Enterprise Server v3 Prelude Application Deployment Guide*.
`disable-timeout-in-minutes`	`30`	(optional) Specifies the time it takes this application to reach a quiescent state after having been disabled.
`ref`	none	References the `name` attribute of an "application" on page 20 element.

applications

Contains deployed Java EE modules.

Superelements

"domain" on page 30

Subelements

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

TABLE 1–7 `applications` Subelements

Element	Required	Description
"application" on page 20	zero or more	Specifies an application.

audit-module

Specifies an optional plug-in module that implements audit capabilities.

Superelements

"security-service" on page 79

Subelements

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

TABLE 1–8 `audit-module` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–9 `audit-module` Attributes

Attribute	Default	Description
`name`	`default`	Specifies the name of this audit module.
`classname`	`com.sun.enterprise.security.Audit`	Specifies the Java class that implements this audit module.

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 GlassFish Enterprise Server v3 Prelude Administration Guide*.

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

```
<auth-realm name="file"
     classname="com.sun.enterprise.security.auth.realm.file.FileRealm">
     <property name="file" value="${com.sun.aas.instanceRoot}/config/admin-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.

Superelements

"security-service" on page 79

Subelements

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

TABLE 1–10 `auth-realm` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–11 `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 Enterprise Server have required and optional properties. A custom realm might have different properties.

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

TABLE 1–12 `auth-realm` Properties

Property	Realms	Description
`jaas-context`	all	Specifies the JAAS (Java Authentication and Authorization Service) context.
`file`	`file`	Specifies the file that stores user names, passwords, and group names. The default is *domain-dir*`/config/keyfile`.
`assign-groups`	all	(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 `uid=%s` (`%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 `base-dn`, but it can be tuned, if necessary.

TABLE 1–12 `auth-realm` Properties *(Continued)*

Property	Realms	Description
`group-search-filter`	`ldap`	(optional) Specifies the search filter to find group memberships for the user. The default is `uniquemember=%d` (`%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 `search-filter` 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 `search-bind-dn` .
`datasource-jndi`	`jdbc`	Specifies the `jndi-name` of the "jdbc-resource" on page 56 for the database.
`user-table`	`jdbc`	Specifies the name of the user table in the database.
`user-name-column`	`jdbc`	Specifies the name of the user name column in the database's user table.
`password-column`	`jdbc`	Specifies the name of the password column in the database's user table.
`group-table`	`jdbc`	Specifies the name of the group table in the database.
`group-name-column`	`jdbc`	Specifies the name of the group name column in the database's group table.
`db-user`	`jdbc`	(optional) Allows you to specify the database user name in the realm instead of the "jdbc-connection-pool" on page 51. This prevents other applications from looking up the database, getting a connection, and browsing the user table. By default, the `jdbc-connection-pool` configuration is used.
`db-password`	`jdbc`	(optional) Allows you to specify the database password in the realm instead of the "jdbc-connection-pool" on page 51. This prevents other applications from looking up the database, getting a connection, and browsing the user table. By default, the `jdbc-connection-pool` configuration is used.
`digest-algorithm`	`jdbc`	(optional) Specifies the digest algorithm. The default is `MD5`. You can use any algorithm supported in the JDK, or `none`.
`encoding`	`jdbc`	(optional) Specifies the encoding. Allowed values are `Hex` and `Base64`. If `digest-algorithm` is specified, the default is `Hex`. If `digest-algorithm` is not specified, by default no encoding is specified.
`charset`	`jdbc`	(optional) Specifies the charset for the digest algorithm.

C

config

Defines a configuration, which is a collection of settings that controls how a server instance functions.

Superelements

"configs" on page 28

Subelements

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

TABLE 1–13 `config` Subelements

Element	Required	Description
"iiop-service" on page 48	only one	Configures the IIOP service.
"admin-service" on page 19	only one	Determines whether the server to which the configuration applies is an administration server.
"web-container" on page 98	only one	Configures the web container.
"ejb-container" on page 32	only one	Configures the Enterprise JavaBeans™ (EJB™) container.
"mdb-container" on page 65	only one	Configures the message-driven bean (MDB) container.
"jms-service" on page 58	zero or one	Configures the Java Message Service (JMS) provider.
"log-service" on page 63	only one	Configures the system logging service.
"security-service" on page 79	only one	Configures the Java EE security service.
"transaction-service" on page 90	only one	Configures the transaction service.
"monitoring-service" on page 70	only one	Configures the monitoring service.
"java-config" on page 50	only one	Configures the Virtual Machine for the Java platform (JVM™ software).
"thread-pools" on page 89	only one	Configures thread pools.
"system-property" on page 87	zero or more	Specifies a system property.
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–14 `config` Attributes

Attribute	Default	Description
`name`	`server-config`	Specifies the name of the configuration.
`dynamic-reconfiguration-enabled`	`true`	(optional) If `true`, any changes to the system (for example, applications deployed, resources created) are automatically applied to the affected servers without a restart being required. If false, such changes are only picked up by the affected servers when each server restarts.

configs

Contains configurations.

Superelements

"domain" on page 30

Subelements

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

TABLE 1–15 `configs` Subelements

Element	Required	Description
"config" on page 27	only one	Defines a configuration.

connection-pool

Defines a pool of client HTTP connections used by the "http-listener" on page 39 subelements of the parent "http-service" on page 44 element.

Superelements

"http-service" on page 44

Subelements

none

Attributes

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

TABLE 1–16 `connection-pool` Attributes

Attribute	Default	Description
`queue-size-in-bytes`	`4096`	(optional) Specifies the maximum number of messages that can be queued until threads are available to process them for "http-listener" on page 39 elements. A value of `-1` specifies no limit.
`max-pending-count`	`4096`	(optional) Specifies the maximum number of pending connections on an "http-listener" on page 39.
`receive-buffer-size-in-bytes`	`4096`	(optional) Specifies the size of the receive buffer for all "http-listener" on page 39 elements.
`send-buffer-size-in-bytes`	`8192`	(optional) Specifies the size of the send buffer for all "http-listener" on page 39 elements.

D

das-config

Defines a domain administration server configuration. The domain administration server runs the Administration Console.

Superelements

"admin-service" on page 19

Subelements

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

TABLE 1–17 `das-config` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `das-config` element. For more information about deployment topics such as dynamic reloading and autodeployment, see the *Sun GlassFish Enterprise Server v3 Prelude Developer's Guide*.

TABLE 1–18 `das-config` Attributes

Attribute	Default	Description
`dynamic-reload-enabled`	`true`	(optional) If `true`, checks the timestamp on a `.reload` file at every module and application directory level, to trigger dynamic reloading.
`dynamic-reload-poll-interval-in-seconds`	`2`	(optional) Controls the polling frequency of dynamic reloading.
`autodeploy-enabled`	`true`	(optional) If `true`, enables autodeployment, which lets you quickly deploy applications and modules to a running Enterprise Server without performing an explicit server instance restart or a separate deployment operation.
`autodeploy-polling-interval-in-seconds`	`2`	(optional) Controls the polling frequency of autodeployment.
`autodeploy-dir`	`autodeploy`	(optional) Specifies the source directory (absolute or relative to *domain-dir*) in which autodeployment looks for deployable components.
`autodeploy-verifier-enabled`	`false`	(optional) If `true`, the verifier is run before autodeployment. If verification fails, deployment is not performed.
`autodeploy-jsp-precompilation-enabled`	`false`	(optional) If `true`, JSP pages are precompiled during autodeployment.
`auto-deploy-retry-timeout`	`4`	(optional) Specifies the number of autodeployment retry attempts before autodeployment times out.
`deploy-xml-validation`	`full`	(optional) Specifies the type of XML validation performed on standard and Enterprise Server deployment descriptors: ■ `full` - If XML validation fails, deployment fails. ■ `parsing` - XML validation errors are reported but deployment occurs. ■ `none` - No XML validation is performed.
`admin-session-timeout-in-minutes`	`sun-web.xml` `timeoutSeconds` property value or `web.xml` `session-timeout` attribute value	(optional) Specifies the Administration Console timeout.

domain

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

Superelements

none

Subelements

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

TABLE 1–19 `domain` Subelements

Element	Required	Description
"system-applications" on page 86	zero or one	Contains system applications.
"applications" on page 23	zero or one	Contains deployed Java EE applications, Java EE modules, and lifecycle modules.
"resources" on page 77	zero or one	Contains configured resources.
"configs" on page 28	only one	Contains configurations.
"servers" on page 81	only one	Contains server instances.
"system-property" on page 87	zero or more	Specifies a system property.
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–20 `domain` Attributes

Attribute	Default	Description
`application-root`	*domain-dir*`/applications`	(optional) Specifies the absolute path where deployed applications reside for this domain.
`log-root`	*domain-dir*`/logs`	(optional) Specifies where the domain's log files are kept. The directory in which the log is kept must be writable by whatever user account the server runs as. See the "log-service" on page 63 description for details about logs.
`locale`	operating system default	(optional) Specifies the domain's language.

E

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.

Note – For GlassFish v3 Prelude, EJB modules are not supported unless the optional EJB container add-on component is downloaded from the Update Tool. Only stateless session beans with local interfaces and entity beans that use the Java Persistence API are supported. Stateful, message-driven, and EJB 2.0 and 2.1 entity beans are not supported. Remote interfaces and remote business interfaces for any of the bean types are not supported.

Superelements

"config" on page 27

Subelements

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

TABLE 1–21 `ejb-container` Subelements

Element	Required	Description
"ejb-timer-service" on page 34	zero or one	Configures the EJB timer service.
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–22 `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 `max-pool-size` . 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 removed when the `pool-idle-timeout-in-seconds` timer expires. A cleaner thread removes any unused instances. Must be `0` or greater and less than `max-pool-size` . The pool is not resized below the `steady-pool-size`. 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.
`cache-resize-quantity`	`32`	(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 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 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 never become candidates for passivation. Applies to entity beans.

TABLE 1–22 `ejb-container` Attributes *(Continued)*

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" on page 80 element determines the location of the session store. Applies to stateful session beans. **Note –** This attribute is not implemented for GlassFish v3 Prelude.
`victim-selection-policy`	`nru`	(optional) Specifies how 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. **Note –** This attribute is not implemented for GlassFish v3 Prelude.
`commit-option`	`B`	(optional) Determines which commit option is used for entity beans. Legal values are `B` or `C`. **Note –** This attribute is not implemented for GlassFish v3 Prelude.
`session-store`	*domain-dir*`/session-store`	(optional) Specifies the directory where passivated stateful session beans and persisted HTTP sessions are stored in the file system. **Note –** This attribute is not implemented for GlassFish v3 Prelude.

ejb-timer-service

Configures the EJB timer service.

Superelements

"ejb-container" on page 32

Subelements

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

TABLE 1–23 `ejb-timer-service` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–24 `ejb-timer-service` Attributes

Attribute	Default	Description
`minimum-delivery-interval-in-millis`	`7000`	(optional) Specifies the minimum time before an expiration for a particular timer can occur. This guards against extremely small timer increments that can overload the server.
`max-redeliveries`	`1`	(optional) Specifies the maximum number of times the EJB timer service attempts to redeliver a timer expiration due for exception or rollback.
`timer-datasource`	`jdbc/__TimerPool`	(optional) Overrides, for the server instance, the `cmp-resource` value specified in `sun-ejb-jar.xml` for the timer service system application (`__ejb_container_timer_app`). **Note –** This attribute is not implemented for GlassFish v3 Prelude.
`redelivery-interval-internal-in-millis`	`5000`	(optional) Specifies how long the EJB timer service waits after a failed `ejbTimeout` delivery before attempting a redelivery.

engine

Specifies an engine for an "application" on page 20. An engine runs a sniffer during deployment, which is responsible for identifying a type of deployment artifact (such as a WAR file) and setting up the associated container (such as the web container). Multiple engines, each with its own sniffer, can be associated with a given application.

Note – For GlassFish v3 Prelude, EJB modules are not supported unless the optional EJB container add-on component is downloaded from the Update Tool. Only stateless session beans with local interfaces and entity beans that use the Java Persistence API are supported. Stateful, message-driven, and EJB 2.0 and 2.1 entity beans are not supported. Remote interfaces and remote business interfaces for any of the bean types are not supported.

Web services are not supported unless the optional Metro (JSR 109) add-on component is downloaded from the Update Tool. Without the Metro component, a servlet or EJB module cannot be a web service endpoint.

JRuby applications are not supported unless the optional JRuby add-on component is downloaded from the Update Tool.

For information about the Update Tool, see the *Sun GlassFish Enterprise Server v3 Prelude Installation Guide*.

Superelements

"application" on page 20

Subelements

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

TABLE 1–25 `engine` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–26 `engine` Attributes

Attribute	Default	Description
`sniffer`	none	Specifies the type of sniffer. Allowed values are as follows: ■ `web` — Specifies that the parent application is a web application. ■ `security` — Specifies that security is enabled for the parent application. ■ `jpa` — Specifies that the parent application uses the Java Persistence API. ■ `ejb` — Specifies that the parent application includes an EJB module. The EJB container add-on component must be installed in the Enterprise Server. ■ `webservices` — Specifies that the parent application is a web service endpoint. The Metro add-on component must be installed in the Enterprise Server. ■ `jruby` — Specifies that the parent application is a JRuby application. The JRuby add-on component must be installed in the Enterprise Server.

H

http-access-log

Defines an access log file for a "virtual-server" on page 92. The "access-log" on page 18 subelement of the virtual server's parent "http-service" on page 44 element determines the access log file's format and rotation settings.

Superelements

"virtual-server" on page 92

Subelements

none

Attributes

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

TABLE 1–27 `http-access-log` Attributes

Attribute	Default	Description
`log-directory`	`${com.sun.aas.instanceRoot}/logs/access`	(optional) Specifies the location of the access log file. The `${com.sun.aas.instanceRoot}` system property refers to the *domain-dir*. See "system-property" on page 87.
`iponly`	`true`	(optional) If `true`, specifies that only the IP address of the user agent is listed. If `false`, performs a DNS lookup.

http-file-cache

Configures the HTTP file cache.

Superelements

"http-service" on page 44

Subelements

none

Attributes

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

TABLE 1–28 `http-file-cache` Attributes

Attribute	Default	Description
`globally-enabled`	`false`	(optional) If `true`, enables the file cache.
`file-caching-enabled`	`false`	(optional) If `true`, enables caching of the file content if the file size exceeds the `small-file-size-limit-in-bytes`.
`max-age-in-seconds`	`30`	(optional) Specifies the maximum age of a file cache entry.
`medium-file-size-limit-in-bytes`	`537600`	(optional) Specifies the maximum size of a file that can be cached as a memory mapped file.
`medium-file-space-in-bytes`	`1048576`	(optional) Specifies the total size of all files that are cached as memory mapped files.
`small-file-size-limit-in-bytes`	`2048`	(optional) Specifies the maximum size of a file that can be read into memory.

TABLE 1–28 `http-file-cache` Attributes *(Continued)*

Attribute	Default	Description
`small-file-space-in-bytes`	`1048576`	(optional) Specifies the total size of all files that are read into memory.
`file-transmission-enabled`	`false`	(optional) If `true`, enables the use of `TransmitFileSystem` calls. Meaningful only for Windows.
`max-files-count`	`1024`	(optional) Specifies the maximum number of files in the file cache.
`hash-init-size`	`0`	(optional) Specifies the initial number of hash buckets.

http-listener

Defines an HTTP listen socket. The "connection-pool" on page 28 subelement of the parent "http-service" on page 44 element also configures some listen socket settings.

When you first install the Enterprise Server, three `http-listener` elements exist by default, with `id` values of `admin-listener`, `http-listener1`, and `http-listener2`. The `http-listener2`, which is set up with SSL, is disabled by default.

Superelements

"http-service" on page 44

Subelements

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

TABLE 1–29 `http-listener` Subelements

Element	Required	Description
"ssl" on page 84	zero or one	Defines Secure Socket Layer (SSL) parameters.
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–30 `http-listener` Attributes

Attribute	Default	Description
`id`	none	The unique listener name. An `http-listener` name cannot begin with a number.

TABLE 1–30 `http-listener` Attributes *(Continued)*

Attribute	Default	Description
`address`	none	IP address of the listener. Can be in dotted-pair or IPv6 notation. Can be `any` (for `INADDR_ANY`) to listen on all IP addresses. Can be a hostname.
`port`	none	Port number on which the listener listens. Legal values are `1` - `65535`. On UNIX, creating sockets that listen on ports `1` - `1024` requires superuser privileges. Configuring an SSL listener to listen on port 443 is standard.
`external-port`	none	(optional) Specifies the external port on which the connection is made.
`family`		(optional) Deprecated. Do not use.
`blocking-enabled`	`false`	(optional) If `true`, uses a blocking socket for servicing a request.
`acceptor-threads`	`1`	(optional) Specifies the number of processors in the machine. To set the number of request processing threads, use the `thread-count` attribute of the "request-processing" on page 76 element.
`security-enabled`	`false`	(optional) Determines whether the listener runs SSL. To turn SSL2 or SSL3 on or off and set ciphers, use an `ssl` subelement.
`default-virtual-server`	none	References the `id` attribute of the default "virtual-server" on page 92 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. If your server uses an alias, the `server-name` should be the alias name. If a colon and port number are appended, that port is used in URLs the server sends to the client. If load balancing is enabled, use the server name of the load balancer.
`redirect-port`	none	(optional) If the listener is supporting non-SSL requests and a request is received for which a matching `<security-constraint>` requires SSL transport, the request is automatically redirected to the port number specified here. If load balancing is enabled, use the redirect port of the load balancer.
`xpowered-by`	`true`	(optional) If true, `X-Powered-By` headers are used according to the Servlet 2.4 and JSP 2.0 specifications.
`enabled`	`true`	(optional) Determines whether the listener is active. If set to `false`, any attempts to connect to the listener result in a socket exception (`java.net.ConnectException`). In Enterprise Server versions prior to 9.1, a listener whose `enabled` attribute was set to `false` returned a 404 response code for any requests sent to it. To achieve this behavior in the current Enterprise Server version, set the listener's `enabled` attribute to `true`, and set every associated virtual server's state to `off`. A "virtual-server" on page 92 lists its associated listeners in its `http-listeners` attribute.

Properties

The following table describes properties for the `http-listener` element. Any of these properties can be defined as an "http-service" on page 44 property, so that it applies to all `http-listener` elements.

TABLE 1–31 `http-listener` Properties

Property	Default	Description
`recycle-objects`	`true`	If `true`, recycles internal objects instead of using the VM garbage collector.
`reader-threads`	`0`	Specifies the number of reader threads, which read bytes from the non-blocking socket.
`acceptor-queue-length`	`4096`	Specifies the length of the acceptor thread queue. Once full, connections are rejected.
`reader-queue-length`	`4096`	Specifies the length of the reader thread queue. Once full, connections are rejected.
`use-nio-direct-bytebuffer`	`true`	If `true`, specifies that the NIO direct `ByteBuffer` is used. In a limited resource environment, it might be faster to use non-direct Java's `ByteBuffer` by setting a value of `false`.
`authPassthrough Enabled`	`false`	If `true`, indicates that this `http-listener` element receives traffic from an SSL-terminating proxy server. Overrides the `authPassthroughEnabled` property of the parent "http-service" on page 44 element.
`proxyHandler`	`com.sun.enterprise.web.ProxyHandlerImpl`	Specifies the fully qualified class name of a custom implementation of the `com.sun.appserv.ProxyHandler` abstract class that this `http-listener` uses. Only used if the `authPassthroughEnabled` property of this `http-listener` and the parent "http-service" on page 44 element are both set to `true`. Overrides the `proxyHandler` property of the parent `http-service` element.
`proxiedProtocol`	none	Specifies a comma-separated list of protocols that can use the same port. Allowed values are `ws/tcp` (SOAP over TCP), `http`, `https` and `tls`. For example, if you set this property to `http,https` and set the `port` to `4567`, you can access the port with either `http://`*host*`:4567/` or `https://`*host*`:4567/`. Specifying this property at the "http-service" on page 44 level overrides settings at the `http-listener` level. If this property is not set at either level, this feature is disabled.
`bufferSize`	`4096`	Specifies the size, in bytes, of the buffer to be provided for input streams created by HTTP listeners.
`connectionTimeout`	`30`	Specifies the number of seconds HTTP listeners wait, after accepting a connection, for the request URI line to be presented.

TABLE 1–31 `http-listener` Properties *(Continued)*

Property	Default	Description
`maxKeepAliveRequests`	`250`	Specifies the maximum number of HTTP requests that can be pipelined until the connection is closed by the server. Set this property to `1` to disable HTTP/1.0 keep-alive, as well as HTTP/1.1 keep-alive and pipelining.
`traceEnabled`	`true`	If `true`, enables the TRACE operation. Set this property to `false` to make the Enterprise Server less susceptible to cross-site scripting attacks.

`cometSupport`	`false`	If `true`, enables Comet support for this listener.
		If your servlet or JSP page uses Comet technology, make sure it is initialized when the Enterprise Server starts up by adding the `load-on-startup` element to your `web.xml` file. For example:
		`<servlet>`
		`  <servlet-name>CheckIn</servlet-name>`
		`  <servlet-class>CheckInServlet</servlet-class>`
		`  <load-on-startup>0</load-on-startup>`
		`</servlet>`

`jkEnabled`	`false`	If `true`, enables `mod_jk` support for this listener.
`compression`	`off`	Specifies use of HTTP/1.1 GZIP compression to save server bandwidth. Allowed values are: ■ `off` – Disables compression. ■ `on` – Compresses data. ■ `force` – Forces data compression in all cases. ■ positive integer – Specifies the minimum amount of data required before the output is compressed. If the `content-length` is not known, the output is compressed only if compression is set to `on` or `force`.
`compressableMimeType`	`text/html,text/xml, text/plain`	Specifies a comma-separated list of MIME types for which HTTP compression is used.
`noCompression UserAgents`	empty `String` (`regexp` matching disabled)	Specifies a comma-separated list of regular expressions matching user-agents of HTTP clients for which compression should not be used.
`minCompressionSize` or `compressionMinSize`	none	Specifies the minimum size of a file when compression is applied.
`crlFile`	none	Specifies the location of the Certificate Revocation List (CRL) file to consult during SSL client authentication. This can be an absolute or relative file path. If relative, it is resolved against *domain-dir*. If unspecified, CRL checking is disabled.
`trustAlgorithm`	none	Specifies the name of the trust management algorithm (for example, `PKIX`) to use for certification path validation.

TABLE 1–31 `http-listener` Properties *(Continued)*

Property	Default	Description
`trustMaxCertLength`	`5`	Specifies the maximum number of non-self-issued intermediate certificates that can exist in a certification path. This property is considered only if `trustAlgorithm` is set to `PKIX`. A value of zero implies that the path can only contain a single certificate. A value of `-1` implies that the path length is unconstrained (there is no maximum). Setting a value less than `-1` causes an exception to be thrown.
`disableUploadTimeout`	`true`	if `false`, the connection for a servlet that reads bytes slowly is closed after the `connectionUploadTimeout` is reached.
`connectionUpload Timeout`	`5`	Specifies the timeout for uploads. Applicable only if `disableUploadTimeout` is set to `false`.
`uriEncoding`	`UTF-8`	Specifies the character set used to decode the request URIs received on this HTTP listener. Must be a valid IANA character set name. Overrides the `uriEncoding` property of the parent "http-service" on page 44 element.

http-protocol

Configures HTTP protocol settings.

Superelements

"http-service" on page 44

Subelements

none

Attributes

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

TABLE 1–32 `http-protocol` Attributes

Attribute	Default	Description
`version`	`HTTP/1.1`	(optional) Specifies the version of the HTTP protocol used.
`dns-lookup-enabled`	`false`	(optional) If `true`, looks up the DNS entry for the client.
`forced-type`	`text/html; charset=iso-8859-1`	(optional) Specifies the request type used if no MIME mapping is available that matches the file extension. The format is a semicolon-delimited string consisting of the content-type, encoding, language, and charset.

TABLE 1–32 `http-protocol` Attributes *(Continued)*

Attribute	Default	Description
`default-type`	`text/html; charset=iso-8859-1`	(optional) Specifies the default response type. The format is a semicolon-delimited string consisting of the content-type, encoding, language, and charset.
`forced-response-type`	`text/plain; charset=iso-8859-1`	(optional) Deprecated. Do not use.
`default-response-type`	`text/plain; charset=iso-8859-1`	(optional) Deprecated. Do not use.
`ssl-enabled`	`true`	(optional) Not implemented. Use `ssl` subelements of "http-listener" on page 39 elements.

http-service

Defines the HTTP service.

Superelements

"config" on page 27

Subelements

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

TABLE 1–33 `http-service` Subelements

Element	Required	Description
"access-log" on page 18	zero or one	Defines access log settings for each "http-access-log" on page 37 subelement of each "virtual-server" on page 92.
"http-listener" on page 39	one or more	Defines an HTTP listen socket.
"virtual-server" on page 92	one or more	Defines a virtual server.
"request-processing" on page 76	zero or one	Configures request processing threads.
"keep-alive" on page 62	zero or one	Configures keep-alive threads.
"connection-pool" on page 28	zero or one	Defines a pool of client HTTP connections.
"http-protocol" on page 43	zero or one	Configures HTTP protocol settings.
"http-file-cache" on page 38	zero or one	Configures the HTTP file cache.
"property" on page 73	zero or more	Specifies a property or a variable.

Properties

The following table describes properties for the `http-service` element. These properties apply to all "http-listener" on page 39 subelements, except for `accessLoggingEnabled`, `accessLogBufferSize`, and `accessLogWriterInterval`, which apply to all "virtual-server" on page 92 subelements.

TABLE 1–34 `http-service` Properties

Property	Default	Description
`monitoring-cache-enabled`	`true`	If `true`, enables the monitoring cache.
`monitoring-cache-refresh-in-millis`	`5000`	Specifies the interval between refreshes of the monitoring cache.
`ssl-cache-entries`	`10000`	Specifies the number of SSL sessions to be cached.
`ssl3-session-timeout`	`86400`	Specifies the interval at which SSL3 sessions are cached.
`ssl-session-timeout`	`100`	Specifies the interval at which SSL2 sessions are cached.
`recycle-objects`	`true`	If `true`, recycles internal objects instead of using the VM garbage collector.
`reader-threads`	`0`	Specifies the number of reader threads, which read bytes from the non-blocking socket.
`acceptor-queue-length`	`4096`	Specifies the length of the acceptor thread queue. Once full, connections are rejected.
`reader-queue-length`	`4096`	Specifies the length of the reader thread queue. Once full, connections are rejected.
`use-nio-direct-bytebuffer`	`true`	If `true`, specifies that the NIO direct `ByteBuffer` is used. In a limited resource environment, it might be faster to use non-direct Java's `ByteBuffer` by setting a value of `false`.
`authPassthroughEnabled`	`false`	If `true`, indicates that the "http-listener" on page 39 subelements receive traffic from an SSL-terminating proxy server, which is responsible for forwarding any information about the original client request (such as client IP address, SSL `keysize`, and authenticated client certificate chain) to the HTTP listeners using custom request headers. Each `http-listener` subelement can override this setting for itself.

TABLE 1–34 `http-service` Properties *(Continued)*

Property	Default	Description
`proxyHandler`	`com.sun.enterprise.web.ProxyHandlerImpl`	Specifies the fully qualified class name of a custom implementation of the `com.sun.appserv.ProxyHandler` abstract class, which allows a back-end application server instance to retrieve information about the original client request that was intercepted by an SSL-terminating proxy server (for example, a load balancer). An implementation of this abstract class inspects a given request for the custom request headers through which the proxy server communicates the information about the original client request to the Enterprise Server instance, and returns that information to its caller. The default implementation reads the client IP address from an HTTP request header named `Proxy-ip`, the SSL `keysize` from an HTTP request header named `Proxy-keysize`, and the SSL client certificate chain from an HTTP request header named `Proxy-auth-cert`. The `Proxy-auth-cert` value must contain the BASE-64 encoded client certificate chain without the `BEGIN CERTIFICATE` and `END CERTIFICATE` boundaries and with `\n` replaced with `% d% a`. Only used if `authPassthroughEnabled` is set to `true`. Each "http-listener" on page 39 subelement can override the `proxyHandler` setting for itself.
`proxiedProtocol`	none	Specifies a comma-separated list of protocols that can use the same port. Allowed values are `ws/tcp` (SOAP over TCP), `http`, `https` and `tls`. For example, if you set this property to `http,https` and the `port` is `4567`, you can access the port with either `http://`*host*`:4567/` or `https://`*host*`:4567/`. Specifying this property at the `http-service` level overrides settings at the "http-listener" on page 39 level. If this property is not set at either level, this feature is disabled.
`bufferSize`	`4096`	Specifies the size, in bytes, of the buffer to be provided for input streams created by HTTP listeners.
`connectionTimeout`	`30`	Specifies the number of seconds HTTP listeners wait, after accepting a connection, for the request URI line to be presented.
`maxKeepAliveRequests`	`250`	Specifies the maximum number of HTTP requests that can be pipelined until the connection is closed by the server. Set this property to `1` to disable HTTP/1.0 keep-alive, as well as HTTP/1.1 keep-alive and pipelining.
`traceEnabled`	`true`	If `true`, enables the TRACE operation. Set this property to `false` to make the Enterprise Server less susceptible to cross-site scripting attacks.
`accessLoggingEnabled`	`false`	If `true`, enables access logging for all "virtual-server" on page 92 subelements that do not specify this property. If `false`, disables access logging for all `virtual-server` subelements that do not specify this property.

TABLE 1–34 `http-service` Properties *(Continued)*

Property	Default	Description
`accessLogBufferSize`	`32768`	Specifies the size, in bytes, of the buffer where access log calls are stored. If the value is less than `5120`, a warning message is issued, and the value is set to `5120`.
`accessLogWriterInterval`	`300`	Specifies the number of seconds before the log is written to the disk. The access log is written when the buffer is full or when the interval expires. If the value is `0`, the buffer is always written even if it is not full. This means that each time the server is accessed, the log message is stored directly to the file.
`sso-enabled`	`false`	If `true`, single sign-on is enabled by default for all web applications on all virtual servers on this server instance that are configured for the same realm. If `false`, single sign-on is disabled by default for all virtual servers, and users must authenticate separately to every application on each virtual server. The `sso-enabled` property setting of the "virtual-server" on page 92 element overrides this setting for an individual virtual server. At the `http-service` level, you cannot change the `sso-max-inactive-seconds` and `sso-reap-interval-seconds` values from their defaults. However, you can change these values at the `virtual-server` level.
`disableUploadTimeout`	`true`	if `false`, the connection for a servlet that reads bytes slowly is closed after the `connectionUploadTimeout` is reached.
`connectionUploadTimeout`	`5`	Specifies the timeout for uploads. Applicable only if `disableUploadTimeout` is set to `false`.
`uriEncoding`	`UTF-8`	Specifies the character set used to decode the request URIs received on "http-listener" on page 39 subelements that do not define this property. Must be a valid IANA character set name.

I

iiop-listener

Defines an IIOP listen socket. To enable SSL for this listener, include an `ssl` subelement.

Note – This element is not implemented for GlassFish v3 Prelude.

Superelements

"iiop-service" on page 48

Subelements

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

TABLE 1–35 `iiop-listener` Subelements

Element	Required	Description
"ssl" on page 84	zero or one	Defines SSL parameters.
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–36 `iiop-listener` Attributes

Attribute	Default	Description
`id`	none	The listener name. An `iiop-listener` name cannot begin with a number.
`address`	none	IP address of the listener. Can be in dotted-pair or IPv6 notation, or just a name.
`port`	`1072`	(optional) Port number for the listener. Legal values are `1` - `65535`. On UNIX, creating sockets that listen on ports `1` - `1024` requires superuser privileges.
`security -enabled`	`false`	(optional) Determines whether the listener runs SSL. To turn SSL2 or SSL3 on or off and set ciphers, use an `ssl` element.
`enabled`	`true`	(optional) Determines whether the listener is active.

iiop-service

Defines the IIOP service.

Note – This element is not implemented for GlassFish v3 Prelude.

Superelements

"config" on page 27

Subelements

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

TABLE 1–37 `iiop-service` Subelements

Element	Required	Description
"orb" on page 71	only one	Configures the ORB.
"ssl-client-config" on page 85	zero or one	Defines SSL parameters for the ORB.
"iiop-listener" on page 47	zero or more	Defines an IIOP listen socket.

Attributes

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

TABLE 1–38 `iiop-service` Attributes

Attribute	Default	Description
`client-authentication-required`	`false`	(optional) If `true`, the server rejects unauthenticated requests and inserts an authentication-required bit in IORs sent to clients.

J

jacc-provider

Specifies a Java Authorization Contract for Containers (JACC) provider for pluggable authorization.

Superelements

"security-service" on page 79

Subelements

The following table describes subelements for the `jacc-provider` element.

TABLE 1–39 `jacc-provider` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `jacc-provider` element.

TABLE 1–40 `jacc-provider` Attributes

Attribute	Default	Description
`name`	`default`	Specifies the name of the JACC provider.
`policy-provider`	`com.sun.enterprise.security.provider.PolicyWrapper`	Corresponds to and can be overridden by the system property `javax.security.jacc.policy.provider`.
`policy-configuration-factory-provider`	`com.sun.enterprise.security.provider.PolicyConfigurationFactoryImpl`	Corresponds to and can be overridden by the system property `javax.security.jacc.PolicyConfigurationFactory.provider`.

java-config

Specifies the Virtual Machine for the Java platform (JVM software) configuration parameters.

Superelements

"config" on page 27

Subelements

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

TABLE 1–41 `java-config` Subelements

Element	Required	Description
"profiler" on page 72	zero or one	Configures a profiler for use with the Enterprise Server.
"jvm-options" on page 61	zero or more	Contains JVM command line options.
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–42 `java-config` Attributes

Attribute	Default	Description
`java-home`	`${com.sun.aas.javaRoot}`	The path to the directory where the JDK is installed.
`debug-enabled`	`false`	(optional) If `true`, the server starts up in debug mode ready for attachment with a JPDA-based debugger.

TABLE 1–42 `java-config` Attributes *(Continued)*

Attribute	Default	Description
`debug-options`	`-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n`	(optional) Specifies JPDA (Java Platform Debugger Architecture) options. A list of debugging options is available at http://java.sun.com/products/jpda/doc/conninv.html#Invocation. For more information about debugging, see the *Sun GlassFish Enterprise Server v3 Prelude Developer's Guide*.
`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 details about the `rmic` command, see http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/rmic.html.
`javac-options`	`-g`	(optional) Specifies options passed to the Java compiler at application deployment time.
`classpath-suffix`	none	Deprecated. Do not use.
`system-classpath`	none	Deprecated. Do not use.
`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 Enterprise Server installation relative path for its native shared libraries, the standard JRE native library path, the shell environment setting (`LD_LIBRARY_PATH` on UNIX), and any path specified in the `profiler` 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`	none	(optional) A comma separated list of class names, each of which must implement the `com.sun.appserv.BytecodePreprocessor` interface. Each of the specified preprocessor classes is called in the order specified.
`env-classpath-ignored`	`true`	Deprecated. Do not use.

jdbc-connection-pool

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

Superelements

"resources" on page 77

Subelements

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

TABLE 1–43 `jdbc-connection-pool` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `jdbc-connection-pool` element. Changing the following attributes requires a server restart: `datasource-classname`, `associate-with-thread`, `lazy-connection-association`, and `lazy-connection-enlistment`.

TABLE 1–44 `jdbc-connection-pool` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the connection pool. A "jdbc-resource" on page 56 element's `pool-name` attribute refers to this `name`.
`description`	none	(optional) Specifies a text description of this element.
`datasource-classname`	none	Specifies the class name of the associated vendor-supplied data source. This class must implement `javax.sql.DataSource`, `javax.sql.XADataSource` , `javax.sql.ConnectionPoolDatasource`, or a combination.
`res-type`	`javax.sql.DataSource`	(optional) Specifies the interface the data source class implements. The value of this attribute can be `javax.sql.DataSource`, `javax.sql.XADataSource` , or `javax.sql.ConnectionPoolDatasource`. 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.
`steady-pool-size`	`8`	(optional) Specifies the initial and minimum number of connections maintained in the pool.
`max-pool-size`	`32`	(optional) Specifies the maximum number of connections that can be created to satisfy client requests.
`max-wait-time-in-millis`	`60000`	(optional) Specifies the amount of time, in milliseconds, that the caller is willing to wait for a connection. If `0`, the caller is blocked indefinitely until a resource is available or an error occurs.

TABLE 1–44 `jdbc-connection-pool` Attributes *(Continued)*

Attribute	Default	Description
`pool-resize-quantity`	`2`	(optional) Specifies the number of idle connections to be destroyed if the existing number of connections is above the `steady-pool-size` (subject to the `max-pool-size` limit). This is enforced periodically at the `idle-timeout-in-seconds` interval. An idle connection is one that has not been used for a period of `idle-timeout-in-seconds`. When the pool size reaches `steady-pool-size`, connection removal stops.
`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. This timeout value must be kept shorter than the server side (database) timeout value to prevent the accumulation of unusable connections in the application.
`transaction-isolation-level`	default JDBC driver isolation level	(optional) Specifies the transaction isolation level on the pooled database connections. Allowed values are `read-uncommitted`, `read-committed` , `repeatable-read`, or `serializable`. Applications that change the isolation level on a pooled connection programmatically risk polluting the pool, which can lead to errors. See `is-isolation-level-guaranteed` for more details.
`is-isolation-level-guaranteed`	`true`	(optional) Applicable only when `transaction-isolation-level` is explicitly set. If `true`, every connection obtained from the pool is guaranteed to have the desired isolation level. This might impact performance on some JDBC drivers. Only set this attribute to `false` if you are certain that the hosted applications do not return connections with altered isolation levels.
`is-connection-validation-required`	`false`	(optional) Specifies whether connections have to be validated before being given to the application. If a resource's validation fails, it is destroyed, and a new resource is created and returned.
`connection-validation-method`	`auto-commit`	(optional) Legal values are as follows: ■ `auto-commit` (default), which uses `Connection.setAutoCommit(Connection.getAutoCommit())` ■ `meta-data`, which uses `Connection.getMetaData()` ■ `table`, which performs a query on a table specified in the `validation-table-name` attribute Because many JDBC drivers cache the results of `auto-commit` and `meta-data` calls, they do not always provide reliable validations. Check with the driver vendor to determine whether these calls are cached or not. The `table` must exist and be accessible, but it doesn't require any rows. Do not use an existing table that has a large number of rows or a table that is already frequently accessed. More details can be found at Connection Validation in GlassFish JDBC.
`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-method` is set to `table`.

TABLE 1–44 `jdbc-connection-pool` Attributes *(Continued)*

Attribute	Default	Description
`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`.
`non-transactional-connections`	`false`	(optional) If `true`, non-transactional connections can be made to the JDBC connection pool. These connections are not automatically enlisted with the transaction manager.
`allow-non-component-callers`	`false`	(optional) If `true`, non-Java-EE components, such as servlet filters, lifecycle modules, and third party persistence managers, can use this JDBC connection pool. The returned connection is automatically enlisted with the transaction context obtained from the transaction manager. Standard Java EE components can also use such pools. Connections obtained by non-component callers are not automatically closed at the end of a transaction by the container. They must be explicitly closed by the caller.
`connection-leak-timeout-in-seconds`	`0`	Detects potential connection leaks by the application. A connection that is not returned back to the pool by the application within the specified period is assumed to be potentially leaking, and a stack trace of the caller is logged. A zero value disables leak detection. A nonzero value enables leak tracing. Use this attribute along with `connection-leak-reclaim` to avoid potential connection leaks from the application. More details are at Connection Leak Tracing.
`connection-leak-reclaim`	`false`	If `true`, the pool will reclaim a connection after `connection-leak-timeout-in-seconds` occurs.
`connection-creation-retry-attempts`	`0`	Specifies the number of attempts to create a new connection in case of a failure.
`connection-creation-retry-interval-in-seconds`	`10`	Specifies the time interval between attempts to create a connection when `connection-creation-retry-attempts` is greater than `0`.
`validate-atmost-once-period-in-seconds`	`0`	Specifies the time interval within which a connection is validated at most once. Minimizes the number of validation calls.
`statement-timeout-in-seconds`	`-1`	Sets the query timeout property of a statement to enable termination of abnormally long running queries. The default value of `-1` disables this feature. An abnormally long running JDBC query executed by an application may leave it in a hanging state unless a timeout is explicitly set on the statement. This attribute guarantees that all queries automatically time out if not completed within the specified period. When statements are created, the `queryTimeout` is set according to the value specified in this attribute. This works only when the underlying JDBC driver supports `queryTimeout` for `Statement`, `PreparedStatement`, `CallableStatement`, and `ResultSet`.
`lazy-connection-enlistment`	`false`	If `true`, a connection is not enlisted in a transaction until it is used. If `false`, any connection object available to a transaction is enlisted in the transaction.
`lazy-connection-association`	`false`	If `true`, a physical connection is not associated with a logical connection until it is used. If `false`, a physical connection is associated with a logical connection even before it is used.

TABLE 1–44 `jdbc-connection-pool` Attributes *(Continued)*

Attribute	Default	Description
`associate-with-thread`	`false`	Specifies whether a connection is associated with the thread to enable the thread to reuse the connection. If `true`, allows a connection to be saved as a `ThreadLocal` in the calling thread. This connection gets reclaimed only when the calling thread dies or when the calling thread is not in use and the pool has run out of connections. If `false`, the thread must obtain a connection from the pool each time the thread requires a connection. This attribute associates a connection with a thread such that when the same thread is in need of a connection, it can reuse the connection already associated with that thread. In this case, the overhead of getting a connection from the pool is avoided. However, when this value is set to `true`, you should verify that the value of the `max-pool-size` attribute is comparable to the `max-thread-pool-size` attribute of the "thread-pool" on page 88 element. If the `max-thread-pool-size` value is much higher than the `max-pool-size` value, a lot of time is spent associating a connection with a new thread after dissociating it from an older one. Use this attribute in cases where the thread pool should reuse connections to avoid this overhead.
`match-connections`	`false`	Specifies whether a connection that is selected from the pool should be matched with the connections with certain credentials. If `true`, enables connection matching. You can set to `false` if connections are homogeneous. If the connection pool is used by applications that have multiple user credentials, `match-connections` must be `true`. The connection pool matches the request's credential with the connections in the pool and returns a matched connection for use. For new requests with different credentials, unmatched free connections are automatically purged to provide new connections to satisfy the new requests. This attribute need not be `true` if it is known that there is only one credential used by the applications and therefore the pool has homogeneous connections.
`max-connection-usage-count`	`0`	Specifies the number of times a connections is reused by the pool, after which it is closed. A zero value disables this feature. By limiting the maximum number of times a connection can be reused, you can avoid statement leaks if the application does not close statements.
`wrap-jdbc-objects`	`false`	If `true`, wrapped JDBC objects are returned for `Statement`, `PreparedStatement`, `CallableStatement`, `ResultSet`, and `DatabaseMetaData`. This option ensures that `Statement.getConnection()` is the same as `DataSource.getConnection()`. Therefore, this option should be `true` when both `Statement.getConnection()` and `DataSource.getConnection()` are done. The default is `false` to avoid breaking existing applications.

Properties

Most JDBC drivers allow use of standard property lists to specify the user, password, and other resource configuration information. Although properties are optional with respect to the Enterprise Server, some properties might be necessary for most databases. For details, see the JDBC 4.0 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 set*Name*(*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.

Changing JDBC driver properties requires a server restart.

TABLE 1–45 jdbc-connection-pool Properties

Property	Description
user	Specifies the user name for connecting to the database.
password	Specifies the password for connecting to the database.
databaseName	Specifies the database for this connection pool.
serverName	Specifies the database server for this connection pool.
port	Specifies the port on which the database server listens for requests.
networkProtocol	Specifies the communication protocol.
roleName	Specifies the initial SQL role name.
datasourceName	Specifies an underlying XADataSource, or a ConnectionPoolDataSource if connection pooling is done.
description	Specifies a text description.
url	Specifies the URL for this connection pool. Although this is not a standard property, it is commonly used.
LazyConnection Enlistment	Deprecated. Use the equivalent attribute.
LazyConnection Association	Deprecated. Use the equivalent attribute.
AssociateWithThread	Deprecated. Use the equivalent attribute.
MatchConnections	Deprecated. Use the equivalent attribute.

jdbc-resource

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

Superelements

"resources" on page 77

Subelements

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

TABLE 1–46 `jdbc-resource` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–47 `jdbc-resource` Attributes

Attribute	Default	Description
`jndi-name`	none	Specifies the JNDI name for the resource.
`description`	none	(optional) Specifies a text description of this element.
`pool-name`	none	Specifies the `name` of the associated "jdbc-connection-pool" on page 51.
`object-type`	`user`	(optional) Defines the type of the resource. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`enabled`	`true`	(optional) Determines whether this resource is enabled at runtime.

jms-host

Configures the host of the built-in Java Message Service (JMS) that is managed by the Enterprise Server.

Note – This element is not implemented for GlassFish v3 Prelude.

Superelements

"jms-service" on page 58

Subelements

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

TABLE 1–48 `jms-host` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–49 `jms-host` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the JMS host.
`host`	*machine-name*	(optional) Specifies the host name of the JMS host.
`port`	`7676`	(optional) Specifies the port number used by the JMS provider.
`admin-user-name`	`admin`	(optional) Specifies the administrator user name for the JMS provider.
`admin-password`	`admin`	(optional) Specifies the administrator password for the JMS provider.

jms-service

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

Note – This element is not implemented for GlassFish v3 Prelude.

Superelements

"config" on page 27

Subelements

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

TABLE 1–50 `jms-service` Subelements

Element	Required	Description
"jms-host" on page 57	zero or more	Specifies a host.
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–51 `jms-service` Attributes

Attribute	Default	Description
`init-timeout-in-seconds`	`60`	(optional) Specifies the amount of time the server instance waits at startup for its configured default JMS host to respond. If there is no response, startup is aborted. If set to `0`, the server instance waits indefinitely.
`type`	`EMBEDDED`	Specifies the type of JMS service: ■ `EMBEDDED` means the JMS provider is started in the same JVM machine as the Enterprise Server, and the networking stack is bypassed. Lazy initialization starts the default embedded broker on the first access of JMS services rather than at Enterprise Server startup. ■ `LOCAL` means the JMS provider is started along with the Enterprise Server. The `LOCAL` setting implicitly sets up a 1:1 relationship between an Enterprise Server instance and a Message Queue broker. ■ `REMOTE` means the JMS provider is remote and is not started by the Enterprise Server.
`start-args`	none	(optional) Specifies the string of arguments supplied for startup of the corresponding JMS instance.
`default-jms-host`	none	Specifies the name of the default "jms-host" on page 57. If `type` is set to `LOCAL`, this `jms-host` is automatically started at Enterprise Server startup.
`reconnect-interval-in-seconds`	`5`	(optional) Specifies the interval between reconnect attempts.
`reconnect-attempts`	`3`	(optional) Specifies the number of reconnect attempts.
`reconnect-enabled`	`true`	(optional) If `true`, reconnection is enabled. The JMS service 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 `onMessage()` method might not be able to complete successfully or the transaction might be rolled back due to a JMS exception. When the JMS service reestablishes the connection, JMS message redelivery semantics apply.
`addresslist-behavior`	`random`	(optional) Specifies whether the reconnection logic selects the broker from the `imqAddressList` in a `random` or sequential (`priority`) fashion.
`addresslist-iterations`	`3`	(optional) Specifies the number of times the reconnection logic iterates over the `imqAddressList` if `addresslist-behavior` is set to `PRIORITY`.
`mq-scheme`	`mq`	(optional) Specifies the scheme for establishing connection with the broker. For example, specify `http` for connecting to the broker over HTTP.
`mq-service`	`jms`	(optional) Specifies the type of broker service. If a broker supports SSL, the type of service can be `ssljms`.

Properties

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

TABLE 1–52 `jms-service` Properties

Property	Default	Description
`instance-name`	`imqbroker`	Specifies the full Sun GlassFish Message Queue broker instance name.
`instance-name-suffix`	none	Specifies a suffix to add to the full Message Queue broker instance name. The suffix is separated from the instance name by an underscore character (_). For example, if the instance name is `imqbroker`, appending the suffix `xyz` changes the instance name to `imqbroker_xyz`.
`append-version`	`false`	If `true`, appends the major and minor version numbers, preceded by underscore characters (_), to the full Message Queue broker instance name. For example, if the instance name is `imqbroker` , appending the version numbers changes the instance name to `imqbroker_8_0`.
`user-name`	`guest`	Specifies the user name for creating the JMS connection. Needed only if the default username/password of guest/guest is not available in the broker.
`password`	`guest`	Specifies the password for creating the JMS connection. Needed only if the default username/password of guest/guest is not available in the broker.

jmx-connector

Configures a JSR 160/255 compliant remote JMX connector, which handles the JMX communication between the domain administration server, the node agents, and the remote server instances. This JMX connector also handles JMX communication between an external management client and the domain administration server.

Only the system JMX connector is started by the server processes at startup. Do not configure additional JMX connectors.

Superelements

"admin-service" on page 19

Subelements

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

TABLE 1–53 `jmx-connector` Subelements

Element	Required	Description
"ssl" on page 84	zero or one	Defines SSL parameters.

TABLE 1–53 `jmx-connector` Subelements *(Continued)*

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–54 `jmx-connector` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the connector used by the designated system JMX connector for JMX communication between server instances. Do not modify this name.
`protocol`	`rmi_jrmp`	(optional) Specifies the protocol that this JMX connector supports. The only supported protocol is `rmi_jrmp`. Do not modify this value.
`address`	`0.0.0.0`	Specifies the IP address of the naming service where the JMX connector server stub is registered. This is not the port of the server socket that does the actual JMX communication. This is the address of the network interface where the RMI registry is started. If your system has multiple network interfaces, modify this value so that only a particular interface is selected.
`port`	`8686`	Specifies the port number on with the naming service (RMI registry) listens for RMI client connections. The only use of this naming service is to download the RMI stubs. If the default port is occupied, a free port is used. Legal values are `1` - `65535`. On UNIX, creating sockets that listen on ports `1` - `1024` requires superuser privileges.
`auth-realm-name`	`admin-realm`	Specifies the `name` of an "auth-realm" on page 24 subelement of the "security-service" on page 79 element for the server instance that is running this JMX connector's server end. Note that this is a dedicated administration security realm.
`security-enabled`	`false`	(optional) Determines whether JMX communication is encrypted.
`enabled`	`true`	(optional) Enables the JMX connector. Do not modify this value.

jvm-options

Contains JVM command line options, for example:

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

For information about JVM options, see
`http://java.sun.com/docs/hotspot/VMOptions.html`.

Superelements

"java-config" on page 50, "profiler" on page 72

Subelements

none - contains data

K

keep-alive

Configures keep-alive threads.

Superelements

"http-service" on page 44

Subelements

none

Attributes

The following table describes attributes for the `keep-alive` element.

TABLE 1–55 `keep-alive` Attributes

Attribute	Default	Description
`thread-count`	`1`	(optional) Specifies the number of keep-alive threads. The value must be 1 or greater.
`max-connections`	`250`	(optional) Specifies the maximum number of keep-alive connections. A value of `0` means requests are always rejected. A value of `-1` sets no limit to the number of keep-alive connections.
`timeout-in-seconds`	`30`	(optional) Specifies the maximum time for which a keep alive connection is kept open. A value of `0` or less means keep alive connections are kept open indefinitely.

L

log-service

Note – The `module-log-levels` element is not implemented for GlassFish v3 Prelude. To set log levels, set properties in the *domain-dir*`/config/logging.properties` file according to the instructions in the file.

Configures the *server log* file, which 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` element. The default name is `server.log`.

Other log files are configured by other elements:

- A *virtual server log* file stores messages from a `virtual-server` element that has an explicitly specified `log-file` attribute. See "virtual-server" on page 92.
- The *access log* file stores HTTP access messages from the default virtual server. The default name is `access.log`. See "access-log" on page 18 and "http-access-log" on page 37.
- The *transaction log* files store transaction messages from the default virtual server. The default name of the directory for these files is `tx`. See "transaction-service" on page 90.

Superelements

"config" on page 27

Subelements

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

TABLE 1–56 `log-service` Subelements

Element	Required	Description
"module-log-levels" on page 67	zero or one	Specifies log levels.
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–57 `log-service` Attributes

Attribute	Default	Description
`file`	`server.log` in the directory specified by the `log-root` attribute of the "domain" on page 30 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 the user account under which the server runs. An absolute path overrides the `log-root` attribute of the "domain" on page 30 element. A relative path is relative to the `log-root` attribute of the "domain" on page 30 element. If no `log-root` value is specified, it is relative to *domain-dir*`/config`.
`use-system-logging`	`false`	(optional) If `true`, uses the UNIX `syslog` service to produce and manage logs.
`log-handler`	none	(optional) Specifies a custom log handler to be added to end of the chain of system handlers to log to a different destination.
`log-filter`	none	(optional) Specifies a log filter to do custom filtering of log records.
`log-to-console`	`false`	(optional) Deprecated and ignored.
`log-rotation-limit-in-bytes`	`2000000`	(optional) Log files are rotated when the file size reaches the specified limit.
`log-rotation-timelimit-in-minutes`	`0`	(optional) Enables time-based log rotation. The valid range is `60` minutes (1 hour) to `14400` minutes (10*24*60 minutes or 10 days). If the value is zero, the files are rotated based on the size specified in `log-rotation-limit-in-bytes`. If the value is greater than zero, `log-rotation-timelimit-in-minutes` takes precedence over `log-rotation-limit-in-bytes`.
`retain-error-statistics-for-hours`	`5`	(optional) Specifies the number of most recent hours for which error statistics are retained in memory. The default and minimum value is `5` hours. The maximum value allowed is `500` hours. Larger values incur additional memory overhead.

M

manager-properties

Specifies session manager properties.

Superelements

"session-manager" on page 82

Subelements

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

TABLE 1–58 `manager-properties` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–59 `manager-properties` Attributes

Attribute	Default	Description
`session-file-name`	none; state is not preserved across restarts	(optional) Specifies the absolute or relative path to the directory in which the session state is preserved between application restarts, if preserving the state is possible. A relative path is relative to the temporary directory for this web application. To disable this behavior and not preserve the session state, specify an empty string as the value of this attribute.
`reap-interval-in-seconds`	`60`	(optional) Specifies the time between checks for expired sessions. If the `persistence-frequency` attribute of the `web-container-availability` element is set to `time-based`, active sessions are stored at this interval. Set this value lower than the frequency at which session data changes. For example, this value should be as low as possible (`1` second) for a hit counter servlet on a frequently accessed web site, or you could lose the last few hits each time you restart the server.
`max-sessions`	`-1`	(optional) Specifies the maximum number of sessions that can be in cache, or `-1` for no limit. After this, an attempt to create a new session causes an `IllegalStateException` to be thrown.
`session-id-generator-classname`	internal class generator	(optional) Not implemented.

mdb-container

Configures the message-driven bean (MDB) container.

Note – This element is not implemented for GlassFish v3 Prelude.

Superelements

"config" on page 27

Subelements

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

TABLE 1–60 `mdb-container` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–61 `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 removed when the `idle-timeout-in-seconds` timer expires. A cleaner thread removes any unused instances. Must be `0` or greater and less than `max-pool-size` . The pool is not resized below the `steady-pool-size`.
`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. A value of `0` means a bean can remain idle indefinitely.

Properties

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

TABLE 1–62 `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 `onMessage()` method when container-managed transactions are used. Deprecated.

message-security-config

Specifies configurations for message security providers.

Superelements

"security-service" on page 79

Subelements

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

TABLE 1–63 `message-security-config` Subelements

Element	Required	Description
"provider-config" on page 74	one or more	Specifies a configuration for one message security provider.

Attributes

The following table describes attributes for the `message-security-config` element.

TABLE 1–64 `message-security-config` Attributes

Attribute	Default	Description
`auth-layer`	`HttpServlet`	Specifies the message layer at which authentication is performed. The value must be `HttpServlet`.
`default-provider`	none	(optional) Specifies the server provider that is invoked for any application not bound to a specific server provider.
`default-client-provider`	none	(optional) Specifies the client provider that is invoked for any application not bound to a specific client provider.

module-log-levels

Note – The `module-log-levels` element is not implemented for GlassFish v3 Prelude. To set log levels, set properties in the *domain-dir*`/config/logging.properties` file according to the instructions in the file.

Controls the level of messages logged by server subsystems to the server log. Allowed values (levels) of each subsystem attribute are, from highest to lowest: `FINEST` , `FINER`, `FINE`, `CONFIG`, `INFO`, `WARNING`, `SEVERE`, and `OFF`. Each value logs all messages for all lower values. The default value is `INFO`, which logs all `INFO`, `SEVERE` , and `WARNING` messages.

Superelements

"log-service" on page 63

Subelements

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

TABLE 1–65 `module-log-levels` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `module-log-levels` element. The attribute names are the names of the Enterprise Server system loggers.

TABLE 1–66 `module-log-levels` Attributes

Attribute	Default	Description
`root`	`INFO`	(optional) Specifies the default level of messages logged by the entire Enterprise Server installation.
`server`	`INFO`	(optional) Specifies the default level of messages logged by the server instance.
`ejb-container`	`INFO`	(optional) Specifies the level of messages logged by the EJB container.
`cmp-container`	`INFO`	(optional) Specifies the level of messages logged by the CMP subsystem of the EJB container.
`mdb-container`	`INFO`	(optional) Specifies the level of messages logged by the MDB container.
`web-container`	`INFO`	(optional) Specifies the level of messages logged by the web container.
`classloader`	`INFO`	(optional) Specifies the level of messages logged by the classloader hierarchy.
`configuration`	`INFO`	(optional) Specifies the level of messages logged by the configuration subsystem.
`naming`	`INFO`	(optional) Specifies the level of messages logged by the naming subsystem.
`security`	`INFO`	(optional) Specifies the level of messages logged by the security subsystem.
`jts`	`INFO`	(optional) Specifies the level of messages logged by the Java Transaction Service.
`jta`	`INFO`	(optional) Specifies the level of messages logged by the Java Transaction API.
`admin`	`INFO`	(optional) Specifies the level of messages logged by the Administration Console subsystem.
`deployment`	`INFO`	(optional) Specifies the level of messages logged by the deployment subsystem.
`verifier`	`INFO`	(optional) Specifies the level of messages logged by the deployment descriptor verifier.
`jaxr`	`INFO`	(optional) Specifies the level of messages logged by the XML registry.
`jaxrpc`	`INFO`	(optional) Specifies the level of messages logged by the XML RPC module.
`saaj`	`INFO`	(optional) Specifies the level of messages logged by the SOAP with Attachments API for Java module.
`corba`	`INFO`	(optional) Specifies the level of messages logged by the ORB.
`javamail`	`INFO`	(optional) Specifies the level of messages logged by the JavaMail subsystem.

TABLE 1–66 `module-log-levels` Attributes *(Continued)*

Attribute	Default	Description
`jms`	`INFO`	(optional) Specifies the level of messages logged by the Java Message Service.
`connector`	`INFO`	(optional) Specifies the level of messages logged by the connector subsystem.
`jdo`	`INFO`	(optional) Specifies the level of messages logged by the Java Data Objects module.
`cmp`	`INFO`	(optional) Specifies the level of messages logged by the CMP subsystem.
`util`	`INFO`	(optional) Specifies the level of messages logged by the utility subsystem.
`resource-adapter`	`INFO`	(optional) Specifies the level of messages logged by the resource adapter subsystem.
`synchronization`	`INFO`	(optional) Specifies the level of messages logged by the synchronization subsystem.
`node-agent`	`INFO`	(optional) Specifies the level of messages logged by the node agent subsystem.
`self-management`	`INFO`	(optional) Specifies the level of messages logged by the self-management (management rules) subsystem.
`group-management-service`	`INFO`	(optional) Specifies the level of messages logged by the Group Management Service.
`management-event`	`INFO`	(optional) Specifies the level of messages logged by the self-management event subsystem.

module-monitoring-levels

Controls the level of monitoring of server subsystems. Allowed values of each subsystem attribute are `LOW`, `HIGH`, and `OFF`.

Superelements

"monitoring-service" on page 70

Subelements

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

TABLE 1–67 `module-monitoring-levels` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

TABLE 1–68 `module-monitoring-levels` Attributes

Attribute	Default	Description
`thread-pool`	`OFF`	(optional) Specifies the level of monitoring of the thread pool subsystem.
`orb`	`OFF`	(optional) Specifies the level of monitoring of the ORB.
`ejb-container`	`OFF`	(optional) Specifies the level of monitoring of the EJB container.
`web-container`	`OFF`	(optional) Specifies the level of monitoring of the web container.
`transaction-service`	`OFF`	(optional) Specifies the level of monitoring of the transaction service.
`http-service`	`OFF`	(optional) Specifies the level of monitoring of the HTTP service.
`jdbc-connection-pool`	`OFF`	(optional) Specifies the level of monitoring of the JDBC connection pool subsystem.
`connector-connection-pool`	`OFF`	(optional) Specifies the level of monitoring of the connector connection pool subsystem.
`connector-service`	`OFF`	(optional) Specifies the level of monitoring of the connector service.
`jms-service`	`OFF`	(optional) Specifies the level of monitoring of the JMS service.
`jvm`	`OFF`	(optional) Specifies the level of monitoring of the JVM subsystem.

monitoring-service

Configures the monitoring service.

Superelements

"config" on page 27

Subelements

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

TABLE 1–69 `monitoring-service` Subelements

Element	Required	Description
"module-monitoring-levels" on page 69	zero or one	Controls the level of monitoring of server subsystems.
"property" on page 73	zero or more	Specifies a property or a variable.

O

orb

Configures the ORB.

To enable SSL for outbound connections, include an "ssl-client-config" on page 85 subelement in the parent `iiop-service` element.

Note – This element is not implemented for GlassFish v3 Prelude.

Superelements

"iiop-service" on page 48

Subelements

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

TABLE 1–70 `orb` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–71 `orb` Attributes

Attribute	Default	Description
`use-thread-pool-ids`	none	Specifies a comma-separated list of `thread-pool-id` values defined in "thread-pool" on page 88 elements used by the ORB.
`message-fragment-size`	`1024`	(optional) GIOPv1.2 messages larger than this number of bytes are fragmented.
`max-connections`	`1024`	(optional) The maximum number of incoming connections on all IIOP listeners. Legal values are integers.

P

profiler

Configures a profiler for use with the Enterprise Server. For more information about profilers, see the *Sun GlassFish Enterprise Server v3 Prelude Developer's Guide*.

Superelements

"java-config" on page 50

Subelements

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

TABLE 1–72 `profiler` Subelements

Element	Required	Description
"jvm-options" on page 61	zero or more	Contains profiler-specific JVM command line options.
"property" on page 73	zero or more	Specifies a property or a variable.

Note – Subelements of a `profiler` element can occur in any order.

Attributes

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

TABLE 1–73 `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.

property

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

- Optional with respect to the Enterprise Server
- Needed by a system or object that the Enterprise 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.sun.enterprise.security.auth.realm.file.FileRealm">
     <property name="file" value="domain-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.

Superelements

"admin-service" on page 19, "application" on page 20, "audit-module" on page 23, "auth-realm" on page 24, "config" on page 27, "das-config" on page 29, "domain" on page 30, "ejb-container" on page 32, "ejb-timer-service" on page 34, "engine" on page 35, "http-listener" on page 39, "http-service" on page 44, "iiop-listener" on page 47, "jacc-provider" on page 49, "java-config" on page 50, "jdbc-connection-pool" on page 51, "jdbc-resource" on page 56, "jms-host" on page 57, "jms-service" on page 58, "jmx-connector" on page 60, "log-service" on page 63, "manager-properties" on page 64, "mdb-container" on page 65, "module-log-levels" on page 67, "module-monitoring-levels" on page 69, "monitoring-service" on page 70, "orb" on page 71, "profiler" on page 72, "provider-config" on page 74, "security-service" on page 79, "server" on page 80, "session-properties" on page 83, "store-properties" on page 86, "thread-pool" on page 88, "transaction-service" on page 90, "virtual-server" on page 92, "web-container" on page 98,

Subelements

none

Attributes

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

TABLE 1–74 `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`	none	(optional) Specifies a text description of this element.

provider-config

Specifies a configuration for one message security provider.

Although the `request-policy` and `response-policy` subelements are optional, the `provider-config` element does nothing if they are not specified.

Use property subelements to configure provider-specific properties. Property values are passed to the provider when its `initialize` method is called.

Superelements

"message-security-config" on page 66

Subelements

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

TABLE 1–75 `provider-config` Subelements

Element	Required	Description
"request-policy" on page 75	zero or one	Defines the authentication policy requirements of the authentication provider's request processing.
"response-policy" on page 78	zero or one	Defines the authentication policy requirements of the authentication provider's response processing.
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `provider-config` element.

TABLE 1–76 `provider-config` Attributes

Attribute	Default	Description
`provider-id`	none	Specifies a unique identifier for this `provider-config` element.
`provider-type`	none	Specifies whether the provider is a `client`, `server`, or `client-server` authentication provider.
`class-name`	none	Specifies the Java implementation class of the provider. Client authentication providers must implement the `com.sun.enterprise.security.jauth.ClientAuthModule` interface. Server authentication providers must implement the `com.sun.enterprise.security.jauth.ServerAuthModule` interface. Client-server providers must implement both interfaces.

Properties

The following table describes properties for the `provider-config` element.

TABLE 1–77 `provider-config` Properties

Property	Default	Description
`security.config`	*domain-dir*`/config/wss-server-config-1.0.xml`	Specifies the location of the message security configuration file. To point to a configuration file in the *domain-dir*`/config` directory, use the system property `${com.sun.aas.instanceRoot}/config/`, for example: `${com.sun.aas.instanceRoot}/config/wss-server-config-1.0.xml` See "system-property" on page 87.
`debug`	`false`	If `true`, enables dumping of server provider debug messages to the server log.
`dynamic.username.password`	`false`	If `true`, signals the provider runtime to collect the user name and password from the `CallbackHandler` for each request. If `false`, the user name and password for `wsse:UsernameToken(s)` is collected once, during module initialization. This property is only applicable for a `ClientAuthModule`.
`encryption.key.alias`	`s1as`	Specifies the encryption key used by the provider. The key is identified by its `keystore` alias.
`signature.key.alias`	`s1as`	Specifies the signature key used by the provider. The key is identified by its `keystore` alias.

R

request-policy

Defines the authentication policy requirements of the authentication provider's request processing.

Superelements

"provider-config" on page 74

Subelements

none

Attributes

The following table describes attributes for the `request-policy` element.

TABLE 1–78 `request-policy` Attributes

Attribute	Default	Description
`auth-source`	none	Specifies the type of required authentication, either `sender` (user name and password) or `content` (digital signature).
`auth-recipient`	none	Specifies whether recipient authentication occurs before or after content authentication. Allowed values are `before-content` and `after-content`.

request-processing

Configures request processing threads.

Superelements

"http-service" on page 44

Subelements

none

Attributes

The following table describes attributes for the `request-processing` element.

TABLE 1–79 `request-processing` Attributes

Attribute	Default	Description
`thread-count`	`20`	(optional) Specifies the maximum number of request processing threads.
`initial-thread-count`	`2`	(optional) Specifies the number of request processing threads that are available when the server starts up.

TABLE 1–79 `request-processing` Attributes *(Continued)*

Attribute	Default	Description
`thread-increment`	`1`	(optional) Specifies the number of request processing threads added when the number of requests exceeds the `initial-thread-count`.
`request-timeout-in-seconds`	`30`	(optional) Specifies the time at which the request times out.
`header-buffer-length-in-bytes`	`8192`	(optional) Specifies the size of the buffer used by the request processing threads to read the request data.

resource-ref

References a resource deployed to the server.

Superelements

"server" on page 80

Subelements

none

Attributes

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

TABLE 1–80 `resource-ref` Attributes

Attribute	Default	Description
`enabled`	`true`	(optional) Determines whether the resource is enabled.
`ref`	none	References the `name` attribute of a "jdbc-resource" on page 56 or "jdbc-connection-pool" on page 51 element.

resources

Contains configured resources, such as database connections.

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

Superelements

"domain" on page 30

Subelements

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

TABLE 1–81 `resources` Subelements

Element	Required	Description
"jdbc-resource" on page 56	zero or more	Defines a JDBC (Java Database Connectivity) resource.
"jdbc-connection-pool" on page 51	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.

response-policy

Defines the authentication policy requirements of the authentication provider's response processing.

Superelements

"provider-config" on page 74

Subelements

none

Attributes

The following table describes attributes for the `response-policy` element.

TABLE 1–82 `response-policy` Attributes

Attribute	Default	Description
`auth-source`	none	Specifies the type of required authentication, either `sender` (user name and password) or `content` (digital signature).
`auth-recipient`	none	Specifies whether recipient authentication occurs before or after content authentication. Allowed values are `before-content` and `after-content`.

S

security-service

Defines parameters and configuration information needed by the Java EE security service. For SSL configuration, see "ssl" on page 84.

Superelements

"config" on page 27

Subelements

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

TABLE 1–83 `security-service` Subelements

Element	Required	Description
"auth-realm" on page 24	one or more	Defines a realm for authentication.
"jacc-provider" on page 49	one or more	Specifies a Java Authorization Contract for Containers (JACC) provider for pluggable authorization.
"audit-module" on page 23	zero or more	Specifies an optional plug-in module that implements audit capabilities.
"message-security-config" on page 66	zero or more	Specifies configurations for message security providers.
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–84 `security-service` Attributes

Attribute	Default	Description
`default-realm`	`file`	(optional) Specifies the active authentication realm (an "auth-realm" on page 24 `name` attribute) for this server instance.
`default-principal`	none	(optional) Used as the identity of the default security context when necessary and when no principal is provided. This attribute need not be set for normal server operation.
`default-principal-password`	none	(optional) The password of the default principal. This attribute need not be set for normal server operation.
`anonymous-role`	`attribute is deprecated`	(optional) Deprecated. Do not use.
`audit-enabled`	`false`	(optional) If `true`, additional access logging is performed to provide audit information. Audit information consists of: ■ Authentication success and failure events ■ Servlet and EJB access grants and denials
`jacc`	`default`	(optional) Specifies the name of the "jacc-provider" on page 49 element to use for setting up the JACC infrastructure. Do not change the default value unless you are adding a custom JACC provider.
`audit-modules`	`default`	(optional) Specifies a space-separated list of audit provider modules used by the audit subsystem. The default value refers to the internal log-based audit module.
`activate-default-principal-to-role-mapping`	`false`	(optional) Applies a default principal for role mapping to any application that does not have an application-specific mapping defined. Every role is mapped to an instance of a `java.security.Principal` implementation class defined by `mapped-principal-class`. This class has the same name as the role.
`mapped-principal-class`	none	(optional) Customizes the `java.security.Principal` implementation class used when `activate-default-principal-to-role-mapping` is set to `true`.

server

Defines a server instance, which is a Java EE compliant container. One server instance is specially designated as a domain administration server (DAS). The "admin-service" on page 19 subelement of the "config" on page 27 element referenced by a server's `config-ref` attribute determines whether the server is the DAS.

Note – Server instances are not the same thing as virtual servers. Each server instance is a completely separate server that contains one or more virtual servers.

Note – In GlassFish v3 prelude, there is only one server, the DAS.

Superelements

"servers" on page 81

Subelements

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

TABLE 1–85 `server` Subelements

Element	Required	Description
"application-ref" on page 22	zero or more	References an application or module deployed to the server instance.
"resource-ref" on page 77	zero or more	References a resource deployed to the server instance.
"system-property" on page 87	zero or more	Specifies a system property.
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–86 `server` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the server instance.
`config-ref`	default "config" on page 27 element's `name`, `server-config`	(optional) References the `name` of the "config" on page 27 used by the server instance.

servers

Contains server instances.

Superelements

"domain" on page 30

Subelements

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

TABLE 1–87 `servers` Subelements

Element	Required	Description
"server" on page 80	only one	Defines a server instance.

session-config

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

Superelements

"web-container" on page 98

Subelements

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

TABLE 1–88 `session-config` Subelements

Element	Required	Description
"session-manager" on page 82	zero or one	Specifies session manager configuration information.
"session-properties" on page 83	zero or one	Specifies session properties.

session-manager

Specifies session manager information.

Note – The session manager interface is unstable. An unstable interface might be experimental or transitional, and hence might change incompatibly, be removed, or be replaced by a more stable interface in the next release.

Superelements

"session-config" on page 82

Subelements

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

TABLE 1–89 `session-manager` Subelements

Element	Required	Description
"manager-properties" on page 64	zero or one	Specifies session manager properties.
"store-properties" on page 86	zero or one	Specifies session persistence (storage) properties.

session-properties

Specifies session properties.

Superelements

"session-config" on page 82

Subelements

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

TABLE 1–90 `session-properties` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

TABLE 1–91 `session-properties` Attributes

Attribute	Default	Description
`timeout-in-seconds`	`600`	(optional) Specifies the default maximum inactive interval (in seconds) for all sessions created in this web module. If set to `0` or less, sessions in this web module never expire. If a `session-timeout` element is specified in the `web.xml` file, the `session-timeout` value overrides any `timeout-in-seconds` value. If neither `session-timeout` nor `timeout-in-seconds` is specified, the `timeout-in-seconds` default is used. Note that the `session-timeout` element in `web.xml` is specified in minutes, not seconds.

Properties

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

TABLE 1–92 `session-properties` Properties

Property	Default	Description
`enableCookies`	`true`	Uses cookies for session tracking if set to `true`.
`enableURLRewriting`	`true`	Enables URL rewriting. This provides session tracking via URL rewriting when the browser does not accept cookies. You must also use an `encodeURL` or `encodeRedirectURL` call in the servlet or JavaServer Pages™ (JSP™) page.
`idLengthBytes`	`128`	Specifies the number of bytes in this web module's session ID.

ssl

Defines SSL (Secure Socket Layer) parameters.

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

The grandparent "http-service" on page 44 element has properties that configure global SSL settings.

Superelements

"http-listener" on page 39, "iiop-listener" on page 47, "jmx-connector" on page 60, "ssl-client-config" on page 85

Subelements

none

Attributes

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

TABLE 1–93 `ssl` Attributes

Attribute	Default	Description
`cert-nickname`	`s1as`	The nickname of the server certificate in the certificate database or the PKCS#11 token. In the certificate, the name format is *tokenname*:*nickname*. Including the *tokenname*: part of the name in this attribute is optional.
`ssl2-enabled`	`false`	(optional) Determines whether SSL2 is enabled. If both SSL2 and SSL3 are enabled for a "virtual-server" on page 92, the server tries SSL3 encryption first. If that fails, the server tries SSL2 encryption.

TABLE 1–93 `ssl` Attributes *(Continued)*

Attribute	Default	Description
`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`.
`ssl3-enabled`	`true`	(optional) Determines whether SSL3 is enabled. The default is `true` . If both SSL2 and SSL3 are enabled for a "virtual-server" on page 92, 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 `+SSL_RSA_WITH_RC4_128_MD5` . Allowed values are `SSL_RSA_WITH_RC4_128_MD5`, `SSL_RSA_WITH_3DES_EDE_CBC_SHA`, `SSL_RSA_WITH_DES_CBC_SHA`, `SSL_RSA_EXPORT_WITH_RC4_40_MD5`, `SSL_RSA_WITH_NULL_MD5`, `SSL_RSA_WITH_RC4_128_SHA`, and `SSL_RSA_WITH_NULL_SHA`. Values available in previous releases are supported for backward compatibility.
`tls-enabled`	`true`	(optional) Determines whether TLS is enabled.
`tls-rollback-enabled`	`true`	(optional) Determines whether TLS rollback is enabled. TLS rollback should be enabled for Microsoft Internet Explorer 5.0 and 5.5. For more information, see the*Sun GlassFish Enterprise Server v3 Prelude Administration Guide*.
`client-auth-enabled`	`false`	(optional) Determines whether SSL3 client authentication is performed on every request, independent of ACL-based access control.

ssl-client-config

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

Note – This element is not implemented for GlassFish v3 Prelude.

Superelements

"iiop-service" on page 48

Subelements

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

TABLE 1–94 `ssl-client-config` Subelements

Element	Required	Description
"ssl" on page 84	only one	Defines SSL parameters.

store-properties

Specifies session persistence (storage) properties.

Superelements

"session-manager" on page 82

Subelements

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

TABLE 1–95 `store-properties` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

TABLE 1–96 `store-properties` Attributes

Attribute	Default	Description
`directory`	*domain-dir* `/generated/jsp/j2ee-apps/`*appname*`/`*appname*`_war`	(optional) Specifies the absolute or relative pathname of the directory into which individual session files are written. A relative path is relative to the temporary work directory for this web application. For GlassFish v3 Prelude, you can use this setting only for individual web applications. For details, see the *Sun GlassFish Enterprise Server v3 Prelude Application Deployment Guide*.
`reap-interval-in-seconds`	`60`	(optional) Not implemented. Use the `reap-interval-in-seconds` attribute of the "manager-properties" on page 64 element instead.

system-applications

Contains system applications. Do not delete or edit these applications.

Superelements

"domain" on page 30

Subelements

The following table describes subelements for the `system-applications` element.

TABLE 1–97 `system-applications` Subelements

Element	Required	Description
"application" on page 20	zero or more	Specifies an application.

system-property

Specifies a system property. A system property defines a common value for a setting at one of these levels, from highest to lowest: "domain" on page 30, "server" on page 80, or "config" on page 27. A value set at a higher level can be overridden at a lower level. Some system properties are predefined; see "system-property" on page 87. You can also create system properties using this element.

The following example shows the use of a predefined system property:

```
<log-service file="${com.sun.aas.instanceRoot}/logs/server.log">
    <module-log-levels admin=INFO .../>
</log-service>
```

The following example shows the creation and use of a system property:

```
<config name="config1">
    ...
    <http-service>
        ...
        <http-listener id="ls1" host="0.0.0.0" port="${ls1-port}"/>
        ...
    </http-service>
    ...
    <system-property name="ls1-port" value="8080"/>
</config>
```

Superelements

"config" on page 27, "domain" on page 30, "server" on page 80

Subelements

none

Attributes

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

TABLE 1–98 `system-property` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the system property.
`value`	none	Specifies the value of the system property.
`description`	none	(optional) Specifies a text description of this element.

Properties

The following table lists predefined system properties.

TABLE 1–99 Predefined System Properties

Property	Default	Description
`com.sun.aas.installRoot`	depends on operating system	Specifies the directory where the Enterprise Server is installed.
`com.sun.aas.instanceRoot`	depends on operating system	Specifies the top level directory for a server instance.
`com.sun.aas.hostName`	none	Specifies the name of the host (machine).
`com.sun.aas.javaRoot`	depends on operating system	Specifies the installation directory for the Java runtime.
`com.sun.aas.imqLib`	depends on operating system	Specifies the library directory for the Sun GlassFish Message Queue software.
`com.sun.aas.configName`	`server-config`	Specifies the name of the "config" on page 27 used by a server instance.
`com.sun.aas.instanceName`	`server1`	Specifies the name of the server instance. This property is not used in the default configuration, but can be used to customize configuration.
`com.sun.aas.domainName`	`domain1`	Specifies the name of the domain. This property is not used in the default configuration, but can be used to customize configuration.

T

thread-pool

Defines a thread pool.

Note – This element is not implemented for GlassFish v3 Prelude.

Superelements

"thread-pools" on page 89

Subelements

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

TABLE 1–100 `thread-pool` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

TABLE 1–101 `thread-pool` Attributes

Attribute	Default	Description
`thread-pool-id`	none	Specifies the thread pool ID.
`min-thread-pool-size`	`0`	(optional) Specifies the minimum number of threads in the pool. These are created when the thread pool is instantiated.
`max-thread-pool-size`	`200`	(optional) Specifies the maximum number of threads the pool can contain.
`idle-thread-timeout-in-seconds`	`120`	(optional) Specifies the amount of time after which idle threads are removed from the pool.
`num-work-queues`	`1`	(optional) Specifies the total number of work queues serviced by this thread pool.

thread-pools

Contains thread pools.

Note – This element is not implemented for GlassFish v3 Prelude.

Superelements

"config" on page 27

Subelements

The following table describes subelements for the `thread-pools` element.

TABLE 1–102 `thread-pools` Subelements

Element	Required	Description
"thread-pool" on page 88	one or more	Defines a thread pool.

transaction-service

Configures the Java Transaction Service (JTS).

Note – For GlassFish v3 Prelude, global (XA) transactions are not supported unless the optional JTS and Object Management Group (OMG) add-on components are downloaded from the Update Tool. Without the these components, only local transactions are supported.

Transaction recovery is not implemented for GlassFish v3 Prelude, even if the JTS and OMG add-on components are installed. Therefore, all transaction service attributes and properties pertaining to transaction recovery or transaction logs are not implemented.

Superelements

"config" on page 27

Subelements

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

TABLE 1–103 `transaction-service` Subelements

Element	Required	Description
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–104 `transaction-service` Attributes

Attribute	Default	Description
`automatic-recovery`	`false`	(optional) If `true`, the server instance attempts transaction recovery during startup.
`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.

TABLE 1–104 `transaction-service` Attributes *(Continued)*

Attribute	Default	Description
`tx-log-dir`	directory specified by the `log-root` attribute of the "domain" on page 30 element	(optional) Specifies the parent directory of the transaction log directory *instance-name*`/tx`. The directory in which the transaction logs are kept must be writable by the user account under which the server runs. A relative path is relative to the `log-root` attribute of the "domain" on page 30 element.
`heuristic-decision`	`rollback`	(optional) If the outcome of a distributed transaction cannot be determined because other participants are unreachable, this property determines the outcome. Allowed values are `rollback` and `commit`.
`retry-timeout-in-seconds`	`600`	(optional) Determines the retry time in the following scenarios: ■ At the transaction recovery time, if resources are unreachable. ■ If there are any transient exceptions in the second phase of a two phase commit protocol. A negative value specifies infinite retries. A value of `0` (zero) specifies no retries. A positive value indicates the time after which a retry is attempted.
`keypoint-interval`	`65536` (64 K)	(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 results in a larger transaction log file, but fewer keypoint operations and potentially better performance. A smaller value results in smaller log files, but slightly reduced performance due to the greater frequency of keypoint operations.

Properties

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

TABLE 1–105 `transaction-service` Properties

Property	Default	Description
`oracle-xa-recovery-workaround`	`true`	If `true`, the Oracle XA Resource workaround is used in transaction recovery.
`disable-distributed-transaction-logging`	`false`	If `true`, disables transaction logging, which might improve performance. 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.
`pending-txn-cleanup-interval`	none if this property is absent, `60` if this property is present but has no value	Specifies the interval, in seconds, at which an asynchronous thread checks for pending transactions and completes them.

TABLE 1–105 `transaction-service` Properties *(Continued)*

Property	Default	Description
`use-last-agent-optimization`	`true`	If `true`, enables last agent optimization, which improves the throughput of transactions. If one non-XA resource is used with XA resources in the same transaction, the non XA resource is the last agent.
`wait-time-before-recovery-insec`	`60`	Specifies the wait time, in seconds, after which an instance starts the recovery for a dead instance.
`db-logging-resource`	none	Specifies the JNDI name of the JDBC resource for the database to which transactions are logged. For more information, see Chapter 10, "Using the Transaction Service," in *Sun GlassFish Enterprise Server v3 Prelude Developer's Guide*.

V

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 can 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 the Enterprise Server is first installed, a default virtual server is created. (You can also assign a default virtual server to each new "http-listener" on page 39 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 Enterprise 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.

Superelements

"http-service" on page 44

Subelements

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

TABLE 1–106 `virtual-server` Subelements

Element	Required	Description
"http-access-log" on page 37	zero or one	Defines an access log file.
"property" on page 73	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–107 `virtual-server` Attributes

Attribute	Default	Description
`id`	none	Virtual server ID. This is a unique ID that allows lookup of a specific virtual server. A virtual server ID cannot begin with a number.
`http-listeners`	none	(optional) In a comma-separated list, references `id` attributes of "http-listener" on page 39 elements that specify the connection(s) the virtual server uses. Required only for a `virtual-server` that is not referenced by the `default-virtual-server` attribute of an `http-listener`.
`default-web-module`	none	(optional) References the `name` attribute of the default web "application" on page 20 for this virtual server, which responds to requests that cannot be resolved to other web modules deployed to this virtual server (see the "application-ref" on page 22 element).
`hosts`	`${com.sun.aas.hostName}`	A comma-separated list of values, each of which selects the current virtual server when included in the `Host` request header. Two or more `virtual-server` elements that reference or are referenced by the same `http-listener` cannot have any `hosts` values in common.
`state`	`on`	(optional) Determines whether a `virtual-server` is active (`on`) or inactive (`off`, `disabled`). The default is `on` (active). When inactive, a `virtual-server` does not service requests. If a `virtual-server` is `disabled`, only the global server administrator can turn it `on`.
`docroot`	*domain-dir*`/docroot`	(optional) Specifies the document root for this virtual server.
`log-file`	`server.log` in the directory specified by the `log-root` attribute of the "domain" on page 30 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 the user account under which the server runs. See the "log-service" on page 63 description for details about logs.

Properties

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

TABLE 1–108 `virtual-server` Properties

Property	Default	Description
`sso-enabled`	`false`	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. This setting overrides the `sso-enabled` property setting of the "http-service" on page 44 element.
`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.
`ssoCookieSecure`	`dynamic`	Sets the `Secure` attribute of any `JSESSIONIDSSO` cookies associated with the web applications deployed to this virtual server. Applicable only if the `sso-enabled` property is set to `true`. Allowed values are as follows: ■ `true` — Sets `Secure` to `true`. ■ `false` — Sets `Secure` to `false`. ■ `dynamic` — The `JSESSIONIDSSO` cookie inherits the `Secure` setting of the first session participating in SSO. To set the `Secure` attribute of a `JSESSIONID` cookie, use the `cookieSecure` `cookie-properties` property in the `sun-web.xml` file. For details, see "cookie-properties" in *Sun GlassFish Enterprise Server v3 Prelude Application Deployment Guide*.
`setCacheControl`	none	Specifies a comma-separated list of `Cache-Control` response directives. For a list of valid directives, see section 14.9 of the document at http://www.ietf.org/rfc/rfc2616.txt.
`accessLogging Enabled`	`false`	If `true`, enables access logging for this virtual server only. If `false`, disables access logging for this virtual server only.
`accessLogBuffer Size`	`32768`	Specifies the size, in bytes, of the buffer where access log calls are stored. If the value is less than `5120`, a warning message is issued, and the value is set to `5120`. To set this property for all virtual servers, set it as a property of the parent "http-service" on page 44 element.

TABLE 1–108 `virtual-server` Properties *(Continued)*

Property	Default	Description
`accessLogWriter Interval`	`300`	Specifies the number of seconds before the log is written to the disk. The access log is written when the buffer is full or when the interval expires. If the value is `0`, the buffer is always written even if it is not full. This means that each time the server is accessed, the log message is stored directly to the file. To set this property for all virtual servers, set it as a property of the parent "http-service" on page 44 element.
`allowRemoteAddress`	none	Specifies a comma-separated list of regular expression patterns that the remote client's IP address is compared to. If this property is specified, the remote address *must* match for this request to be accepted. If this property is not specified, all requests are accepted *unless* the remote address matches a `denyRemoteAddress` pattern.
`denyRemoteAddress`	none	Specifies a comma-separated list of regular expression patterns that the remote client's IP address is compared to. If this property is specified, the remote address must *not* match for this request to be accepted. If this property is not specified, request acceptance is governed solely by the `allowRemoteAddress` property.
`allowRemoteHost`	none	Specifies a comma-separated list of regular expression patterns that the remote client's hostname (as returned by `[java.net.]Socket.getInetAddress().getHostName()`) is compared to. If this property is specified, the remote hostname *must* match for this request to be accepted. If this property is not specified, all requests are accepted *unless* the remote hostname matches a `denyRemoteHost` pattern.
`denyRemoteHost`	none	Specifies a comma-separated list of regular expression patterns that the remote client's hostname (as returned by `[java.net.]Socket.getInetAddress().getHostName()`) is compared to. If this property is specified, the remote hostname must *not* match for this request to be accepted. If this property is not specified, request acceptance is governed solely by the `allowRemoteHost` property.
`authRealm`	none	Specifies the `name` attribute of an "auth-realm" on page 24 element, which overrides the server instance's default realm for stand-alone web applications deployed to this virtual server. A realm defined in a stand-alone web application's `web.xml` file overrides the virtual server's realm.
`securePages WithPragma`	`true`	Set this property to `false` to ensure that for all web applications on this virtual server file downloads using SSL work properly in Internet Explorer. You can set this property for a specific web application. For details, see "sun-web-app" in *Sun GlassFish Enterprise Server v3 Prelude Application Deployment Guide*.

TABLE 1–108 `virtual-server` Properties *(Continued)*

Property	Default	Description

`alternatedocroot_n`	none	Specifies an alternate document root (docroot), where *n* is a positive integer that allows specification of more than one. Alternate docroots allow web applications to serve requests for certain resources from outside their own docroot, based on whether those requests match one (or more) of the URI patterns of the web application's alternate docroots.
		
		If a request matches an alternate docroot's URI pattern, it is mapped to the alternate docroot by appending the request URI (minus the web application's context root) to the alternate docroot's physical location (directory). If a request matches multiple URI patterns, the alternate docroot is determined according to the following precedence order:
		■ Exact match
		■ Longest path match
		■ Extension match
		
		For example, the following properties specify three alternate docroots. The URI pattern of the first alternate docroot uses an exact match, whereas the URI patterns of the second and third alternate docroots use extension and longest path prefix matches, respectively.
		
		`<property name="alternatedocroot_1"`
		`value="from=/my.jpg dir=/srv/images/jpg"/>`
		`<property name="alternatedocroot_2"`
		`value="from=*.jpg dir=/srv/images/jpg"/>`
		`<property name="alternatedocroot_3"`
		`value="from=/jpg/* dir=/src/images"/>`
		
		The `value` of each alternate docroot has two components: The first component, `from`, specifies the alternate docroot's URI pattern, and the second component, `dir`, specifies the alternate docroot's physical location (directory). Spaces are allowed in the `dir` component.
		
		You can set this property for a specific web application. For details, see "sun-web-app" in *Sun GlassFish Enterprise Server v3 Prelude Application Deployment Guide*.

`contextXmlDefault`	none	Specifies the location, relative to *domain-dir*, of the `context.xml` file for this virtual server, if one is used. For more information about the `context.xml` file, see "Using a context.xml File" in *Sun GlassFish Enterprise Server v3 Prelude Developer's Guide* and The Context Container (`http://tomcat.apache.org/tomcat-5.5-doc/config/context.html`). Context parameters, environment entries, and resource definitions in `context.xml` are supported in the Enterprise Server.
`allowLinking`	`false`	If `true`, resources that are symbolic links in web applications on this virtual server are served. The value of this property in the `sun-web.xml` file takes precedence if defined. For details, see "sun-web-app" in *Sun GlassFish Enterprise Server v3 Prelude Application Deployment Guide*. **Caution –** Setting this property to `true` on Windows systems exposes JSP source code.

TABLE 1–108 `virtual-server` Properties *(Continued)*

Property	Default	Description

`send-error_n`	none	Specifies custom error page mappings for the virtual server, which are inherited by all web applications deployed on the virtual server. A web application can override these custom error page mappings in its `web.xml` deployment descriptor. The value of each `send-error_n` property has three components, which may be specified in any order:
		
		The first component, `code`, specifies the three-digit HTTP response status code for which the custom error page should be returned in the response.
		
		The second component, `path`, specifies the absolute or relative file system path of the custom error page. A relative file system path is interpreted as relative to the *domain-dir*`/config` directory.
		
		The third component, `reason`, is optional and specifies the text of the reason string (such as `Unauthorized` or `Forbidden`) to be returned.
		
		For example:
		
		`<property name="send-error_1"`
		`value="code=401 path=/myhost/401.html reason=MY-401-REASON"/>`
		
		This example property definition causes the contents of `/myhost/401.html` to be returned with 401 responses, along with this response line:
		
		`HTTP/1.1 401 MY-401-REASON`

`redirect_n`	none	Specifies that a request for an old URL is treated as a request for a new URL. These properties are inherited by all web applications deployed on the virtual server. The value of each `redirect_n` property has two components, which may be specified in any order:
		
		The first component, `from`, specifies the prefix of the requested URI to match.
		
		The second component, `url-prefix`, specifies the new URL prefix to return to the client. The from prefix is simply replaced by this URL prefix.
		
		For example:
		
		`<property name="redirect_1"`
		`value="from=/dummy url-prefix=http://etude"/>`

`valve_n`	none	Specifies a fully qualified class name of a custom valve, where *n* is a positive integer that allows specification of more than one. The valve class must implement the `org.apache.catalina.Valve` interface from Tomcat or previous Enterprise Server releases, or the `org.glassfish.web.valve.GlassFishValve` interface from the current Enterprise Server release. For example:
		
		`<property name="valve_1"`
		`value="org.glassfish.extension.Valve"/>`
		
		You can set this property for a specific web application. For details, see "sun-web-app" in *Sun GlassFish Enterprise Server v3 Prelude Application Deployment Guide*.

TABLE 1–108 `virtual-server` Properties *(Continued)*

Property	Default	Description

`listener_n`	none	Specifies a fully qualified class name of a custom Catalina listener, where *n* is a positive integer that allows specification of more than one. The listener class must implement the `org.apache.catalina.ContainerListener` or `org.apache.catalina.LifecycleListener` interface. For example:
		`<property name="listener_1"`
		`value="org.glassfish.extension.MyLifecycleListener"/>`
		You can set this property for a specific web application. For details, see "sun-web-app" in *Sun GlassFish Enterprise Server v3 Prelude Application Deployment Guide*.

W

web-container

Configures the web container.

Superelements

"config" on page 27

Subelements

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

TABLE 1–109 `web-container` Subelements

Element	Required	Description
"session-config" on page 82	zero or one	Specifies session configuration information for the web container.
"property" on page 73	zero or more	Specifies a property or a variable.

Properties

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

TABLE 1–110 `web-container` Properties

Property	Default	Description
`dispatcher-max-depth`	`20`	Prevents recursive `include` or `forward` statements from creating an infinite loop by setting a maximum nested dispatch level. If this level is exceeded, the following message is written to the server log: `Exceeded maximum depth for nested request dispatches`

Index

D

K

L

M

N

O

P

Q

R

S

T

U

V

W

X