Sun GlassFish Enterprise Server v3 Prelude Application Deployment Guide

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

Part No: 820–4502–10
October 2008

081028@21288

Contents

Preface

This *Application Deployment Guide* describes deployment of applications and application components to the Sun GlassFish™ Enterprise Server, and includes information about deployment descriptors.

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 820-4502.

◆ ◆ ◆ CHAPTER 1

Assembling and Deploying Applications

This chapter describes Sun Java™ System Enterprise Server modules and how these modules are assembled. This chapter also describes tools for assembly and deployment.

The Enterprise Server modules include Java Platform, Enterprise Edition (Java EE platform) standard features and Enterprise Server specific features. Only Enterprise Server specific features are described in detail in this chapter.

The following topics are presented in this chapter:

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. The `sun-ejb-jar.xml` elements related to these features are ignored.

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, and the `sun-web.xml` and `sun-ejb-jar.xml` elements related to web services are ignored.

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

Overview of Assembly and Deployment

Application assembly (also known as packaging) is the process of combining discrete components into a single unit that can be deployed to a Java-EE-compliant application server. This section covers the following topics:

About Modules

A module is a collection of one or more components that execute in the same container type (for example, web or EJB) with annotations or deployment descriptors of that type. In the Java EE modules, one descriptor is Java EE standard, the other is optional and Enterprise Server specific. Annotations can be used instead of Java EE standard descriptors.

Types of modules are as follows:

- **Web Application Archive (WAR)** — A web application is a collection of servlets, HTML pages, classes, and other resources that can be bundled and deployed to several Java EE application servers. A WAR file can consist of the following items: servlets, JavaServer Pages™ (JSP™) files, JSP tag libraries, utility classes, static pages, client-side applets, beans, bean classes, and annotations or deployment descriptors (`web.xml` and `sun-web.xml`).
- **EJB JAR File** — The EJB JAR file is the standard format for assembling enterprise beans. This file contains the bean classes (home, remote, local, and implementation), all of the utility classes, and annotations or deployment descriptors (`ejb-jar.xml` and `sun-ejb-jar.xml`).

Package definitions must be used in the source code of all modules so the class loader can properly locate the classes after the modules have been deployed.

Because the information in a deployment descriptor is declarative, it can be changed without requiring modifications to source code. At run time, the Java EE server reads this information and acts accordingly.

EJB JAR and Web modules can be deployed separately, outside of any application. EJB components are assembled in a JAR file with annotations or `ejb-jar.xml` and `sun-ejb-jar.xml` deployment descriptors. Web components are assembled in a WAR file with annotations or `web.xml` and `sun-web.xml` deployment descriptors.

Java EE Standard Annotation

The Enterprise Server supports modules annotated according to the following specifications:

- JSR 250 Common Annotation Specification (`http://www.jcp.org/en/jsr/detail?id=250`)
- JSR 181 Annotation for Web Services Specification (`http://www.jcp.org/en/jsr/detail?id=181`)
- EJB 3.0 Specification (`http://www.jcp.org/en/jsr/detail?id=220`)

The following annotation and deployment descriptor combinations are supported:

- Java EE modules can be packaged with full Java EE 5.0 compliant standard and runtime deployment descriptors. If the standard deployment descriptors have specified the attribute `metadata-complete`, annotations in the module are ignored.
- Java EE modules can be fully annotated with metadata defined by the listed specifications. Annotation eliminates the need for Java EE standard deployment descriptors. In most cases, the Enterprise Server deployment descriptors are also optional.
- Java EE modules can be partially annotated with some deployment information in standard deployment descriptors. In case of conflicts, deployment descriptor values supersede the annotated metadata, but a warning message is logged.

Java EE Standard Descriptors

Java EE standard deployment descriptors are described in the Java EE specification, v5. You can find the specification at `http://java.sun.com/products/`. Information about the XML schemas that define Java EE standard deployment descriptors is available at `http://java.sun.com/xml/ns/javaee/`.

Enterprise Server Descriptors

The Enterprise Server uses additional, optional deployment descriptors for configuring features specific to the Enterprise Server.

For complete descriptions of these files, see Appendix A, “Deployment Descriptor Files.”

Naming Standards

Names of applications and individually deployed EJB JAR and WAR modules must be unique within an Enterprise Server domain. Modules of the same type within an application must have unique names.

If you do not explicitly specify a name, the default name is the first portion of the file name (without the `.war` or `.jar` extension). Modules of different types can have the same name within an application, because the directories holding the individual modules are named with `_jar` and `_war` suffixes. This is the case when you use the Administration Console or the `asadmin` command. If you use the `asadmin deploy` command to deploy a directory, your directory structure must follow this same convention. See "Directory Deployment" on page 25 and "Tools for Deployment" on page 26.

You can specify a name in one of these ways:

- If deploying using the Administration Console, you can specify a name in the Application Name field.
- If deploying using the `asadmin deploy` command, you can override the default name by specifying the `--name` option.

Make sure your package and file names do not contain spaces or characters that are illegal for your operating system.

Using a Java package-like naming scheme is recommended for module filenames, module names as found in the `<module-name>` portion of the `ejb-jar.xml` files, and EJB names as found in the `<ejb-name>` portion of the `ejb-jar.xml` files. The use of this package-like naming scheme ensures that name collisions do not occur. The benefits of this naming practice apply not only to the Enterprise Server, but to other Java EE application servers as well.

JNDI lookup names for EJB components must also be unique.

Directory Structure

When you deploy a module, the module is expanded from the WAR or JAR file to an open directory structure. The directories are named with `_jar` and `_war` suffixes. If you use the `asadmin deploy` command to deploy a directory, your directory structure must follow this same convention; see "Directory Deployment" on page 25. Module directory structures follow the structure outlined in the Java EE specification.

Assembling Modules and Applications

Assembling (or packaging) modules and applications in Enterprise Server conforms to all of the customary Java-EE-defined specifications. The only difference is that when you assemble in Enterprise Server, you can include optional Enterprise Server specific deployment descriptors that enhance the functionality of the Enterprise Server.

For example, when you assemble an EJB JAR module, you annotate or create two deployment descriptor files with these names: `ejb-jar.xml` and `sun-ejb-jar.xml`. For more information about `sun-ejb-jar.xml`, see Appendix A, "Deployment Descriptor Files."

The Enterprise Server provides these tools for assembling and verifying a module or an application:

- "The NetBeans IDE" on page 21

The NetBeans IDE

You can use the NetBeans™ Integrated Development Environment (IDE), or another IDE, to assemble Java EE applications and modules. The GlassFish edition of the Enterprise Server is bundled with the NetBeans 6.1 IDE. For more information about using the NetBeans IDE, see `http://www.netbeans.org`.

Deploying Modules and Applications

This section describes the different ways to deploy Java EE modules to the Enterprise Server. It covers the following topics:

- "Deployment Errors" on page 21
- "The Deployment Life Cycle" on page 21
- "Deployment for Development" on page 22
- "Tools for Deployment" on page 26
- "Deploying a Web Service" on page 27
- "Deploying a WAR Module" on page 28
- "Deploying an EJB JAR Module" on page 29
- "Access to Shared Frameworks" on page 29

Deployment Errors

If an error occurs during deployment, the module is not deployed. This prevents a partial deployment that could leave the server in an inconsistent state.

In addition, certain warning conditions allow a module to be deployed but return a warning message to the deployment client.

The Deployment Life Cycle

After installing the Enterprise Server and starting a domain, you can deploy (install) Java EE modules. During deployment and as the application is changed, a module can go through the following stages:

1. Initial Deployment

   Before deploying a module, start the domain.

Because modules are packaged in archive files, specify the archive file name during deployment.

Deployment is *dynamic*: you don't need to restart the server after deploying a module. If you do restart, all deployed modules are still deployed and available.

2. Enabling or Disabling

   By default, a deployed module is enabled, which means that it is runnable. To prevent access, disable the application or module. A disabled module is not uninstalled from the domain and can be easily enabled after deployment. For more information, see "Disabling a Deployed Application or Module" on page 23.

3. Redeployment

   To replace a deployed module, redeploy it. Redeploying automatically undeploys the previously deployed module and replaces it with the new one.

4. Undeployment

   To uninstall a module, undeploy it.

Deployment for Development

This section covers the following topics related to deployment for development:

Note – You can overwrite a previously deployed module by using the `asadmin redeploy` command, the `--force` option of `asadmin deploy`, or the Redeploy button in the Administration Console. However, you must remove a preconfigured resource before you can update it.

Dynamic Deployment

You can deploy, redeploy, and undeploy a module without restarting the server. This is called dynamic deployment. Although primarily for developers, dynamic deployment can be used in operational environments to bring new applications and modules online without requiring a server restart.

Whenever a redeployment is done, the sessions at that transit time become invalid unless you use the `keepSessions=true` property of the `asadmin redeploy` command. For example:

```
asadmin redeploy --properties keepSessions=true --name hello.war
```

For details, see the *Sun GlassFish Enterprise Server v3 Prelude Reference Manual*.

Keep Sessions is also a checkbox option when you redeploy using the Administration Console. For details, click the Help button in the Administration Console.

Disabling a Deployed Application or Module

You can disable a deployed module without removing it from the server. Disabling a module makes it inaccessible to clients.

To disable a module using the `asadmin disable` command, see the *Sun GlassFish Enterprise Server v3 Prelude Reference Manual*.

▼ To Disable a Module in the Administration Console

1. **Open the Applications component.**
2. **Go to the page for the type of module.**

   For example, for a web application, go to the Web Applications page.
3. **Click on the box to the left of the name of each module you wish to disable.**
4. **Click on the Disable button.**

See Also For details, click the Help button in the Administration Console.

Dynamic Reloading

If dynamic reloading is enabled (it is by default), you do not have to redeploy a module when you change its code or deployment descriptors. Simply copy the changed class files or descriptors into the deployment directory for the module. Then update the timestamp of a file named `.reload` as described in "To Reload Code or Deployment Descriptor Changes" on page 24. The server checks for changes periodically and redeploys the module, automatically and dynamically, with the changes. Deployment directories are as follows:

domain-dir`/applications/`*module-name*

Note – Deployment directories may change between Enterprise Server releases.

Dynamic reloading is useful in a development environment, because it allows code changes to be tested quickly. In a production environment, however, dynamic reloading might degrade performance. In addition, whenever a reload is done, the sessions at that transit time become invalid. The client must restart the session. JSP file changes don't need to be dynamically reloaded. Modified JSP files are automatically recompiled and reloaded when they are accessed.

▼ To Enable Dynamic Reloading in the Administration Console

1. **Select the Application Server component.**
2. **Select the Advanced tab.**
3. **Select the Applications Configuration tab.**
4. **Check the Reload Enabled box to enable dynamic reloading.**
5. **Enter a number of seconds in the Reload Poll Interval field.**

   This sets the interval at which modules are checked for code changes and dynamically reloaded. The default is 2.

See Also For details, click the Help button in the Administration Console.

▼ To Reload Code or Deployment Descriptor Changes

1. **Create an empty file named** `.reload` **at the root of the deployed module.**

   domain-dir`/applications/`*module-name*`/.reload`

   Note – Deployment directories may change between Enterprise Server releases.

   ---

2. **Explicitly update the** `.reload` **file's timestamp (**`touch .reload` **in UNIX) each time you make changes.**

Automatic Deployment

Automatic deployment, also called *autodeployment*, involves copying a module file (JAR, WAR) into a special directory, where it is automatically deployed by the Enterprise Server. To undeploy an automatically deployed module, simply remove its file from the special autodeployment directory. This is useful in a development environment, because it allows new code to be tested quickly. Autodeployment is enabled by default.

Autodeployment of a web services JSR 181 annotated file is supported. For more information, see JSR 181 (`http://www.jcp.org/en/jsr/detail?id=181`) and Chapter 5, "Developing Web Services," in *Sun GlassFish Enterprise Server v3 Prelude Developer's Guide*.

Autodeployment is enabled by default.

▼ To Enable and Configure or to Disable Autodeployment

1. **Select the Application Server component.**
2. **Select the Advanced tab.**
3. **Select the Applications Configuration tab.**
4. **Check the Auto Deploy Enabled box to enable autodeployment, or uncheck this box to disable autodeployment.**
5. **Enter a number of seconds in the Auto Deploy Poll Interval field.**

   This sets the interval at which modules are checked for code changes and dynamically reloaded. The default is 2.
6. **You can change the Auto Deploy Directory.**

   You can enter an absolute or relative path. A relative path is relative to *domain-dir*. The default is *domain-dir*/`autodeploy`.
7. **Check the Precompile Enabled box to precompile any JSP files.**

See Also For details, click the Help button in the Administration Console.

Directory Deployment

A directory containing an unpackaged module is sometimes called an exploded directory. To deploy a directory instead of an EAR or WAR, you can do one of the following:

- Use the Administration Console as described in "The Administration Console Deployment Pages" on page 26 and enter a path to an exploded directory instead of a path to an archive file.
- Use the `asadmin deploy` command and specify a path to an exploded directory instead of a path to an archive file. See the *Sun GlassFish Enterprise Server v3 Prelude Reference Manual*.

The contents of the directory must match the contents of a corresponding Java EE archive file. For example, if you deploy a Web application from a directory, the contents of the directory must be the same as a corresponding WAR file. In addition, the directories holding the modules must be named with `_jar` and `_war` suffixes. For information about the required directory contents, see the appropriate specifications.

The directory must be accessible to the machine on which the Enterprise Server runs.

You can change the deployment descriptor files directly in the exploded directory. If your environment is configured to use dynamic reloading, you can also dynamically reload modules deployed from the directory. For more information, see "Dynamic Reloading" on page 23.

Note – On Windows, if you are deploying a directory on a mapped drive, you must be running the Enterprise Server as the same user to which the mapped drive is assigned, or the Enterprise Server won't see the directory.

Tools for Deployment

This section discusses the various tools that can be used to deploy modules and applications. The deployment tools include:

- "The `asadmin` Deployment Commands" on page 26
- "The Administration Console Deployment Pages" on page 26

The `asadmin` Deployment Commands

You can use the following `asadmin` commands to deploy or undeploy web or EJB modules on local servers.

- `deploy` — Deploys a web or EJB module. If the module is already deployed, you can force redeployment by setting the `--force` option to `true`. You can also deploy a module in an expanded directory structure. See "Directory Deployment" on page 25.
- `redeploy` — Redeploys a web or EJB module that is already deployed. Whenever a redeployment is done, the sessions at that transit time become invalid unless you use the `keepSessions=true` property of the `asadmin redeploy` command.
- `undeploy` — Undeploys a web or EJB module.
- `disable` — Immediately disables a web or EJB module. Disabling a web module makes it inaccessible to clients.
- `enable` — Immediately enables a web or EJB module.
- `list-components` — Lists all deployed web or EJB modules.
- `deploydir` — Deploys a web or EJB module from an expanded directory structure. Deprecated; use `asadmin deploy` instead.

For details, see the *Sun GlassFish Enterprise Server v3 Prelude Reference Manual*.

The Administration Console Deployment Pages

You can use the Administration Console to deploy modules to both local and remote Enterprise Server sites.

▼ To Use the Administration Console for Deployment

1. **Open the Applications component.**
2. **Go to the page for the type of application or module.**

   For example, for a web application, go to the Web Applications page. You can undeploy, enable, or disable an application or module from the table on this page.
3. **Click on the Deploy button.**

   On this page, you type the path to the WAR, JAR, or the exploded directory structure. You can also specify deployment settings that vary according to the type of module.

See Also For details, click the Help button in the Administration Console.

For more information on deploying from an exploded directory structure, see "Directory Deployment" on page 25.

Deploying a Web Service

Note – 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, and the `sun-web.xml` and `sun-ejb-jar.xml` elements related to web services are ignored.

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

You deploy a web service endpoint to the Enterprise Server just as you would any servlet or stateless session bean (SLSB).

Web service management is fully supported in the Administration Console. If the deployed module has a web service endpoint, it is detected automatically during deployment. Once the module is deployed, click on the Web Service component. The table in the right frame lists deployed web service endpoints.

You can use the `--registryjndiname` option of the `asadmin deploy` command to publish the web service as part of deployment, but this is optional. See "Tools for Deployment" on page 26.

To deploy a JSR 181 annotated file, use the autodeployment feature. You can compile and deploy in one step, as in the following example:

```
javac -cp javaee.jar -d domain-dir/autodeploy MyWS.java
```

For more information about JSR 181, see `http://jcp.org/en/jsr/detail?id=181`. For more information about autodeployment, see "Automatic Deployment" on page 24.

The Sun-specific deployment descriptor files `sun-web.xml` and `sun-ejb-jar.xml` provide optional web service enhancements in their "webservice-endpoint" on page 134 and "webservice-description" on page 133 elements.

For more information about web services, see JSR 181 (`http://www.jcp.org/en/jsr/detail?id=181`) and Chapter 5, "Developing Web Services," in *Sun GlassFish Enterprise Server v3 Prelude Developer's Guide*.

Deploying a WAR Module

You deploy a WAR module as described in "Tools for Deployment" on page 26. If you do not specify a context root, the default is the name of the WAR file without the extension.

If a web application accesses a `DataSource` that is not specified in a `resource-ref` in `sun-web.xml`, or there is no `sun-web.xml` file, the `resource-ref-name` defined in `web.xml` is used. A warning message is logged recording the JNDI name used to look up the resource.

You can precompile JSP files during deployment by checking the appropriate box in the Administration Console, or by using the `--precompilejsp` option of the `asadmin deploy` command.

You can keep the generated source for JSP files by adding the `keepgenerated` flag to the `jsp-config` element in `sun-web.xml`. For example:

```
<sun-web-app>
   ...
   <jsp-config>
      <property name=keepgenerated value=true />
   </jsp-config>
</sun-web-app>
```

If you include this property when you deploy the WAR module, the generated source is kept in *domain-dir*`/generated/jsp/`*module-name*.

For more information about JSP precompilation, see "jsp-config" on page 76.

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 virtual servers, see "virtual-server" in *Sun GlassFish Enterprise Server v3 Prelude Administration Reference*.

Whenever a redeployment is done, the sessions at that transit time become invalid unless you use the `keepSessions=true` property of the `asadmin redeploy` command. For example:

```
asadmin redeploy --properties keepSessions=true --name hello.war
```

For details, see the *Sun GlassFish Enterprise Server v3 Prelude Reference Manual*.

Keep Sessions is also a checkbox option when you redeploy using the Administration Console. For details, click the Help button in the Administration Console.

Web module context roots must be unique within the server.

Deploying an EJB JAR Module

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. The `sun-ejb-jar.xml` elements related to these features are ignored.

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

You deploy an EJB JAR module as described in "Tools for Deployment" on page 26.

If no JNDI name for the EJB JAR module is specified in the `jndi-name` element immediately under the `ejb` element in `sun-ejb-jar.xml`, or there is no `sun-ejb-jar.xml` file, a default, non-clashing JNDI name is derived. A warning message is logged recording the JNDI name used to look up the EJB JAR module.

Access to Shared Frameworks

When Java EE applications and modules use shared framework classes (such as utility classes and libraries) the classes can be put in the path for the Common Classloader or an application-specific class loader rather than in an application or module. If you assemble a large, shared library into every module that uses it, the result is a huge file that takes too long to register with the server. In addition, several versions of the same class could exist in different classloaders, which is a waste of resources. For more information, see Chapter 2, "Class Loaders," in *Sun GlassFish Enterprise Server v3 Prelude Developer's Guide*.

◆ ◆ ◆ APPENDIX A

Deployment Descriptor Files

This chapter describes deployment descriptor files specific to the Sun GlassFish Enterprise Server in the following 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. The `sun-ejb-jar.xml` elements related to these features are ignored.

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, and the `sun-web.xml` and `sun-ejb-jar.xml` elements related to web services are ignored.

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.

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

Note – GlassFish v3 Prelude does not support clustering or load balancing features, Sun GlassFish Message Queue software, container-managed persistence (CMP), or descriptor-based schema generation. Elements related to these features are ignored.

Sun Java System Application Server Descriptors

Sun GlassFish Enterprise Server uses optional deployment descriptors in addition to the Java EE standard descriptors for configuring features specific to the Enterprise Server.

Note – Settings in the Enterprise Server deployment descriptors override corresponding settings in the Java EE deployment descriptors and in the Enterprise Server's `domain.xml` file unless otherwise stated. For more information about the `domain.xml` file, see the *Sun GlassFish Enterprise Server v3 Prelude Administration Reference*.

Each deployment descriptor (or XML) file has a corresponding DTD file, which defines the elements, data, and attributes that the deployment descriptor file can contain. The DTD files for the Enterprise Server deployment descriptors are located in the *as-install*`/lib/dtds` directory.

Note – Do not edit the DTD files; their contents change only with new versions of the Enterprise Server.

For general information about DTD files and XML, see the XML specification at `http://www.w3.org/TR/REC-xml`.

The following table lists the Enterprise Server deployment descriptors and their DTD files.

TABLE A–1 Sun GlassFish Enterprise Server Descriptors

Deployment Descriptor	DTD File	Description
`sun-web.xml`	`sun-web-app_2_5-0.dtd`	Configures a web application (WAR file).
`sun-ejb-jar.xml`	`sun-ejb-jar_3_0-0.dtd`	Configures an enterprise bean (EJB JAR file).

Note – The Enterprise Server deployment descriptors must be readable and writable by the file owners.

In each deployment descriptor file, subelements must be defined in the order in which they are listed under each **Subelements** heading, unless otherwise noted.

The sun-web.xml File

The element hierarchy in the `sun-web.xml` file is as follows:

```
sun-web-app
.  context-root
.  security-role-mapping
.  .  role-name
.  .  principal-name
.  .  group-name
.  servlet
.  .  servlet-name
.  .  principal-name
.  .  webservice-endpoint
.  .  .  port-component-name
.  .  .  endpoint-address-uri
.  .  .  login-config
.  .  .  .  auth-method
.  .  .  message-security-binding
.  .  .  .  message-security
.  .  .  .  .  message
.  .  .  .  .  .  java-method
.  .  .  .  .  .  .  method-name
.  .  .  .  .  .  .  method-params
.  .  .  .  .  .  .  .  method-param
.  .  .  .  .  .  operation-name
.  .  .  .  .  request-protection
.  .  .  .  .  response-protection
.  .  .  transport-guarantee
.  .  .  service-qname
.  .  .  tie-class
.  .  .  servlet-impl-class
.  .  .  debugging-enabled
.  .  .  property (with attributes)
.  .  .  .  description
.  idempotent-url-pattern
.  session-config
.  .  session-manager
.  .  .  manager-properties
```

```
. . . . property (with attributes)
. . . . . description
. . . store-properties
. . . . property (with attributes)
. . . . . description
. . session-properties
. . . property (with attributes)
. . . . description
. . cookie-properties
. . . property (with attributes)
. . . . description
. ejb-ref
. . ejb-ref-name
. . jndi-name
. resource-ref
. . res-ref-name
. . jndi-name
. . default-resource-principal
. . . name
. . . password
. resource-env-ref
. . resource-env-ref-name
. . jndi-name
. service-ref
. . service-ref-name
. . port-info
. . . service-endpoint-interface
. . . wsdl-port
. . . . namespaceURI
. . . . localpart
. . . stub-property
. . . . name
. . . . value
. . . call-property
. . . . name
. . . . value
. . . message-security-binding
. . . . message-security
. . . . . message
. . . . . . java-method
. . . . . . . method-name
. . . . . . . method-params
. . . . . . . . method-param
. . . . . . operation-name
. . . . . request-protection
. . . . . response-protection
. . call-property
. . . name
```

```
. . . value
. . wsdl-override
. . service-impl-class
. . service-qname
. . . namespaceURI
. . . localpart
. message-destination-ref
. . message-destination-ref-name
. . jndi-name
. cache
. . cache-helper
. . . property (with attributes)
. . . . description
. . default-helper
. . . property (with attributes)
. . . . description
. . property (with attributes)
. . . description
. . cache-mapping
. . . servlet-name
. . . url-pattern
. . . cache-helper-ref
. . . dispatcher
. . . timeout
. . . refresh-field
. . . http-method
. . . key-field
. . . constraint-field
. . . . constraint-field-value
. class-loader
. . property (with attributes)
. . . description
. jsp-config
. locale-charset-info
. . locale-charset-map
. . parameter-encoding
. parameter-encoding
. property (with attributes)
. . description
. message-destination
. . message-destination-name
. . jndi-name
. webservice-description
. . webservice-description-name
. . wsdl-publish-location
```

Here is a sample `sun-web.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sun-web-app PUBLIC '-//Sun Microsystems, Inc.//DTD
Application Server 9.0 Servlet 2.5//EN'
'http://www.sun.com/software/appserver/dtds/sun-web-app_2_5-0.dtd'>
<sun-web-app>
   <session-config>
      <session-manager/>
   </session-config>
   <resource-ref>
      <res-ref-name>mail/Session</res-ref-name>
      <jndi-name>mail/Session</jndi-name>
   </resource-ref>
   <jsp-config/>
</sun-web-app>
```

The sun-ejb-jar.xml File

The element hierarchy in the `sun-ejb-jar.xml` file is as follows:

```
sun-ejb-jar
.  security-role-mapping
.  .  role-name
.  .  principal-name
.  .  group-name
.  enterprise-beans
.  .  name
.  .  unique-id
.  .  ejb
.  .  .  ejb-name
.  .  .  jndi-name
.  .  .  ejb-ref
.  .  .  .  ejb-ref-name
.  .  .  .  jndi-name
.  .  .  resource-ref
.  .  .  .  res-ref-name
.  .  .  .  jndi-name
.  .  .  .  default-resource-principal
.  .  .  .  .  name
.  .  .  .  .  password
.  .  .  resource-env-ref
.  .  .  .  resource-env-ref-name
.  .  .  .  jndi-name
.  .  .  service-ref
.  .  .  .  service-ref-name
.  .  .  .  port-info
.  .  .  .  .  service-endpoint-interface
```

```
. . . . . wsdl-port
. . . . . . namespaceURI
. . . . . . localpart
. . . . . stub-property
. . . . . . name
. . . . . . value
. . . . . call-property
. . . . . . name
. . . . . . value
. . . . . message-security-binding
. . . . . . message-security
. . . . . . . message
. . . . . . . . java-method
. . . . . . . . . method-name
. . . . . . . . . method-params
. . . . . . . . . . method-param
. . . . . . . . operation-name
. . . . . . . . request-protection
. . . . . . . . response-protection
. . . . call-property
. . . . . name
. . . . . value
. . . . wsdl-override
. . . . service-impl-class
. . . . service-qname
. . . . . namespaceURI
. . . . . localpart
. . . message-destination-ref
. . . . message-destination-ref-name
. . . . jndi-name
. . . pass-by-reference
. . . cmp
. . . . mapping-properties
. . . . is-one-one-cmp
. . . . one-one-finders
. . . . . finder
. . . . . . method-name
. . . . . . query-params
. . . . . . query-filter
. . . . . . query-variables
. . . . . . query-ordering
. . . . prefetch-disabled
. . . . . query-method
. . . . . . method-name
. . . . . . method-params
. . . . . . . method-param
. . . principal
. . . . name
```

```
. . . mdb-connection-factory
. . . . jndi-name
. . . . default-resource-principal
. . . . . name
. . . . . password
. . . jms-durable-subscription-name
. . . jms-max-messages-load
. . . ior-security-config
. . . . transport-config
. . . . . integrity
. . . . . confidentiality
. . . . . establish-trust-in-target
. . . . . establish-trust-in-client
. . . . as-context
. . . . . auth-method
. . . . . realm
. . . . . required
. . . . sas-context
. . . . . caller-propagation
. . . is-read-only-bean
. . . refresh-period-in-seconds
. . . commit-option
. . . cmt-timeout-in-seconds
. . . use-thread-pool-id
. . . gen-classes
. . . . remote-impl
. . . . local-impl
. . . . remote-home-impl
. . . . local-home-impl
. . . bean-pool
. . . . steady-pool-size
. . . . resize-quantity
. . . . max-pool-size
. . . . pool-idle-timeout-in-seconds
. . . . max-wait-time-in-millis
. . . bean-cache
. . . . max-cache-size
. . . . resize-quantity
. . . . is-cache-overflow-allowed
. . . . cache-idle-timeout-in-seconds
. . . . removal-timeout-in-seconds
. . . . victim-selection-policy
. . . mdb-resource-adapter
. . . . resource-adapter-mid
. . . . activation-config
. . . . . description
. . . . . activation-config-property
. . . . . . activation-config-property-name
```

```
. . . . . . activation-config-property-value
. . . webservice-endpoint
. . . . port-component-name
. . . . endpoint-address-uri
. . . . login-config
. . . . . auth-method
. . . . . realm
. . . . message-security-binding
. . . . . message-security
. . . . . . message
. . . . . . . java-method
. . . . . . . . method-name
. . . . . . . . method-params
. . . . . . . . . method-param
. . . . . . . operation-name
. . . . . . request-protection
. . . . . . response-protection
. . . . transport-guarantee
. . . . service-qname
. . . . tie-class
. . . . servlet-impl-class
. . . . debugging-enabled
. . . . property (with subelements)
. . . . . name
. . . . . value
. . . flush-at-end-of-method
. . . . method
. . . . . description
. . . . . ejb-name
. . . . . method-name
. . . . . method-intf
. . . . . method-params
. . . . . . method-param
. . . checkpointed-methods
. . . checkpoint-at-end-of-method
. . . . method
. . . . . description
. . . . . ejb-name
. . . . . method-name
. . . . . method-intf
. . . . . method-params
. . . . . . method-param
. . pm-descriptors
. . cmp-resource
. . . jndi-name
. . . default-resource-principal
. . . . name
. . . . password
```

```
.  .  .  property (with subelements)
.  .  .  .  name
.  .  .  .  value
.  .  .  create-tables-at-deploy
.  .  .  drop-tables-at-undeploy
.  .  .  database-vendor-name
.  .  .  schema-generator-properties
.  .  .  .  property (with subelements)
.  .  .  .  .  name
.  .  .  .  .  value
.  .  message-destination
.  .  .  message-destination-name
.  .  .  jndi-name
.  .  webservice-description
.  .  .  webservice-description-name
.  .  .  wsdl-publish-location
```

Note – If any configuration information for an enterprise bean is not specified in the `sun-ejb-jar.xml` file, it defaults to a corresponding setting in the EJB container if an equivalency exists.

Here is a sample `sun-ejb-jar.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sun-ejb-jar PUBLIC '-//Sun Microsystems, Inc.//DTD
Application Server 9.0 EJB 3.0//EN'
'http://www.sun.com/software/appserver/dtds/sun-ejb-jar_3_0-0.dtd'>
<sun-ejb-jar>
<display-name>First Module</display-name>
<enterprise-beans>
    <ejb>
        <ejb-name>CustomerEJB</ejb-name>
        <jndi-name>customer</jndi-name>
        <bean-pool>
            <steady-pool-size>10</steady-pool-size>
            <resize-quantity>10</resize-quantity>
            <max-pool-size>100</max-pool-size>
            <pool-idle-timeout-in-seconds>600</pool-idle-timeout-in-seconds>
        </bean-pool>
        <bean-cache>
            <max-cache-size>100</max-cache-size>
            <resize-quantity>10</resize-quantity>
            <removal-timeout-in-seconds>3600</removal-timeout-in-seconds>
            <victim-selection-policy>LRU</victim-selection-policy>
        </bean-cache>
    </ejb>
```

```
    <cmp-resource>
        <jndi-name>jdbc/__default</jndi-name>
        <create-tables-at-deploy>true</create-tables-at-deploy>
        <drop-tables-at-undeploy>true</drop-tables-at-undeploy>
    </cmp-resource>
</enterprise-beans>
</sun-ejb-jar>
```

Alphabetical Listing of All Elements

A

activation-config

Specifies an activation configuration, which includes the runtime configuration properties of the message-driven bean in its operational environment. For example, this can include information about the name of a physical JMS destination. Matches and overrides the `activation-config` element in the `ejb-jar.xml` file.

Superelements

"mdb-resource-adapter" on page 88 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–2 `activation-config` subelements

Element	Required	Description
"description" on page 61	zero or one	Specifies a text description of the activation configuration.
"activation-config-property" on page 42	one or more	Specifies an activation configuration property.

activation-config-property

Specifies the name and value of an activation configuration property.

Superelements

"activation-config" on page 41 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–3 `activation-config-property` subelements

Element	Required	Description
"activation-config-property-name" on page 42	only one	Specifies the name of an activation configuration property.
"activation-config-property-value" on page 43	only one	Specifies the value of an activation configuration property.

activation-config-property-name

Specifies the name of an activation configuration property.

Superelements

"activation-config-property" on page 42 (`sun-ejb-jar.xml`)

Subelements

none - contains data

activation-config-property-value

Specifies the value of an activation configuration property.

Superelements

"activation-config-property" on page 42 (`sun-ejb-jar.xml`)

Subelements

none - contains data

as-context

Specifies the authentication mechanism used to authenticate the client.

Superelements

"ior-security-config" on page 73 (`sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `as-context` element.

TABLE A–4 `as-context` Subelements

Element	Required	Description
"auth-method" on page 43	only one	Specifies the authentication method. The only supported value is `USERNAME_PASSWORD`.
"realm" on page 105	only one	Specifies the realm in which the user is authenticated.
"required" on page 108	only one	Specifies whether the authentication method specified in the `auth-method` element must be used for client authentication.

auth-method

Specifies the authentication method.

If the parent element is "as-context" on page 43, the only supported value is `USERNAME_PASSWORD`.

If the parent element is "login-config" on page 84, specifies the authentication mechanism for the web service endpoint. As a prerequisite to gaining access to any web resources protected by an authorization constraint, a user must be authenticated using the configured mechanism.

Superelements

"login-config" on page 84 (`sun-web.xml`), "as-context" on page 43 (`sun-ejb-jar.xml`)

Subelements

none - contains data

B

bean-cache

Specifies the entity bean cache properties. Used for entity beans and stateful session beans.

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

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `bean-cache` element.

TABLE A–5 `bean-cache` Subelements

Element	Required	Description
"max-cache-size" on page 86	zero or one	Specifies the maximum number of beans allowable in cache.
"is-cache-overflow-allowed" on page 74	zero or one	Deprecated.
"cache-idle-timeout-in-seconds" on page 49	zero or one	Specifies the maximum time that a stateful session bean or entity bean is allowed to be idle in cache before being passivated. Default value is 10 minutes (600 seconds).
"removal-timeout-in-seconds" on page 106	zero or one	Specifies the amount of time a bean remains before being removed. If `removal-timeout-in-seconds` is less than `idle-timeout`, the bean is removed without being passivated.
"resize-quantity" on page 109	zero or one	Specifies the number of beans to be created if the pool is empty (subject to the `max-pool-size` limit). Values are from 0 to MAX_INTEGER.

TABLE A–5 `bean-cache` Subelements *(Continued)*

Element	Required	Description
"victim-selection-policy" on page 132	zero or one	Specifies the algorithm that must be used by the container to pick victims. Applies only to stateful session beans.

Example

```
<bean-cache>
   <max-cache-size>100</max-cache-size>
   <cache-resize-quantity>10</cache-resize-quantity>
   <removal-timeout-in-seconds>3600</removal-timeout-in-seconds>
   <victim-selection-policy>LRU</victim-selection-policy>
      <cache-idle-timeout-in-seconds>600</cache-idle-timeout-in-seconds>
   <removal-timeout-in-seconds>5400</removal-timeout-in-seconds>
</bean-cache>
```

bean-pool

Specifies the pool properties of stateless session beans, entity beans, and message-driven bean.

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

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–6 `bean-pool` Subelements

Element	Required	Description
"steady-pool-size" on page 120	zero or one	Specifies the initial and minimum number of beans maintained in the pool. Default is 32.
"resize-quantity" on page 109	zero or one	Specifies the number of beans to be created if the pool is empty (subject to the `max-pool-size` limit). Values are from 0 to MAX_INTEGER.
"max-pool-size" on page 87	zero or one	Specifies the maximum number of beans in the pool. Values are from 0 to MAX_INTEGER. Default is to the EJB container value or 60.
"max-wait-time-in-millis" on page 87	zero or one	Deprecated.

TABLE A–6 `bean-pool` Subelements *(Continued)*

Element	Required	Description
"pool-idle-timeout-in-seconds" on page 98	zero or one	Specifies the maximum time that a bean is allowed to be idle in the pool. After this time, the bean is removed. This is a hint to the server. Default time is 600 seconds (10 minutes).

Example

```
<bean-pool>
   <steady-pool-size>10</steady-pool-size>
   <resize-quantity>10</resize-quantity>
   <max-pool-size>100</max-pool-size>
   <pool-idle-timeout-in-seconds>600</pool-idle-timeout-in-seconds>
</bean-pool>
```

C

cache

Configures caching for web application components.

Superelements

"sun-web-app" on page 124 (`sun-web.xml`)

Subelements

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

TABLE A–7 `cache` Subelements

Element	Required	Description
"cache-helper" on page 48	zero or more	Specifies a custom class that implements the `CacheHelper` interface.
"default-helper" on page 60	zero or one	Allows you to change the properties of the default, built-in "cache-helper" on page 48 class.
"property (with attributes)" on page 101	zero or more	Specifies a cache property, which has a name and a value.
"cache-mapping" on page 49	zero or more	Maps a URL pattern or a servlet name to its cacheability constraints.

Attributes

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

TABLE A–8 cache Attributes

Attribute	Default	Description
max-entries	4096	(optional) Specifies the maximum number of entries the cache can contain. Must be a positive integer.
timeout-in-seconds	30	(optional) Specifies the maximum amount of time in seconds that an entry can remain in the cache after it is created or refreshed. Can be overridden by a "timeout" on page 129 element.
enabled	true	(optional) Determines whether servlet and JSP caching is enabled.

Properties

The following table describes properties for the cache element.

TABLE A–9 cache Properties

Property	Default	Description
cacheClassName	com.sun.appserv.web.cache.LruCache	Specifies the fully qualified name of the class that implements the cache functionality. See "Cache Class Names" on page 47 for possible values.
MultiLRUSegmentSize	4096	Specifies the number of entries in a segment of the cache table that should have its own LRU (least recently used) list. Applicable only if cacheClassName is set to com.sun.appserv.web.cache.MultiLruCache.
MaxSize	unlimited; Long.MAX_VALUE	Specifies an upper bound on the cache memory size in bytes (KB or MB units). Example values are 32 KB or 2 MB. Applicable only if cacheClassName is set to com.sun.appserv.web.cache.BoundedMultiLruCache.

Cache Class Names

The following table lists possible values of the cacheClassName property.

TABLE A–10 cacheClassName Values

Value	Description
com.sun.appserv.web.cache.LruCache	A bounded cache with an LRU (least recently used) cache replacement policy.
com.sun.appserv.web.cache.BaseCache	An unbounded cache suitable if the maximum number of entries is known.
com.sun.appserv.web.cache.MultiLruCache	A cache suitable for a large number of entries (>4096). Uses the MultiLRUSegmentSize property.

TABLE A–10 cacheClassName Values *(Continued)*

Value	Description
com.sun.appserv.web.cache.BoundedMultiLruCache	A cache suitable for limiting the cache size by memory rather than number of entries. Uses the MaxSize property.

cache-helper

Specifies a class that implements the com.sun.appserv.web.cache.CacheHelper interface.

Superelements

"cache" on page 46 (sun-web.xml)

Subelements

The following table describes subelements for the cache-helper element.

TABLE A–11 cache-helper Subelements

Element	Required	Description
"property (with attributes)" on page 101	zero or more	Specifies a property, which has a name and a value.

Attributes

The following table describes attributes for the cache-helper element.

TABLE A–12 cache-helper Attributes

Attribute	Default	Description
name	default	Specifies a unique name for the helper class, which is referenced in the "cache-mapping" on page 49 element.
class-name	none	Specifies the fully qualified class name of the cache helper, which must implement the com.sun.appserv.web.CacheHelper interface.

cache-helper-ref

Specifies the name of the "cache-helper" on page 48 used by the parent "cache-mapping" on page 49 element.

Superelements

"cache-mapping" on page 49 (sun-web.xml)

Subelements

none - contains data

cache-idle-timeout-in-seconds

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. Default is 600.

Applies to stateful session beans and entity beans.

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

Superelements

"bean-cache" on page 44 (`sun-ejb-jar.xml`)

Subelements

none - contains data

cache-mapping

Maps a URL pattern or a servlet name to its cacheability constraints.

Superelements

"cache" on page 46 (`sun-web.xml`)

Subelements

The following table describes subelements for the `cache-mapping` element.

TABLE A–13 `cache-mapping` Subelements

Element	Required	Description
"servlet-name" on page 118	requires one `servlet-name` or `url-pattern`	Contains the name of a servlet.
"url-pattern" on page 131	requires one `servlet-name` or `url-pattern`	Contains a servlet URL pattern for which caching is enabled.

TABLE A–13 `cache-mapping` Subelements *(Continued)*

Element	Required	Description
"cache-helper-ref" on page 48	required if `dispatcher`, `timeout`, `refresh-field`, `http-method`, `key-field`, and `constraint-field` are not used	Contains the `name` of the "cache-helper" on page 48 used by the parent `cache-mapping` element.
"dispatcher" on page 62	zero or one if `cache-helper-ref` is not used	Contains a comma-separated list of `RequestDispatcher` methods for which caching is enabled.
"timeout" on page 129	zero or one if `cache-helper-ref` is not used	Contains the "cache-mapping" on page 49 specific maximum amount of time in seconds that an entry can remain in the cache after it is created or refreshed.
"refresh-field" on page 105	zero or one if `cache-helper-ref` is not used	Specifies a field that gives the application component a programmatic way to refresh a cached entry.
"http-method" on page 72	zero or more if `cache-helper-ref` is not used	Contains an HTTP method that is eligible for caching.
"key-field" on page 80	zero or more if `cache-helper-ref` is not used	Specifies a component of the key used to look up and extract cache entries.
"constraint-field" on page 56	zero or more if `cache-helper-ref` is not used	Specifies a cacheability constraint for the given `url-pattern` or `servlet-name`.

call-property

Specifies JAX-RPC property values that can be set on a `javax.xml.rpc.Call` object before it is returned to the web service client. The property names can be any properties supported by the JAX-RPC `Call` implementation.

Superelements

"port-info" on page 99, "service-ref" on page 116 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

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

TABLE A–14 `call-property` subelements

Element	Required	Description
"name" on page 94	only one	Specifies the name of the entity.
"value" on page 132	only one	Specifies the value of the entity.

caller-propagation

Specifies whether the target accepts propagated caller identities. The values are `NONE`, `SUPPORTED`, or `REQUIRED`.

Superelements

"sas-context" on page 113 (`sun-ejb-jar.xml`)

Subelements

none - contains data

checkpoint-at-end-of-method

Specifies that the stateful session bean state is checkpointed, or persisted, after the specified methods are executed. The `availability-enabled` attribute of the parent "ejb" on page 63 element must be set to `true`.

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

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `checkpoint-at-end-of-method` element.

TABLE A–15 `checkpoint-at-end-of-method` Subelements

Element	Required	Description
"method" on page 92	one or more	Specifies a bean method.

checkpointed-methods

Deprecated. Supported for backward compatibility. Use "checkpoint-at-end-of-method" on page 51 instead.

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

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

class-loader

Configures the class loader for the web module.

Superelements

"sun-web-app" on page 124 (`sun-web.xml`)

Subelements

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

TABLE A–16 `class-loader` Subelements

Element	Required	Description
"property (with attributes)" on page 101	zero or more	Specifies a property, which has a name and a value.

Attributes

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

TABLE A–17 `class-loader` Attributes

Attribute	Default	Description
`extra-class-path`	null	(optional) Specifies a colon or semicolon separated list of additional classpaths for this web module. Paths can be absolute or relative to the web module's root, for example: `extra-class-path="WEB-INF/lib/extra/extra.jar"`
`delegate`	`true`	(optional) If `true`, the web module follows the standard class loader delegation model and delegates to its parent class loader first before looking in the local class loader. You must set this to `true` for a web module that accesses EJB components or that acts as a web service client or endpoint. If `false`, the web module follows the delegation model specified in the Servlet specification and looks in its class loader before looking in the parent class loader. It's safe to set this to `false` only for a web module that does not interact with any other modules. **Note –** For Prelude, the `delegate` value is ignored and assumed to be set to `true`.
`dynamic-reload-interval`		(optional) Not implemented. Included for backward compatibility with previous Sun GlassFish Web Server versions.

Note – If the `delegate` element is set to `false`, the class loader delegation behavior complies with the Servlet 2.4 specification, section 9.7.2. If set to its default value of `true`, classes and resources residing in container-wide library JAR files are loaded in preference to classes and resources packaged within the WAR file.

Portable programs that use this element should not be packaged with any classes or interfaces that are a part of the Java EE specification. The behavior of a program that includes such classes or interfaces in its WAR file is undefined.

Properties

The following table describes properties for the `class-loader` element.

TABLE A–18 `class-loader` Properties

Property	Default	Description
`ignoreHiddenJarFiles`	`false`	If `true`, specifies that all JAR and ZIP files in the `WEB-INF/lib` directory that start with a period (`.`) are ignored by the class loader.

cmp

Describes runtime information for a CMP entity bean object for EJB 1.1 and EJB 2.1 beans.

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

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–19 `cmp` Subelements

Element	Required	Description
"mapping-properties" on page 86	zero or one	This element is not implemented.
"is-one-one-cmp" on page 74	zero or one	This element is not implemented.
"one-one-finders" on page 95	zero or one	Describes the finders for CMP 1.1 beans.

TABLE A–19 cmp Subelements *(Continued)*

Element	Required	Description
"prefetch-disabled" on page 100	zero or one	Disables prefetching of entity bean states for the specified query methods.

cmp-resource

Specifies the database to be used for storing CMP beans.

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

Superelements

"enterprise-beans" on page 67 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–20 `cmp-resource` Subelements

Element	Required	Description
"jndi-name" on page 76	only one	Specifies the absolute `jndi-name` of a JDBC resource.
"default-resource-principal" on page 61	zero or one	Specifies the default runtime bindings of a resource reference.
"property (with subelements)" on page 102	zero or more	Specifies a property name and value. Used to configure `PersistenceManagerFactory` properties.
"create-tables-at-deploy" on page 59	zero or one	If `true`, specifies that database tables are created for beans that are automatically mapped by the EJB container.
"drop-tables-at-undeploy" on page 62	zero or one	If `true`, specifies that database tables that were automatically created when the bean(s) were last deployed are dropped when the bean(s) are undeployed.
"database-vendor-name" on page 59	zero or one	Specifies the name of the database vendor for which tables can be created.
"schema-generator-properties" on page 113	zero or one	Specifies field-specific type mappings and allows you to set the `use-unique-table-names` property.

cmt-timeout-in-seconds

Overrides the Transaction Timeout setting of the Transaction Service for an individual bean. The default value, `0`, specifies that the default Transaction Service timeout is used. If positive, this value is used for all methods in the bean that start a new container-managed transaction. This value is *not* used if the bean joins a client transaction.

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

none - contains data

commit-option

Specifies the commit option used on transaction completion. Valid values for the Enterprise Server are `B` or `C`. Default value is `B`. Applies to entity beans.

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

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

none - contains data

confidentiality

Specifies if the target supports privacy-protected messages. The values are `NONE`, `SUPPORTED`, or `REQUIRED`.

Superelements

"transport-config" on page 130 (`sun-ejb-jar.xml`)

Subelements

none - contains data

constraint-field

Specifies a cacheability constraint for the given "url-pattern" on page 131 or "servlet-name" on page 118.

All `constraint-field` constraints must pass for a response to be cached. If there are `value` constraints, at least one of them must pass.

Superelements

"cache-mapping" on page 49 (`sun-web.xml`)

Subelements

The following table describes subelements for the `constraint-field` element.

TABLE A–21 `constraint-field` Subelements

Element	Required	Description
"constraint-field-value" on page 56	zero or more	Contains a value to be matched to the input parameter value.

Attributes

The following table describes attributes for the `constraint-field` element.

TABLE A–22 `constraint-field` Attributes

Attribute	Default	Description
`name`	none	Specifies the input parameter name.
`scope`	`request.parameter`	(optional) Specifies the scope from which the input parameter is retrieved. Allowed values are `context.attribute`, `request.header`, `request.parameter`, `request.cookie`, `request.attribute`, and `session.attribute`.
`cache-on-match`	`true`	(optional) If `true`, caches the response if matching succeeds. Overrides the same attribute in a "constraint-field-value" on page 56 subelement.
`cache-on-match-failure`	`false`	(optional) If `true`, caches the response if matching fails. Overrides the same attribute in a "constraint-field-value" on page 56 subelement.

constraint-field-value

Specifies a value to be matched to the input parameter value. The matching is case sensitive. For example:

```
<value match-expr="in-range">1-60</value>
```

Superelements

"constraint-field" on page 56 (`sun-web.xml`)

Subelements

none - contains data

Attributes

The following table describes attributes for the `constraint-field-value` element.

TABLE A–23 `constraint-field-value` Attributes

Attribute	Default	Description
`match-expr`	`equals`	(optional) Specifies the type of comparison performed with the value. Allowed values are `equals`, `not-equals`, `greater`, `lesser`, and `in-range`. If `match-expr` is `greater` or `lesser`, the value must be a number. If `match-expr` is `in-range`, the value must be of the form *n1*`-`*n2*, where *n1* and *n2* are numbers.
`cache-on-match`	`true`	(optional) If `true`, caches the response if matching succeeds.
`cache-on-match-failure`	`false`	(optional) If `true`, caches the response if matching fails.

context-root

Contains the web context root for the web application. Overrides the corresponding element in the `web.xml` file.

Superelements

"sun-web-app" on page 124 (`sun-web.xml`)

Subelements

none - contains data

cookie-properties

Specifies session cookie properties.

Superelements

"session-config" on page 118 (`sun-web.xml`)

Subelements

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

TABLE A–24 `cookie-properties` Subelements

Element	Required	Description
"property (with attributes)" on page 101	zero or more	Specifies a property, which has a name and a value.

Properties

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

TABLE A–25 `cookie-properties` Properties

Property	Default	Description
`cookiePath`	Context path at which the web module is installed.	Specifies the pathname that is set when the cookie is created. The browser sends the cookie if the pathname for the request contains this pathname. If set to `/` (slash), the browser sends cookies to all URLs served by the Enterprise Server. You can set the path to a narrower mapping to limit the request URLs to which the browser sends cookies.
`cookieMaxAgeSeconds`	`-1`	Specifies the expiration time (in seconds) after which the browser expires the cookie.
`cookieDomain`	(unset)	Specifies the domain for which the cookie is valid.
`cookieComment`	`Sun GlassFish Enterprise Server Session Tracking Cookie`	Specifies the comment that identifies the session tracking cookie in the cookie file. Applications can provide a more specific comment for the cookie.
`cookieSecure`	`dynamic`	Sets the `Secure` attribute of any `JSESSIONID` cookies associated with the web application. Allowed values are as follows: ■ `true` — Sets `Secure` to `true`. ■ `false` — Sets `Secure` to `false`. ■ `dynamic` — The `JSESSIONID` cookie inherits the `Secure` setting of the request that initiated the session. To set the `Secure` attribute of a `JSESSIONIDSSO` cookie, use the `ssoCookieSecure` `virtual-server` property in the `domain.xml` file. For details, see "virtual-server" in *Sun GlassFish Enterprise Server v3 Prelude Administration Reference*.

create-tables-at-deploy

Specifies whether database tables are created for beans that are automatically mapped by the EJB container. If `true`, creates tables in the database. If `false` (the default if this element is not present), does not create tables.

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

Superelements

"cmp-resource" on page 54 (`sun-ejb-jar.xml`)

Subelements

none - contains data

D

database-vendor-name

Specifies the name of the database vendor for which tables can be created. Allowed values are `javadb`, `db2`, `mssql`, `oracle`, `postgresql`, `pointbase`, `derby` (also for CloudScape), and `sybase`, case-insensitive.

If no value is specified, a connection is made to the resource specified by the "jndi-name" on page 76 subelement of the "cmp-resource" on page 54 element, and the database vendor name is read. If the connection cannot be established, or if the value is not recognized, SQL-92 compliance is presumed.

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

Superelements

"cmp-resource" on page 54 (`sun-ejb-jar.xml`)

Subelements

none - contains data

debugging-enabled

Specifies whether the debugging servlet is enabled for this web service endpoint. Allowed values are `true` (the default) and `false`.

Superelements

"webservice-endpoint" on page 134 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

default-helper

Passes property values to the built-in `default` "cache-helper" on page 48 class.

Superelements

"cache" on page 46 (`sun-web.xml`)

Subelements

The following table describes subelements for the `default-helper` element.

TABLE A–26 `default-helper` Subelements

Element	Required	Description
"property (with attributes)" on page 101	zero or more	Specifies a property, which has a name and a value.

Properties

The following table describes properties for the `default-helper` element.

TABLE A–27 `default-helper` Properties

Property	Default	Description
`cacheKeyGeneratorAttrName`	Uses the built-in `default` "cache-helper" on page 48 key generation, which concatenates the servlet path with "key-field" on page 80 values, if any.	The caching engine looks in the `ServletContext` for an attribute with a name equal to the value specified for this property to determine whether a customized `CacheKeyGenerator` implementation is used. An application can provide a customized key generator rather than using the `default` helper. See "The CacheKeyGenerator Interface" in *Sun GlassFish Enterprise Server v3 Prelude Developer's Guide*.

default-resource-principal

Specifies the default principal (user) for the resource.

If this element is used in conjunction with a JMS Connection Factory resource, the `name` and `password` subelements must be valid entries in the Sun Java™ System Message Queue broker user repository.

Superelements

"resource-ref" on page 111 (`sun-web.xml`, `sun-ejb-jar.xml`); "cmp-resource" on page 54, "mdb-connection-factory" on page 87 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–28 `default-resource-principal` Subelements

Element	Required	Description
"name" on page 94	only one	Specifies the default resource principal name used to sign on to a resource manager.
"password" on page 98	only one	Specifies password of the default resource principal.

description

Specifies a text description of the containing element.

Superelements

"property (with attributes)" on page 101 (`sun-web.xml`); "activation-config" on page 41, "method" on page 92 (`sun-ejb-jar.xml`)

Subelements

none - contains data

dispatcher

Specifies a comma-separated list of `RequestDispatcher` methods for which caching is enabled on the target resource. Valid values are `REQUEST`, `FORWARD`, `INCLUDE`, and `ERROR` . If this element is not specified, the default is `REQUEST`. See SRV.6.2.5 of the Servlet 2.4 specification for more information.

Superelements

"cache-mapping" on page 49 (`sun-web.xml`)

Subelements

none - contains data

drop-tables-at-undeploy

Specifies whether database tables that were automatically created when the bean(s) were last deployed are dropped when the bean(s) are undeployed. If `true`, drops tables from the database. If `false` (the default if this element is not present), does not drop tables.

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

Superelements

"cmp-resource" on page 54 (`sun-ejb-jar.xml`)

Subelements

none - contains data

E

ejb

Defines runtime properties for a single enterprise bean within the application. The subelements listed below apply to particular enterprise beans as follows:

- All types of beans: `ejb-name`, `ejb-ref`, `resource-ref`, `resource-env-ref`, `ior-security-config`, `gen-classes`, `jndi-name`, `use-thread-pool-id`, `message-destination-ref`, `pass-by-reference`, `service-ref`
- Stateless session beans: `bean-pool`, `webservice-endpoint`
- Stateful session beans: `bean-cache`, `webservice-endpoint`, `checkpoint-at-end-of-method`
- Entity beans: `commit-option`, `bean-cache`, `bean-pool`, `cmp`, `is-read-only-bean`, `refresh-period-in-seconds`, `flush-at-end-of-method`
- Message-driven beans: `mdb-resource-adapter`, `mdb-connection-factory`, `jms-durable-subscription-name`, `jms-max-messages-load`, `bean-pool`

Superelements

"enterprise-beans" on page 67 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–29 `ejb` Subelements

Element	Required	Description
"ejb-name" on page 66	only one	Matches the `ejb-name` in the corresponding `ejb-jar.xml` file.
"jndi-name" on page 76	zero or more	Specifies the absolute `jndi-name`.
"ejb-ref" on page 66	zero or more	Maps the absolute JNDI name to the `ejb-ref` element in the corresponding Java EE XML file.
"resource-ref" on page 111	zero or more	Maps the absolute JNDI name to the `resource-ref` in the corresponding Java EE XML file.
"resource-env-ref" on page 110	zero or more	Maps the absolute JNDI name to the `resource-env-ref` in the corresponding Java EE XML file.
"service-ref" on page 116	zero or more	Specifies runtime settings for a web service reference.

TABLE A–29 `ejb` Subelements *(Continued)*

Element	Required	Description
"message-destination-ref" on page 90	zero or more	Specifies the name of a physical message destination.
"pass-by-reference" on page 97	zero or one	Specifies the passing method used by an enterprise bean calling a remote interface method in another bean that is colocated within the same process.
"cmp" on page 53	zero or one	Specifies runtime information for a container-managed persistence (CMP) entity bean for EJB 1.1 and EJB 2.1 beans.
"principal" on page 100	zero or one	Specifies the principal (user) name in an enterprise bean that has the `run-as` role specified.
"mdb-connection-factory" on page 87	zero or one	Specifies the connection factory associated with a message-driven bean.
"jms-durable-subscription-name" on page 75	zero or one	Specifies the durable subscription associated with a message-driven bean.
"jms-max-messages-load" on page 75	zero or one	Specifies the maximum number of messages to load into a Java Message Service session at one time for a message-driven bean to serve. The default is 1.
"ior-security-config" on page 73	zero or one	Specifies the security information for the IOR.
"is-read-only-bean" on page 74	zero or one	Specifies that this entity bean is read-only.
"refresh-period-in-seconds" on page 106	zero or one	Specifies the rate at which a read-only-bean must be refreshed from the data source.
"commit-option" on page 55	zero or one	Has valid values of B or C. Default value is B.
"cmt-timeout-in-seconds" on page 55	zero or one	Overrides the Transaction Timeout setting of the Transaction Service for an individual bean.
"use-thread-pool-id" on page 131	zero or one	Specifies the thread pool from which threads are selected for remote invocations of this bean.
"gen-classes" on page 71	zero or one	Specifies all the generated class names for a bean.
"bean-pool" on page 45	zero or one	Specifies the bean pool properties. Used for stateless session beans, entity beans, and message-driven beans.
"bean-cache" on page 44	zero or one	Specifies the bean cache properties. Used only for stateful session beans and entity beans.
"mdb-resource-adapter" on page 88	zero or one	Specifies runtime configuration information for a message-driven bean.
"webservice-endpoint" on page 134	zero or more	Specifies information about a web service endpoint.
"flush-at-end-of-method" on page 70	zero or one	Specifies the methods that force a database flush after execution. Used for entity beans.

TABLE A–29 `ejb` Subelements *(Continued)*

Element	Required	Description
"checkpointed-methods" on page 51	zero or one	Deprecated. Supported for backward compatibility. Use "checkpoint-at-end-of-method" on page 51 instead.
"checkpoint-at-end-of-method" on page 51	zero or one	Specifies that the stateful session bean state is checkpointed, or persisted, after the specified methods are executed. The `availability-enabled` attribute must be set to `true`.

Attributes

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

TABLE A–30 `ejb` Attributes

Attribute	Default	Description
`availability-enabled`	`false`	(optional) If set to `true`, and if availability is enabled in the EJB container, high-availability features apply to this bean if it is a stateful session bean.

Example

```
<ejb>
   <ejb-name>CustomerEJB</ejb-name>
   <jndi-name>customer</jndi-name>
   <resource-ref>
      <res-ref-name>jdbc/SimpleBank</res-ref-name>
      <jndi-name>jdbc/__default</jndi-name>
   </resource-ref>
   <is-read-only-bean>false</is-read-only-bean>
   <commit-option>B</commit-option>
   <bean-pool>
      <steady-pool-size>10</steady-pool-size>
      <resize-quantity>10</resize-quantity>
      <max-pool-size>100</max-pool-size>
      <pool-idle-timeout-in-seconds>600</pool-idle-timeout-in-seconds>
   </bean-pool>
   <bean-cache>
      <max-cache-size>100</max-cache-size>
      <resize-quantity>10</resize-quantity>
      <removal-timeout-in-seconds>3600</removal-timeout-in-seconds>
      <victim-selection-policy>LRU</victim-selection-policy>
   </bean-cache>
</ejb>
```

ejb-name

In the `sun-ejb-jar.xml` file, matches the `ejb-name` in the corresponding `ejb-jar.xml` file. The name must be unique among the names of the enterprise beans in the same EJB JAR file.

There is no architected relationship between the `ejb-name` in the deployment descriptor and the JNDI name that the deployer assigns to the EJB component's home.

Superelements

"ejb" on page 63, "method" on page 92 (`sun-ejb-jar.xml`)

Subelements

none - contains data

ejb-ref

Maps the `ejb-ref-name` in the corresponding Java EE deployment descriptor file `ejb-ref` entry to the absolute `jndi-name` of a resource.

The `ejb-ref` element is used for the declaration of a reference to an EJB's home. Applies to session beans or entity beans.

Superelements

"sun-web-app" on page 124 (`sun-web.xml`), "ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–31 `ejb-ref` Subelements

Element	Required	Description
"ejb-ref-name" on page 66	only one	Specifies the `ejb-ref-name` in the corresponding Java EE deployment descriptor file `ejb-ref` entry.
"jndi-name" on page 76	only one	Specifies the absolute `jndi-name` of a resource.

ejb-ref-name

Specifies the `ejb-ref-name` in the corresponding Java EE deployment descriptor file `ejb-ref` entry.

Superelements

"ejb-ref" on page 66 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

endpoint-address-uri

Specifies the relative path combined with the web server root to form the fully qualified endpoint address for a web service endpoint. This is a required element for EJB endpoints and an optional element for servlet endpoints.

For servlet endpoints, this value is relative to the web application context root. For EJB endpoints, the URI is relative to root of the web server (the first portion of the URI is a context root). The context root portion must not conflict with the context root of any web application deployed to the same web server.

In all cases, this value must be a fixed pattern (no "*' allowed).

If the web service endpoint is a servlet that implements only a single endpoint and has only one `url-pattern`, it is not necessary to set this value, because the web container derives it from the `web.xml` file.

Superelements

"webservice-endpoint" on page 134 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

Example

If the web server is listening at `http://localhost:8080`, the following `endpoint-address-uri`:

```
<endpoint-address-uri>StockQuoteService/StockQuotePort</endpoint-address-uri>
```

results in the following target endpoint address:

```
http://localhost:8080/StockQuoteService/StockQuotePort
```

enterprise-beans

Specifies all the runtime properties for an EJB JAR file in the application.

Superelements

"sun-ejb-jar" on page 123 (`sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `enterprise-beans` element.

TABLE A–32 `enterprise-beans` Subelements

Element	Required	Description
"name" on page 94	zero or one	Specifies the name string.
"unique-id" on page 131	zero or one	Specifies a unique system identifier. This data is automatically generated and updated at deployment/redeployment. Do not specify or edit this value.
"ejb" on page 63	zero or more	Defines runtime properties for a single enterprise bean within the application.
"pm-descriptors" on page 98	zero or one	Deprecated.
"cmp-resource" on page 54	zero or one	Specifies the database to be used for storing container-managed persistence (CMP) beans in an EJB JAR file.
"message-destination" on page 89	zero or more	Specifies the name of a logical message destination.
"webservice-description" on page 133	zero or more	Specifies a name and optional publish location for a web service.

Example

```
<enterprise-beans>
 <ejb>
     <ejb-name>CustomerEJB</ejb-name>
     <jndi-name>customer</jndi-name>
     <resource-ref>
         <res-ref-name>jdbc/SimpleBank</res-ref-name>
         <jndi-name>jdbc/__default</jndi-name>
     </resource-ref>
     <is-read-only-bean>false</is-read-only-bean>
     <commit-option>B</commit-option>
     <bean-pool>
         <steady-pool-size>10</steady-pool-size>
        <resize-quantity>10</resize-quantity>
         <max-pool-size>100</max-pool-size>
         <pool-idle-timeout-in-seconds>600</pool-idle-timeout-in-seconds>
     </bean-pool>
     <bean-cache>
         <max-cache-size>100</max-cache-size>
         <resize-quantity>10</resize-quantity>
```

```
        <removal-timeout-in-seconds>3600</removal-timeout-in-seconds>
        <victim-selection-policy>LRU</victim-selection-policy>
    </bean-cache>
 </ejb>
</enterprise-beans>
```

establish-trust-in-client

Specifies if the target is capable of authenticating a client. The values are `NONE`, `SUPPORTED`, or `REQUIRED`.

Superelements

"transport-config" on page 130 (`sun-ejb-jar.xml`)

Subelements

none - contains data

establish-trust-in-target

Specifies if the target is capable of authenticating *to* a client. The values are `NONE`, `SUPPORTED`, or `REQUIRED`.

Superelements

"transport-config" on page 130 (`sun-ejb-jar.xml`)

Subelements

none - contains data

F

finder

Describes the finders for CMP 1.1 with a method name and query.

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

Superelements

"one-one-finders" on page 95 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–33 `finder` Subelements

Element	Required	Description
"method-name" on page 93	only one	Specifies the method name for the finder.
"query-params" on page 104	zero or one	Specifies the query parameters for the CMP 1.1 finder.
"query-filter" on page 103	zero or one	Specifies the query filter for the CMP 1.1 finder.
"query-variables" on page 104	zero or one	Specifies variables in query expression for the CMP 1.1 finder.
"query-ordering" on page 104	zero or one	Specifies the query ordering for the CMP 1.1 finder.

flush-at-end-of-method

Specifies the methods that force a database flush after execution. Applicable to entity beans.

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

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `flush-at-end-of-method` element.

TABLE A–34 `flush-at-end-of-method` Subelements

Element	Required	Description
"method" on page 92	one or more	Specifies a bean method.

G

gen-classes

Specifies all the generated class names for a bean.

Note – This value is automatically generated by the server at deployment or redeployment time. Do not specify it or change it after deployment.

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

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `gen-classes` element.

TABLE A–35 `gen-classes` Subelements

Element	Required	Description
"remote-impl" on page 107	zero or one	Specifies the fully-qualified class name of the generated `EJBObject` impl class.
"local-impl" on page 81	zero or one	Specifies the fully-qualified class name of the generated `EJBLocalObject` impl class.
"remote-home-impl" on page 107	zero or one	Specifies the fully-qualified class name of the generated EJBHome impl class.
"local-home-impl" on page 80	zero or one	Specifies the fully-qualified class name of the generated `EJBLocalHome` impl class.

group-name

Specifies a group name in the current realm.

Superelements

"security-role-mapping" on page 115 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

H

http-method

Specifies an HTTP method that is eligible for caching. The default is `GET`.

Superelements

"cache-mapping" on page 49 (`sun-web.xml`)

Subelements

none - contains data

I

idempotent-url-pattern

Specifies a URL pattern for idempotent requests.

Superelements

"sun-web-app" on page 124 (`sun-web.xml`)

Subelements

none

Attributes

The following table describes attributes for the `idempotent-url-pattern` element.

TABLE A–36 `idempotent-url-pattern` Attributes

Attribute	Default	Description
`url-pattern`	none	Specifies a URL pattern, which can contain wildcards. The URL pattern must conform to the mappings specified in section SRV 11.2 of the Servlet 2.4 specification.
`no-of-retries`	`-1`	(optional) Specifies the number of times the load balancer retries an idempotent request. A value of `-1` indicates infinite retries.

Example

The following example specifies that all requests for the URI `sun-java/*` are idempotent.

```
<idempotent-url-pattern url-pattern="sun_java/*" no-of-retries="10"/>
```

integrity

Specifies if the target supports integrity-protected messages. The values are `NONE`, `SUPPORTED`, or `REQUIRED`.

Superelements

"transport-config" on page 130 (`sun-ejb-jar.xml`)

Subelements

none - contains data

ior-security-config

Specifies the security information for the input-output redirection (IOR).

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–37 `ior-security-config` Subelements

Element	Required	Description
"transport-config" on page 130	zero or one	Specifies the security information for transport.
"as-context" on page 43	zero or one	Specifies the authentication mechanism used to authenticate the client. If specified, it is `USERNAME_PASSWORD`.
"sas-context" on page 113	zero or one	Describes the sas-context fields.

is-cache-overflow-allowed

This element is deprecated. Do not use.

Superelements

"bean-cache" on page 44 (`sun-ejb-jar.xml`)

is-one-one-cmp

This element is not used.

Superelements

"cmp" on page 53 (`sun-ejb-jar.xml`)

is-read-only-bean

Specifies that this entity bean is a read-only bean if `true`. If this element is absent, the default value of `false` is used.

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

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

none - contains data

J

java-method

Specifies a method.

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

Superelements

"message" on page 88 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

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

TABLE A–38 `java-method` Subelements

Element	Required	Description
"method-name" on page 93	only one	Specifies a method name.
"method-params" on page 94	zero or one	Specifies fully qualified Java type names of method parameters.

jms-durable-subscription-name

Specifies the durable subscription associated with a message-driven bean class. Only applies to the Java Message Service Topic Destination type, and only when the message-driven bean deployment descriptor subscription durability is Durable.

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

none - contains data

jms-max-messages-load

Specifies the maximum number of messages to load into a Java Message Service session at one time for a message-driven bean to serve. The default is 1.

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

none - contains data

jndi-name

Specifies the absolute `jndi-name` of a URL resource or a resource.

For entity beans and session beans, this value specifies the global JNDI name of the `EJBHome` object. It is only needed if the entity or session bean exposes a remote view.

For JMS message-driven beans, this is the JNDI name of the JMS resource from which the message-driven bean consumes JMS messages. This information is alternatively specified within the "activation-config" on page 41 subelement of the "mdb-resource-adapter" on page 88 element.

Superelements

"ejb-ref" on page 66, "message-destination" on page 89, "resource-env-ref" on page 110, "resource-ref" on page 111 (`sun-web.xml`, `sun-ejb-jar.xml`); "cmp-resource" on page 54, "ejb" on page 63, "mdb-connection-factory" on page 87 (`sun-ejb-jar.xml`)

Subelements

none - contains data

jsp-config

Specifies JSP configuration information.

Superelements

"sun-web-app" on page 124 (`sun-web.xml`)

Subelements

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

TABLE A–39 `jsp-config` Subelements

Element	Required	Description
"property (with attributes)" on page 101	zero or more	Specifies a property, which has a name and a value.

Properties

The default property values are tuned for development of JSP files at the cost of performance. To maximize performance, set `jsp-config` properties to these non-default values:

- `development` - `false` (as an alternative, set to `true` and give `modificationTestInterval` a large value)
- `mappedfile` - `false`
- `trimSpaces` - `true`
- `suppressSmap` - `true`
- `fork` - `false` (on Solaris)
- `classdebuginfo` - `false`

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

TABLE A–40 `jsp-config` Properties

Property	Default	Description
`checkInterval`	`0`	If `development` is set to `false` and `checkInterval` is greater than zero, background compilations are enabled. The `checkInterval` is the time in seconds between checks to see if a JSP file needs to be recompiled.
`classdebuginfo`	`true`	Specifies whether the generated Java servlets are compiled with the debug option set (`-g` for `javac`).
`classpath`	created dynamically based on the current web application	Specifies the classpath to use when compiling generated servlets.
`compiler`	`javac`	Specifies the compiler Ant uses to compile JSP files. See the Ant documentation for more information: http://antinstaller.sourceforge.net/manual/manual/
`compilerSourceVM`	Depends on the Enterprise Server's Java runtime	Specifies the JDK release with which source compatibility of the generated servlets is provided. Same as the `-source` *release* option of `javac`. For more information, see http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/javac.html#options.

TABLE A–40 `jsp-config` Properties *(Continued)*

Property	Default	Description
`compilerTargetVM`	Depends on the Enterprise Server's Java runtime	Specifies the Virtual Machine for the Java platform (JVM™ software) version for which the servlet class files are generated. Same as the `-target` *release* option of `javac`. For more information, see http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/javac.html#options.
`defaultBufferNone`	`false`	If `true`, the default for the `buffer` attribute of the `page` directive is `none`.
`development`	`true`	If set to `true`, enables development mode, which allows JSP files to be checked for modification. Specify the frequency at which JSPs are checked using the `modificationTestInterval` property.
`dumpSmap`	`false`	If set to `true`, dumps SMAP information for JSR 45 debugging to a file. Set to `false` if `suppressSmap` is `true`.
`enablePooling`	`true`	If set to `true`, tag handler pooling is enabled.
`enableTldValidation`	`false`	If set to `true`, all Tag Library Descriptor (TLD) files referenced by the web application are validated against their underlying schema or DTD file.
`errorOnUseBeanInvalid ClassAttribute`	`false`	If set to `true`, issues an error when the value of the `class` attribute in a `useBean` action is not a valid bean class.
`fork`	`true`	Specifies that Ant forks the compiling of JSP files, using a JVM machine separate from the one in which Tomcat is running.
`genStrAsByteArray`	`true`	If `true`, text strings are generated as bytes (encoded with the page encoding), if the page is not buffered.
`genStrAsCharArray`	`false`	If set to `true`, generates text strings as `char` arrays, which improves performance in some cases.
`httpMethods`	* for all methods	Specifies a comma separated list of HTTP methods supported by the `JspServlet`.
`ieClassId`	`clsid:8AD9C840-044E-11D1-B3E9-00805F499D93`	Specifies the Java plug-in COM class ID for Internet Explorer. Used by the `<jsp:plugin>` tags.
`ignoreJspFragmentErrors`	`false`	If set to `true`, instructs the compiler to ignore any JSP precompilation errors pertaining to statically included JSP segments that, despite not being top level JSP files, use the `.jsp` or `.jspx` extension (instead of the recommended `.jspf`).
`initialCapacity`	`32`	Specifies the initial capacity of the `HashMap` that maps JSP files to their corresponding servlets.

TABLE A–40 `jsp-config` Properties *(Continued)*

Property	Default	Description
`javaEncoding`	`UTF8`	Specifies the encoding for the generated Java servlet. This encoding is passed to the Java compiler that is used to compile the servlet as well. By default, the web container tries to use `UTF8`. If that fails, it tries to use the `javaEncoding` value. For encodings, see: http://java.sun.com/j2se/1.5.0/docs/guide/intl/encoding.doc.html
`keepgenerated`	`true` with JDK 5 and before and for `jspc`, otherwise `false`	If set to `true`, keeps the generated Java files. If `false`, deletes the Java files.
`mappedfile`	`true`	If set to `true`, generates static content with one print statement per input line, to ease debugging.
`modification TestInterval`	`0`	Specifies the frequency in seconds at which JSPs are checked for modification. A value of `0` causes the JSP to be checked on every access. Used only if `development` is set to `true`.
`reload-interval`	`0`	Specifies the frequency in seconds at which JSP files are checked for modifications. Setting this value to `0` checks JSP files for modifications on every request. Setting this value to `-1` disables checks for JSP modifications and JSP recompilation.
`saveBytecode`	`true` for `jspc`, otherwise `false`	If `true`, generated byte code is saved to `.class` files? This option is meaningful only when the Java compiler API, JSR 199 (available with and used as the default on Java 6) is used for `javac` compilations.
`scratchdir`	The default work directory for the web application	Specifies the working directory created for storing all the generated code.
`suppressSmap`	`false`	If set to `true`, generation of SMAP information for JSR 45 debugging is suppressed.
`trimSpaces`	`false`	If set to `true`, trims white spaces in template text between actions or directives.
`usePrecompiled`	`false`	If set to `true`, an accessed JSP file is not compiled. Its precompiled servlet class is used instead. It is assumed that JSP files have been precompiled, and their corresponding servlet classes have been bundled in the web application's `WEB-INF/lib` or `WEB-INF/classes` directory.
`xpoweredBy`	`true`	If set to `true`, the X-Powered-By response header is added by the generated servlet.

K

key-field

Specifies a component of the key used to look up and extract cache entries. The web container looks for the named parameter, or field, in the specified scope.

If this element is not present, the web container uses the Servlet Path (the path section that corresponds to the servlet mapping that activated the current request). See the Servlet 2.4 specification, section SRV 4.4, for details on the Servlet Path.

Superelements

"cache-mapping" on page 49 (`sun-web.xml`)

Subelements

none

Attributes

The following table describes attributes for the `key-field` element.

TABLE A–41 `key-field` Attributes

Attribute	Default	Description
`name`	none	Specifies the input parameter name.
`scope`	`request.parameter`	(optional) Specifies the scope from which the input parameter is retrieved. Allowed values are `context.attribute`, `request.header`, `request.parameter`, `request.cookie`, `session.id`, and `session.attribute`.

L

local-home-impl

Specifies the fully-qualified class name of the generated `EJBLocalHome impl` class.

Note – This value is automatically generated by the server at deployment or redeployment time. Do not specify it or change it after deployment.

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

Superelements

"gen-classes" on page 71 (`sun-ejb-jar.xml`)

Subelements

none - contains data

local-impl

Specifies the fully-qualified class name of the generated `EJBLocalObject impl` class.

Note – This value is automatically generated by the server at deployment or redeployment time. Do not specify it or change it after deployment.

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

Superelements

"gen-classes" on page 71 (`sun-ejb-jar.xml`)

Subelements

none - contains data

locale-charset-info

Deprecated. For backward compatibility only. Use the "parameter-encoding" on page 96 subelement of "sun-web-app" on page 124 instead. Specifies information about the application's internationalization settings.

Superelements

"sun-web-app" on page 124 (`sun-web.xml`)

Subelements

The following table describes subelements for the `locale-charset-info` element.

TABLE A–42 `locale-charset-info` Subelements

Element	Required	Description
"locale-charset-map" on page 82	one or more	Maps a locale and an agent to a character encoding. Provided for backward compatibility. Used only for request processing, and only if no `parameter-encoding` is defined.
"parameter-encoding" on page 96	zero or one	Determines the default request character encoding and how the web container decodes parameters from forms according to a hidden field value.

Attributes

The following table describes attributes for the `locale-charset-info` element.

TABLE A–43 `locale-charset-info` Attributes

Attribute	Default	Description
`default-locale`	none	Although a value is required, the value is ignored. Use the `default-charset` attribute of the "parameter-encoding" on page 96 element.

locale-charset-map

Maps locales and agents to character encodings. Provided for backward compatibility. Used only for request processing. Used only if the character encoding is not specified in the request and cannot be derived from the optional "parameter-encoding" on page 96 element. For encodings, see `http://java.sun.com/j2se/1.5.0/docs/guide/intl/encoding.doc.html`.

Superelements

"locale-charset-info" on page 81 (`sun-web.xml`)

Subelements

The following table describes subelements for the `locale-charset-map` element.

TABLE A–44 `locale-charset-map` Subelements

Element	Required	Description
"description" on page 61	zero or one	Specifies an optional text description of a mapping.

Attributes

The following table describes attributes for the `locale-charset-map` element.

TABLE A–45 `locale-charset-map` Attributes

Attribute	Default	Description
`locale`	none	Specifies the locale name.
`agent`	none	(optional) Specifies the type of client that interacts with the application server. For a given locale, different agents can have different preferred character encodings. The value of this attribute must exactly match the value of the `user-agent` HTTP request header sent by the client. See Table A–46 for more information.
`charset`	none	Specifies the character encoding to which the locale maps.

Example Agents

The following table specifies example `agent` attribute values.

TABLE A–46 Example `agent` Attribute Values

Agent	`user-agent` Header and `agent` Attribute Value
Internet Explorer 5.00 for Windows 2000	`Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)`
Netscape 4.7.7 for Windows 2000	`Mozilla/4.77 [en] (Windows NT 5.0; U)`
Netscape 4.7 for Solaris	`Mozilla/4.7 [en] (X11; u; Sun OS 5.6 sun4u)`

localpart

Specifies the local part of a QNAME.

Superelements

"service-qname" on page 116, "wsdl-port" on page 135 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

login-config

Specifies the authentication configuration for an EJB web service endpoint. Not needed for servlet web service endpoints. A servlet's security configuration is contained in the `web.xml` file.

Superelements

"webservice-endpoint" on page 134 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

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

TABLE A–47 `login-config` subelements

Element	Required	Description
"auth-method" on page 43	only one	Specifies the authentication method.
"realm" on page 105	zero or one	Specifies the name of the realm used to process all authentication requests.

M

manager-properties

Specifies session manager properties.

Superelements

"session-manager" on page 119 (`sun-web.xml`)

Subelements

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

TABLE A–48 `manager-properties` Subelements

Element	Required	Description
"property (with attributes)" on page 101	zero or more	Specifies a property, which has a name and a value.

Properties

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

TABLE A–49 `manager-properties` Properties

Property	Default	Description
`reapIntervalSeconds`	`60`	Specifies the number of seconds between checks for expired sessions. This is also the interval at which sessions are passivated if `maxSessions` is exceeded. If `persistenceFrequency` is set to `time-based`, active sessions are stored at this interval. To prevent data inconsistency, 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 the last few hits might be lost each time the server is restarted. Applicable only if the `persistence-type` attribute of the parent "session-manager" on page 119 element is `file` or `replicated`. **Note –** The `replicated` persistence type is not supported in GlassFish v3 Prelude.
`maxSessions`	`-1`	Specifies the maximum number of sessions that are permitted in the cache, or `-1` for no limit. After this, an attempt to create a new session causes an `IllegalStateException` to be thrown.
`sessionFilename`	One of the following: *domain-dir*`/generated/jsp/`*module-name*`/`*context-path*`_SESSIONS.ser` *domain-dir*`/generated/jsp/`*app-name*`/`*module-name*`/`*context-path*`_SESSIONS.ser`	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 module. To disable preservation of the session state, set this property's value to an empty string. Applicable only if the `persistence-type` attribute of the parent "session-manager" on page 119 element is `memory`. To disable this behavior and not preserve the session state, specify an empty string as the value of this property.

TABLE A–49 `manager-properties` Properties *(Continued)*

Property	Default	Description
`persistenceFrequency`	`web-method`	Specifies how often the session state is stored. Allowed values are as follows: ■ `web-method` - The session state is stored at the end of each web request prior to sending a response back to the client. This mode provides the best guarantee that the session state is fully updated in case of failure. ■ `time-based` - The session state is stored in the background at the frequency set by `reapIntervalSeconds`. This mode provides less of a guarantee that the session state is fully updated. However, it can provide a significant performance improvement because the state is not stored after each request. Applicable only if the `persistence-type` attribute of the parent "session-manager" on page 119 element is `replicated`. **Note –** The `replicated` persistence type is not supported in GlassFish v3 Prelude.

mapping-properties

This element is not implemented.

Superelements

"cmp" on page 53 (`sun-ejb-jar.xml`)

max-cache-size

Specifies the maximum number of beans allowable in cache. A value of zero indicates an unbounded cache. In reality, there is no hard limit. The max-cache-size limit is just a hint to the cache implementation. Default is 512.

Applies to stateful session beans and entity beans.

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

Superelements

"bean-cache" on page 44 (`sun-ejb-jar.xml`)

Subelements

none - contains data

max-pool-size

Specifies the maximum number of bean instances in the pool. Values are from 0 (1 for message-driven bean) to MAX_INTEGER. A value of 0 means the pool is unbounded. Default is 64.

Applies to all beans.

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

Superelements

"bean-pool" on page 45 (`sun-ejb-jar.xml`)

Subelements

none - contains data

max-wait-time-in-millis

This element is deprecated. Do not use.

Superelements

"bean-pool" on page 45 (`sun-ejb-jar.xml`)

mdb-connection-factory

Specifies the connection factory associated with a message-driven bean. Queue or Topic type must be consistent with the Java Message Service Destination type associated with the message-driven bean class.

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–50 `mdb-connection-factory` Subelements

Element	Required	Description
"jndi-name" on page 76	only one	Specifies the absolute `jndi-name`.
"default-resource-principal" on page 61	zero or one	Specifies the default sign-on (name/password) to the resource manager.

mdb-resource-adapter

Specifies runtime configuration information for a message-driven bean.

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–51 `mdb-resource-adapter` subelements

Element	Required	Description
"resource-adapter-mid" on page 110	zero or one	Specifies a resource adapter module ID.
"activation-config" on page 41	one or more	Specifies an activation configuration.

message

Specifies the methods or operations to which message security requirements apply.

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

Superelements

"message-security" on page 90 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

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

TABLE A–52 `message` Subelements

Element	Required	Description
"java-method" on page 75	zero or one	Specifies the methods or operations to which message security requirements apply.
"operation-name" on page 96	zero or one	Specifies the WSDL name of an operation of a web service.

message-destination

Specifies the name of a logical `message-destination` defined within an application. The `message-destination-name` matches the corresponding `message-destination-name` in the corresponding Java EE deployment descriptor file. Use when the message destination reference in the corresponding Java EE deployment descriptor file specifies a `message-destination-link` to a logical `message-destination`.

Superelements

"sun-web-app" on page 124 (`sun-web.xml`), "enterprise-beans" on page 67 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–53 `message-destination` subelements

Element	Required	Description
"message-destination-name" on page 89	only one	Specifies the name of a logical message destination defined within the corresponding Java EE deployment descriptor file.
"jndi-name" on page 76	only one	Specifies the `jndi-name` of the associated entity.

message-destination-name

Specifies the name of a logical message destination defined within the corresponding Java EE deployment descriptor file.

Superelements

"message-destination" on page 89 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

message-destination-ref

Directly binds a message destination reference to the JNDI name of a `Queue`, `Topic`, or other physical destination. Use only when the message destination reference in the corresponding Java EE deployment descriptor file does *not* specify a `message-destination-link` to a logical `message-destination`.

Superelements

"sun-web-app" on page 124 (`sun-web.xml`), "ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `message-destination-ref` element.

TABLE A–54 `message-destination-ref` subelements

Element	Required	Description
"message-destination-ref-name" on page 90	only one	Specifies the name of a physical message destination defined within the corresponding Java EE deployment descriptor file.
"jndi-name" on page 76	only one	Specifies the `jndi-name` of the associated entity.

message-destination-ref-name

Specifies the name of a physical message destination defined within the corresponding Java EE deployment descriptor file.

Superelements

"message-destination-ref" on page 90 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

message-security

Specifies message security requirements.

- If the grandparent element is "webservice-endpoint" on page 134, these requirements pertain to request and response messages of the endpoint.
- If the grandparent element is "port-info" on page 99, these requirements pertain to the port of the referenced service.

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

Superelements

"message-security-binding" on page 91 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

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

TABLE A–55 `message-security` Subelements

Element	Required	Description
"message" on page 88	one or more	Specifies the methods or operations to which message security requirements apply.
"request-protection" on page 107	zero or one	Defines the authentication policy requirements of the application's request processing.
"response-protection" on page 112	zero or one	Defines the authentication policy requirements of the application's response processing.

message-security-binding

Specifies a custom authentication provider binding for a parent "webservice-endpoint" on page 134 or "port-info" on page 99 element in one or both of these ways:

- By binding to a specific provider
- By specifying the message security requirements enforced by the provider

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

Superelements

"webservice-endpoint" on page 134, "port-info" on page 99 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

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

TABLE A–56 `message-security-binding` Subelements

Element	Required	Description
"message-security" on page 90	zero or more	Specifies message security requirements.

Attributes

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

TABLE A–57 `message-security-binding` Attributes

Attribute	Default	Description
`auth-layer`	none	Specifies the message layer at which authentication is performed. The value must be `SOAP`.
`provider-id`	none	(optional) Specifies the authentication provider used to satisfy application-specific message security requirements. If this attribute is not specified, a default provider is used, if it is defined for the message layer. if no default provider is defined, authentication requirements defined in the `message-security-binding` are not enforced.

method

Specifies a bean method.

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

Superelements

"checkpoint-at-end-of-method" on page 51, "flush-at-end-of-method" on page 70
(`sun-ejb-jar.xml`)

Subelements

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

TABLE A–58 `method` Subelements

Element	Required	Description
"description" on page 61	zero or one	Specifies an optional text description.
"ejb-name" on page 66	zero or one	Matches the `ejb-name` in the corresponding `ejb-jar.xml` file.

TABLE A–58 `method` Subelements *(Continued)*

Element	Required	Description
"method-name" on page 93	only one	Specifies a method name.
"method-intf" on page 93	zero or one	Specifies the method interface to distinguish between methods with the same name in different interfaces.
"method-params" on page 94	zero or one	Specifies fully qualified Java type names of method parameters.

method-intf

Specifies the method interface to distinguish between methods with the same name in different interfaces. Allowed values are `Home`, `Remote`, `LocalHome`, and `Local`.

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

Superelements

"method" on page 92 (`sun-ejb-jar.xml`)

Subelements

none - contains data

method-name

Specifies a method name or * (an asterisk) for all methods. If a method is overloaded, specifies all methods with the same name.

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

Superelements

"java-method" on page 75 (`sun-web.xml`, `sun-ejb-jar.xml`); "finder" on page 69, "query-method" on page 103 , "method" on page 92 (`sun-ejb-jar.xml`)

Subelements

none - contains data

Examples

```
<method-name>findTeammates</method-name>
```

```
<method-name>*</method-name>
```

method-param

Specifies the fully qualified Java type name of a method parameter.

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

Superelements

"method-params" on page 94 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

method-params

Specifies fully qualified Java type names of method parameters.

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

Superelements

"java-method" on page 75 (`sun-web.xml`, `sun-ejb-jar.xml`); "query-method" on page 103, "method" on page 92 (`sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `method-params` element.

TABLE A–59 `method-params` Subelements

Element	Required	Description
"method-param" on page 94	zero or more	Specifies the fully qualified Java type name of a method parameter.

N

name

Specifies the name of the entity.

Superelements

"call-property" on page 50, "default-resource-principal" on page 61, "stub-property" on page 122 (`sun-web.xml`, `sun-ejb-jar.xml`); "enterprise-beans" on page 67, "principal" on page 100, "property (with subelements)" on page 102 (`sun-ejb-jar.xml`)

Subelements

none - contains data

namespaceURI

Specifies the namespace URI.

Superelements

"service-qname" on page 116, "wsdl-port" on page 135 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

O

one-one-finders

Describes the finders for CMP 1.1 beans.

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

Superelements

"cmp" on page 53 (`sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `one-one-finders` element.

TABLE A–60 `one-one-finders` Subelements

Element	Required	Description
"finder" on page 69	one or more	Describes the finders for CMP 1.1 with a method name and query.

operation-name

Specifies the WSDL name of an operation of a web service.

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

Superelements

"message" on page 88 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

P

parameter-encoding

Specifies the default request character encoding and how the web container decodes parameters from forms according to a hidden field value.

If both the "sun-web-app" on page 124 and "locale-charset-info" on page 81 elements have `parameter-encoding` subelements, the subelement of `sun-web-app` takes precedence. For encodings, see `http://java.sun.com/j2se/1.5.0/docs/guide/intl/encoding.doc.html`.

Superelements

"locale-charset-info" on page 81, "sun-web-app" on page 124 (`sun-web.xml`)

Subelements

none

Attributes

The following table describes attributes for the `parameter-encoding` element.

TABLE A–61 `parameter-encoding` Attributes

Attribute	Default	Description
`form-hint-field`	none	(optional) The name of the hidden field in the form. This field specifies the character encoding the web container uses for `request.getParameter` and `request.getReader` calls when the charset is not set in the request's `content-type` header.
`default-charset`	`ISO-8859-1`	(optional) The default request character encoding.

pass-by-reference

Specifies the passing method used by a servlet or enterprise bean calling a remote interface method in another bean that is colocated within the same process.

- If `false` (the default if this element is not present), this application uses pass-by-value semantics.
- If `true`, this application uses pass-by-reference semantics.

Note – The `pass-by-reference` element only applies to remote calls. As defined in the EJB 2.1 specification, section 5.4, calls to local interfaces use pass-by-reference semantics.

If the `pass-by-reference` element is set to its default value of `false`, the passing semantics for calls to remote interfaces comply with the EJB 2.1 specification, section 5.4. If set to `true`, remote calls involve pass-by-reference semantics instead of pass-by-value semantics, contrary to this specification.

Portable programs cannot assume that a copy of the object is made during such a call, and thus that it's safe to modify the original. Nor can they assume that a copy is not made, and thus that changes to the object are visible to both caller and callee. When this element is set to `true`, parameters and return values should be considered read-only. The behavior of a program that modifies such parameters or return values is undefined.

When a servlet or enterprise bean calls a remote interface method in another bean that is colocated within the same process, by default the Enterprise Server makes copies of all the call parameters in order to preserve the pass-by-value semantics. This increases the call overhead and decreases performance.

However, if the calling method does not change the object being passed as a parameter, it is safe to pass the object itself without making a copy of it. To do this, set the pass-by-reference value to `true`.

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

none - contains data

password

Specifies the password for the principal.

Superelements

"default-resource-principal" on page 61 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

pm-descriptors

This element and its subelements are deprecated. Do not use.

Superelements

"enterprise-beans" on page 67 (`sun-ejb-jar.xml`)

pool-idle-timeout-in-seconds

Specifies the maximum time, in seconds, that a bean instance is allowed to remain idle in the pool. When this timeout expires, the bean instance in a pool becomes a candidate for passivation or deletion. This is a hint to the server. A value of 0 specifies that idle beans remain in the pool indefinitely. Default value is 600.

Applies to stateless session beans, entity beans, and message-driven beans.

Note – For a stateless session bean or a message-driven bean, the bean is removed (garbage collected) when the timeout expires.

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

Superelements

"bean-pool" on page 45 (`sun-ejb-jar.xml`)

Subelements

none - contains data

port-component-name

Specifies a unique name for a port component within a web or EJB module.

Superelements

"webservice-endpoint" on page 134 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

port-info

Specifies information for a port within a web service reference.

Either a `service-endpoint-interface` or a `wsdl-port` or both must be specified. If both are specified, `wsdl-port` specifies the port that the container chooses for container-managed port selection.

The same `wsdl-port` value must not appear in more than one `port-info` element within the same `service-ref`.

If a `service-endpoint-interface` is using container-managed port selection, its value must not appear in more than one `port-info` element within the same `service-ref`.

Superelements

"service-ref" on page 116 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `port-info` element.

TABLE A–62 `port-info` subelements

Element	Required	Description
"service-endpoint-interface" on page 115	zero or one	Specifies the web service reference name relative to `java:comp/env`.

TABLE A–62 `port-info` subelements *(Continued)*

Element	Required	Description
"wsdl-port" on page 135	zero or one	Specifies the WSDL port.
"stub-property" on page 122	zero or more	Specifies JAX-RPC property values that are set on a `javax.xml.rpc.Stub` object before it is returned to the web service client.
"call-property" on page 50	zero or more	Specifies JAX-RPC property values that are set on a `javax.xml.rpc.Call` object before it is returned to the web service client.
"message-security-binding" on page 91	zero or one	Specifies a custom authentication provider binding.

prefetch-disabled

Disables prefetching of entity bean states for the specified query methods.

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

Superelements

"cmp" on page 53 (`sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `prefetch-disabled` element.

TABLE A–63 `prefetch-disabled` Subelements

Element	Required	Description
"query-method" on page 103	one or more	Specifies a query method.

principal

Defines a node that specifies a user name on the platform.

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–64 `principal` Subelements

Element	Required	Description
"name" on page 94	only one	Specifies the name of the user.

principal-name

Contains the principal (user) name.

In an enterprise bean, specifies the principal (user) name that has the `run-as` role specified.

Superelements

"security-role-mapping" on page 115 (`sun-web.xml`, `sun-ejb-jar.xml`), "servlet" on page 117 (`sun-web.xml`)

Subelements

none - contains data

Attributes

The following table describes attributes for the `principal-name` element.

TABLE A–65 `principal-name` Attributes

Attribute	Default	Description
`class-name`	`com.sun.enterprise.deployment.PrincipalImpl`	(optional) Specifies the custom principal implementation class corresponding to the named principal.

property (with attributes)

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

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

Superelements

"cache" on page 46, "cache-helper" on page 48, "class-loader" on page 52, "cookie-properties" on page 57, "default-helper" on page 60, "manager-properties" on page 84, "session-properties" on page 120, "store-properties" on page 121, "sun-web-app" on page 124, "webservice-endpoint" on page 134 (`sun-web.xml`)

Subelements

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

TABLE A–66 `property` Subelements

Element	Required	Description
"description" on page 61	zero or one	Specifies an optional text description of a property.

Attributes

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

TABLE A–67 `property` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the property.
`value`	none	Specifies the value of the property.

Example

```
<property name="reapIntervalSeconds" value="20" />
```

property (with subelements)

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

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

Superelements

"cmp-resource" on page 54, "schema-generator-properties" on page 113, "webservice-endpoint" on page 134 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–68 `property` subelements

Element	Required	Description
"name" on page 94	only one	Specifies the name of the property.
"value" on page 132	only one	Specifies the value of the property.

Example

```
<property>
   <name>use-unique-table-names</name>
   <value>true</value>
</property>
```

Q

query-filter

Specifies the query filter for the CMP 1.1 finder.

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

Superelements

"finder" on page 69 (`sun-ejb-jar.xml`)

Subelements

none - contains data

query-method

Specifies a query method.

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

Superelements

"prefetch-disabled" on page 100 (`sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `query-method` element.

TABLE A–69 `query-method` Subelements

Element	Required	Description
"method-name" on page 93	only one	Specifies a method name.
"method-params" on page 94	only one	Specifies the fully qualified Java type names of method parameters.

query-ordering

Specifies the query ordering for the CMP 1.1 finder.

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

Superelements

"finder" on page 69 (`sun-ejb-jar.xml`)

Subelements

none - contains data

query-params

Specifies the query parameters for the CMP 1.1 finder.

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

Superelements

"finder" on page 69 (`sun-ejb-jar.xml`)

Subelements

none - contains data

query-variables

Specifies variables in the query expression for the CMP 1.1 finder.

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

Superelements

"finder" on page 69 (`sun-ejb-jar.xml`)

Subelements

none - contains data

R

realm

Specifies the name of the realm used to process all authentication requests associated with this application. If this element is not specified or does not match the name of a configured realm, the default realm is used. For more information about realms, see "Realm Configuration" in *Sun GlassFish Enterprise Server v3 Prelude Developer's Guide*.

Superelements

"as-context" on page 43, "login-config" on page 84 (`sun-ejb-jar.xml`)

Subelements

none - contains data

refresh-field

Specifies a field that gives the application component a programmatic way to refresh a cached entry.

Superelements

"cache-mapping" on page 49 (`sun-web.xml`)

Subelements

none

Attributes

The following table describes attributes for the `refresh-field` element.

TABLE A–70 `refresh-field` Attributes

Attribute	Default	Description
`name`	none	Specifies the input parameter name.
`scope`	`request.parameter`	(optional) Specifies the scope from which the input parameter is retrieved. Allowed values are `context.attribute`, `request.header`, `request.parameter`, `request.cookie`, `session.id`, and `session.attribute`.

refresh-period-in-seconds

Specifies the rate at which a read-only-bean must be refreshed from the data source. If the value is less than or equal to zero, the bean is never refreshed; if the value is greater than zero, the bean instances are refreshed at the specified interval. This rate is just a hint to the container. Default is 0 (no refresh).

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

none - contains data

removal-timeout-in-seconds

Specifies the amount of time a bean instance can remain idle in the container before it is removed (timeout). A value of 0 specifies that the container does not remove inactive beans automatically. The default value is 5400.

If `removal-timeout-in-seconds` is less than or equal to `cache-idle-timeout-in-seconds`, beans are removed immediately without being passivated.

Applies to stateful session beans.

For related information, see "cache-idle-timeout-in-seconds" on page 49.

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

Superelements

"bean-cache" on page 44 (`sun-ejb-jar.xml`)

Subelements

none - contains data

remote-home-impl

Specifies the fully-qualified class name of the generated `EJBHome impl` class.

Note – This value is automatically generated by the server at deployment or redeployment time. Do not specify it or change it after deployment.

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

Superelements

"gen-classes" on page 71 (`sun-ejb-jar.xml`)

Subelements

none - contains data

remote-impl

Specifies the fully-qualified class name of the generated `EJBObject impl` class.

Note – This value is automatically generated by the server at deployment or redeployment time. Do not specify it or change it after deployment.

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

Superelements

"gen-classes" on page 71 (`sun-ejb-jar.xml`)

Subelements

none - contains data

request-protection

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

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

Superelements

"message-security" on page 90 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none

Attributes

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

TABLE A–71 `request-protection` 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`.

required

Specifies whether the authentication method specified in the "auth-method" on page 43 element must be used for client authentication. The value is `true` or `false` (the default).

Superelements

"as-context" on page 43 (`sun-ejb-jar.xml`)

Subelements

none - contains data

res-ref-name

Specifies the `res-ref-name` in the corresponding Java EE deployment descriptor file `resource-ref` entry. The `res-ref-name` element specifies the name of a resource manager connection factory reference. The name must be unique within an enterprise bean.

Superelements

"resource-ref" on page 111 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

resize-quantity

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

Specifies the number of bean instances to be:

- Created, if a request arrives when the pool has less than "steady-pool-size" on page 120 quantity of beans (applies to pools only for creation). If the pool has more than `steady-pool-size` minus "resize-quantity" on page 109 of beans, then `resize-quantity` is still created.
- Removed, when the "pool-idle-timeout-in-seconds" on page 98 timer expires and a cleaner thread removes any unused instances.
  - For caches, when "max-cache-size" on page 86 is reached, `resize-quantity` beans are selected for passivation using the "victim-selection-policy" on page 132. In addition, the "cache-idle-timeout-in-seconds" on page 49 or "removal-timeout-in-seconds" on page 106 timers passivate beans from the cache.
  - For pools, when the "max-pool-size" on page 87 is reached, `resize-quantity` beans are selected for removal. In addition, the "pool-idle-timeout-in-seconds" on page 98 timer removes beans until `steady-pool-size` is reached.

Values are from 0 to MAX_INTEGER. The pool is not resized below the `steady-pool-size`. Default is 16.

Applies to stateless session beans, entity beans, and message-driven beans.

For EJB pools, the value can be defined in the EJB container. Default is 16.

For EJB caches, the value can be defined in the EJB container. Default is 32.

For message-driven beans, the value can be defined in the EJB container. Default is 2.

Superelements

"bean-cache" on page 44, "bean-pool" on page 45 (`sun-ejb-jar.xml`)

Subelements

none - contains data

resource-adapter-mid

Specifies the module ID of the resource adapter that is responsible for delivering messages to the message-driven bean.

Superelements

"mdb-resource-adapter" on page 88 (`sun-ejb-jar.xml`)

Subelements

none - contains data

resource-env-ref

Maps the `res-ref-name` in the corresponding Java EE deployment descriptor file `resource-env-ref` entry to the absolute `jndi-name` of a resource.

Superelements

"sun-web-app" on page 124 (`sun-web.xml`), "ejb" on page 63 (`sun-ejb-jar.xml`))

Subelements

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

TABLE A–72 `resource-env-ref` Subelements

Element	Required	Description
"resource-env-ref-name" on page 110	only one	Specifies the `res-ref-name` in the corresponding Java EE deployment descriptor file `resource-env-ref` entry.
"jndi-name" on page 76	only one	Specifies the absolute `jndi-name` of a resource.

Example

```
<resource-env-ref>
   <resource-env-ref-name>jms/StockQueueName</resource-env-ref-name>
   <jndi-name>jms/StockQueue</jndi-name>
</resource-env-ref>
```

resource-env-ref-name

Specifies the `res-ref-name` in the corresponding Java EE deployment descriptor file `resource-env-ref` entry.

Superelements

"resource-env-ref" on page 110 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

resource-ref

Maps the `res-ref-name` in the corresponding Java EE deployment descriptor file `resource-ref` entry to the absolute `jndi-name` of a resource.

Note – Connections acquired from JMS connection factories are not shareable in the current release of the Enterprise Server. The `res-sharing-scope` element in the `ejb-jar.xml` file `resource-ref` element is ignored for JMS connection factories.

When `resource-ref` specifies a JMS connection factory for the Sun GlassFish Message Queue, the `default-resource-principal` (name/password) must exist in the Message Queue user repository.

Superelements

"sun-web-app" on page 124 (`sun-web.xml`), "ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–73 `resource-ref` Subelements

Element	Required	Description
"res-ref-name" on page 108	only one	Specifies the `res-ref-name` in the corresponding Java EE deployment descriptor file `resource-ref` entry.
"jndi-name" on page 76	only one	Specifies the absolute `jndi-name` of a resource.
"default-resource-principal" on page 61	zero or one	Specifies the default principal (user) for the resource.

Example

```
<resource-ref>
   <res-ref-name>jdbc/EmployeeDBName</res-ref-name>
   <jndi-name>jdbc/EmployeeDB</jndi-name>
</resource-ref>
```

response-protection

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

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

Superelements

"message-security" on page 90 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none

Attributes

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

TABLE A–74 `response-protection` 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`.

role-name

Contains the `role-name` in the `security-role` element of the corresponding Java EE deployment descriptor file.

Superelements

"security-role-mapping" on page 115 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

S

sas-context

Describes the sas-context fields.

Superelements

"ior-security-config" on page 73 (`sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `sas-context` element.

TABLE A–75 `sas-context` Subelements

Element	Required	Description
"caller-propagation" on page 51	only one	Specifies whether the target accepts propagated caller identities. The values are NONE, SUPPORTED, or REQUIRED.

schema-generator-properties

Specifies field-specific column attributes in `property` subelements.

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

Superelements

"cmp-resource" on page 54 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–76 `schema-generator-properties` Subelements

Element	Required	Description
"property (with subelements)" on page 102	zero or more	Specifies a property name and value.

Properties

The following table describes properties for the `schema-generator-properties` element.

TABLE A–77 `schema-generator-properties` Properties

Property	Default	Description
`use-unique-table-names`	`false`	Specifies that generated table names are unique within each application server domain.
bean-name.*field-name*.*attribute*	none	Defines a column attribute. For attribute descriptions, see Table A–78.

The following table lists the column attributes for properties defined in the `schema-generator-properties` element.

TABLE A–78 `schema-generator-properties` Column Attributes

Attribute	Description
`jdbc-type`	Specifies the JDBC type of the column created for the CMP field. The actual SQL type generated is based on this JDBC type but is database vendor specific.
`jdbc-maximum-length`	Specifies the maximum number of characters stored in the column corresponding to the CMP field. Applies only when the actual SQL that is generated for the column requires a length. For example, a `jdbc-maximum-length` of `32` on a CMP `String` field such as `firstName` normally results in a column definition such as VARCHAR(32). But if the `jdbc-type` is `CLOB` and you are deploying on Oracle, the resulting column definition is CLOB. No length is given, because in an Oracle database, a CLOB has no length.
`jdbc-precision`	Specifies the maximum number of digits stored in a column which represents a numeric type.
`jdbc-scale`	Specifies the number of digits stored to the right of the decimal point in a column that represents a floating point number.
`jdbc-nullable`	Specifies whether the column generated for the CMP field allows null values.

Example

```
<schema-generator-properties>
   <property>
         <name>Employee.firstName.jdbc-type</name>
         <value>char</value>
   </property>
   <property>
         <name>Employee.firstName.jdbc-maximum-length</name>
         <value>25</value>
   </property>
   <property>
         <name>use-unique-table-names</name>
         <value>true</value>
   </property>
</schema-generator-properties>
```

security-role-mapping

Maps roles to users or groups in the currently active realm. See "Realm Configuration" in *Sun GlassFish Enterprise Server v3 Prelude Developer's Guide*.

The role mapping element maps a role, as specified in the EJB JAR `role-name` entries, to a environment-specific user or group. If it maps to a user, it must be a concrete user which exists in the current realm, who can log into the server using the current authentication method. If it maps to a group, the realm must support groups and the group must be a concrete group which exists in the current realm. To be useful, there must be at least one user in that realm who belongs to that group.

Superelements

"sun-web-app" on page 124 (`sun-web.xml`), "sun-ejb-jar" on page 123 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–79 `security-role-mapping` Subelements

Element	Required	Description
"role-name" on page 112	only one	Contains the `role-name` in the `security-role` element of the corresponding Java EE deployment descriptor file.
"principal-name" on page 101	one or more if no `group-name`, otherwise zero or more	Contains a principal (user) name in the current realm. In an enterprise bean, the principal must have the run-as role specified.
"group-name" on page 71	one or more if no `principal-name`, otherwise zero or more	Contains a group name in the current realm.

service-endpoint-interface

Specifies the web service reference name relative to `java:comp/env`.

Superelements

"port-info" on page 99 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

service-impl-class

Specifies the name of the generated service implementation class.

Superelements

"service-ref" on page 116 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

service-qname

Specifies the WSDL service element that is being referred to.

Superelements

"service-ref" on page 116 (`sun-web.xml`, `sun-ejb-jar.xml`); "webservice-endpoint" on page 134 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

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

TABLE A–80 `service-qname` subelements

Element	Required	Description
"namespaceURI" on page 95	only one	Specifies the namespace URI.
"localpart" on page 83	only one	Specifies the local part of a QNAME.

service-ref

Specifies runtime settings for a web service reference. Runtime information is only needed in the following cases:

- To define the port used to resolve a container-managed port
- To define the default Stub/Call property settings for Stub objects
- To define the URL of a final WSDL document to be used instead of the one associated with the `service-ref` in the standard Java EE deployment descriptor

Superelements

"sun-web-app" on page 124 (`sun-web.xml`), "ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–81 `service-ref` subelements

Element	Required	Description
"service-ref-name" on page 117	only one	Specifies the web service reference name relative to `java:comp/env`.
"port-info" on page 99	zero or more	Specifies information for a port within a web service reference.
"call-property" on page 50	zero or more	Specifies JAX-RPC property values that can be set on a `javax.xml.rpc.Call` object before it is returned to the web service client.
"wsdl-override" on page 135	zero or one	Specifies a valid URL pointing to a final WSDL document.
"service-impl-class" on page 116	zero or one	Specifies the name of the generated service implementation class.
"service-qname" on page 116	zero or one	Specifies the WSDL service element that is being referenced.

service-ref-name

Specifies the web service reference name relative to `java:comp/env`.

Superelements

"service-ref" on page 116 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

servlet

Specifies a principal name for a servlet. Used for the `run-as` role defined in `web-xml`.

Superelements

"sun-web-app" on page 124 (`sun-web.xml`)

Subelements

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

TABLE A–82 `servlet` Subelements

Element	Required	Description
"servlet-name" on page 118	only one	Contains the name of a servlet, which is matched to a `servlet-name` in `web.xml`.
"principal-name" on page 101	zero or one	Contains a principal (user) name in the current realm.
"webservice-endpoint" on page 134	zero or more	Specifies information about a web service endpoint.

servlet-impl-class

Specifies the automatically generated name of the servlet implementation class.

Superelements

"webservice-endpoint" on page 134 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

servlet-name

Specifies the name of a servlet, which is matched to a `servlet-name` in `web.xml`. This name must be present in `web.xml`.

Superelements

"cache-mapping" on page 49, "servlet" on page 117 (`sun-web.xml`)

Subelements

none - contains data

session-config

Specifies session configuration information. Overrides the web container settings for an individual web module.

Superelements

"sun-web-app" on page 124 (`sun-web.xml`)

Subelements

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

TABLE A–83 `session-config` Subelements

Element	Required	Description
"session-manager" on page 119	zero or one	Specifies session manager configuration information.
"session-properties" on page 120	zero or one	Specifies session properties.
"cookie-properties" on page 57	zero or one	Specifies session cookie properties.

session-manager

Specifies session manager information.

Superelements

"session-config" on page 118 (`sun-web.xml`)

Subelements

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

TABLE A–84 `session-manager` Subelements

Element	Required	Description
"manager-properties" on page 84	zero or one	Specifies session manager properties.
"store-properties" on page 121	zero or one	Specifies session persistence (storage) properties.

Attributes

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

TABLE A–85 `session-manager` Attributes

Attribute	Default	Description
`persistence-type`	`memory`	(optional) Specifies the session persistence mechanism. Allowed values are `memory`, `file`, and `replicated`. **Note –** The `replicated` persistence type is not supported in GlassFish v3 Prelude.

session-properties

Specifies session properties.

Superelements

"session-config" on page 118 (`sun-web.xml`)

Subelements

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

TABLE A–86 `session-properties` Subelements

Element	Required	Description
"property (with attributes)" on page 101	zero or more	Specifies a property, which has a name and a value.

Properties

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

TABLE A–87 `session-properties` Properties

Property	Default	Description
`timeoutSeconds`	`1800`	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 `timeoutSeconds` value. If neither `session-timeout` nor `timeoutSeconds` is specified, the `timeoutSeconds` default is used. Note that the `session-timeout` element in `web.xml` is specified in minutes, not seconds.
`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 JSP.

steady-pool-size

Specifies the initial and minimum number of bean instances that are maintained in the pool. Default is 32. Applies to stateless session beans and message-driven beans.

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

Superelements

"bean-pool" on page 45 (`sun-ejb-jar.xml`)

Subelements

none - contains data

store-properties

Specifies session persistence (storage) properties.

Superelements

"session-manager" on page 119 (`sun-web.xml`)

Subelements

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

TABLE A–88 `store-properties` Subelements

Element	Required	Description
"property (with attributes)" on page 101	zero or more	Specifies a property, which has a name and a value.

Properties

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

TABLE A–89 `store-properties` Properties

Property	Default	Description
`directory`	*domain-dir*`/generated/jsp/`*app-name*`/`*module-name*`_war`	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 module. Applicable only if the `persistence-type` attribute of the parent "session-manager" on page 119 element is `file`.

TABLE A–89 `store-properties` Properties *(Continued)*

Property	Default	Description
`persistenceScope`	`session`	Specifies how much of the session state is stored. Allowed values are as follows: ■ `session` - The entire session state is stored every time. This mode provides the best guarantee that your session data is correctly stored for any distributable web module. ■ `modified-session` - The entire session state is stored if it has been modified. A session is considered to have been modified if `HttpSession.setAttribute()` or `HttpSession.removeAttribute()` was called. You must guarantee that `setAttribute()` is called every time an attribute is changed. This is not a Java EE specification requirement, but it is required for this mode to work properly. ■ `modified-attribute` - Only modified session attributes are stored. For this mode to work properly, you must follow some guidelines, which are explained immediately following this table. Applicable only if the `persistence-type` attribute of the parent "session-manager" on page 119 element is `replicated`. **Note –** The `replicated` persistence type is not supported in GlassFish v3 Prelude.

If the `persistenceScope` store property is set to `modified-attribute`, a web module must follow these guidelines:

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

stub-property

Specifies JAX-RPC property values that are set on a `javax.xml.rpc.Stub` object before it is returned to the web service client. The property names can be any properties supported by the JAX-RPC `Stub` implementation.

Superelements

"port-info" on page 99 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

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

TABLE A–90 `stub-property` subelements

Element	Required	Description
"name" on page 94	only one	Specifies the name of the entity.
"value" on page 132	only one	Specifies the value of the entity.

Properties

The following table describes properties for the `stub-property` element.

TABLE A–91 `stub-property` properties

Property	Default	Description
`jbi-enabled`	`true`	Determines whether the visibility of this endpoint as a Java Business Integration service is enabled or disabled.

Example

```
<service-ref>
<service-ref-name>service/FooProxy</service-ref-name>
  <port-info>
       <service-endpoint-interface>a.FooPort</service-endpoint-interface>
       <wsdl-port>
          <namespaceURI>urn:Foo</namespaceURI>
          <localpart>FooPort</localpart>
       </wsdl-port>
       <stub-property>
          <name>javax.xml.rpc.service.endpoint.address</name>
          <value>http://localhost:8080/a/Foo</value>
       </stub-property>
  </port-info>
</service-ref>
```

sun-ejb-jar

Defines the Enterprise Server specific configuration for an EJB JAR file. This is the root element; there can only be one `sun-ejb-jar` element in a `sun-ejb-jar.xml` file. See "The sun-ejb-jar.xml File" on page 36.

Superelements

none

Subelements

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

TABLE A–92 `sun-ejb-jar` Subelements

Element	Required	Description
"security-role-mapping" on page 115	zero or more	Maps a role in the corresponding Java EE XML file to a user or group.
"enterprise-beans" on page 67	only one	Describes all the runtime properties for an EJB JAR file in the application.

sun-web-app

Defines Enterprise Server specific configuration for a web module. This is the root element; there can only be one `sun-web-app` element in a `sun-web.xml` file. See "The sun-web.xml File" on page 33.

Superelements

none

Subelements

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

TABLE A–93 `sun-web-app` Subelements

Element	Required	Description
"context-root" on page 57	zero or one	Contains the web context root for the web application.
"security-role-mapping" on page 115	zero or more	Maps roles to users or groups in the currently active realm.
"servlet" on page 117	zero or more	Specifies a principal name for a servlet, which is used for the `run-as` role defined in `web.xml`.
"idempotent-url-pattern" on page 72	zero or more	Specifies a URL pattern for idempotent requests.
"session-config" on page 118	zero or one	Specifies session manager, session cookie, and other session-related information.
"ejb-ref" on page 66	zero or more	Maps the absolute JNDI name to the `ejb-ref` in the corresponding Java EE XML file.

TABLE A–93 `sun-web-app` Subelements *(Continued)*

Element	Required	Description
"resource-ref" on page 111	zero or more	Maps the absolute JNDI name to the `resource-ref` in the corresponding Java EE XML file.
"resource-env-ref" on page 110	zero or more	Maps the absolute JNDI name to the `resource-env-ref` in the corresponding Java EE XML file.
"service-ref" on page 116	zero or more	Specifies runtime settings for a web service reference.
"message-destination-ref" on page 90	zero or more	Specifies the name of a physical message destination.
"cache" on page 46	zero or one	Configures caching for web application components.
"class-loader" on page 52	zero or one	Specifies class loader configuration information.
"jsp-config" on page 76	zero or one	Specifies JSP configuration information.
"locale-charset-info" on page 81	zero or one	Deprecated. Use the `parameter-encoding` subelement of `sun-web-app` instead.
"parameter-encoding" on page 96	zero or one	Determines the default request character encoding and how the web container decodes parameters from forms according to a hidden field value.
"property (with attributes)" on page 101	zero or more	Specifies a property, which has a name and a value.
"message-destination" on page 89	zero or more	Specifies the name of a logical message destination.
"webservice-description" on page 133	zero or more	Specifies a name and optional publish location for a web service.

Attributes

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

TABLE A–94 `sun-web-app` Attributes

Attribute	Default	Description
`error-url`	(blank)	(optional) Not implemented. Do not use.
`httpservlet-security-provider`	none	(optional) Specifies the `HttpServlet` message layer provider that the web container's servlet `auth-constraint` processing calls.

Properties

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

TABLE A–95 `sun-web-app` Properties

Property	Default	Description
`allowLinking`	`false`	If `true`, resources in this web application that are symbolic links are served. You can also define this property for a virtual server. Web applications on the virtual server that do not define this property use the virtual server's value. For details, see "virtual-server" in *Sun GlassFish Enterprise Server v3 Prelude Administration Reference*. **Caution –** Setting this property to `true` on Windows systems exposes JSP source code.

TABLE A–95 `sun-web-app` 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 all the web applications on a specific virtual server. For details, see "virtual-server" in *Sun GlassFish Enterprise Server v3 Prelude Administration Reference*.

TABLE A–95 `sun-web-app` Properties *(Continued)*

Property	Default	Description

`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 all the web applications on a specific virtual server. For details, see "virtual-server" in *Sun GlassFish Enterprise Server v3 Prelude Administration Reference*.

`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`, `org.apache.catalina.LifecycleListener`, or `org.apache.catalina.InstanceListener` interface. For example:
		`<property name="listener_1"`
		`value="org.glassfish.extension.MyLifecycleListener"/>`
		You can set this property for all the web applications on a specific virtual server. For details, see "virtual-server" in *Sun GlassFish Enterprise Server v3 Prelude Administration Reference*.

`crossContextAllowed`	`true`	If `true`, allows this web application to access the contexts of other web applications using the `ServletContext.getContext()` method.
`relativeRedirectAllowed`	`false`	If `true`, allows this web application to send a relative URL to the client using `HttpServletResponse.sendRedirect()`, and instructs the web container not to translate any relative URLs to fully qualified ones.
`reuseSessionID`	`false`	If `true`, sessions generated for this web application use the session ID specified in the request.
`securePagesWithPragma`	`true`	Set this property to `false` to ensure that for this web application file downloads using SSL work properly in Internet Explorer. You can set this property for all the web applications on a specific virtual server. For details, see "virtual-server" in *Sun GlassFish Enterprise Server v3 Prelude Administration Reference*.
`singleThreaded ServletPoolSize`	5	Specifies the maximum number of servlet instances allocated for each `SingleThreadModel` servlet in the web application.

TABLE A–95 `sun-web-app` Properties *(Continued)*

Property	Default	Description
`tempdir`	*domain-dir*`/generated/`*app-name* or *domain-dir*`/generated/`*module-name*	Specifies a temporary directory for use by this web module. This value is used to construct the value of the `javax.servlet.context.tempdir` context attribute. Compiled JSP files are also placed in this directory.
`useResponse CTForHeaders`	`false`	If `true`, response headers are encoded using the response's charset instead of the default (UTF-8).

T

tie-class

Specifies the automatically generated name of a tie implementation class for a port component.

Superelements

"webservice-endpoint" on page 134 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

timeout

Specifies the "cache-mapping" on page 49 specific maximum amount of time in seconds that an entry can remain in the cache after it is created or refreshed. If not specified, the default is the value of the `timeout` attribute of the "cache" on page 46 element.

Superelements

"cache-mapping" on page 49 (`sun-web.xml`)

Subelements

none - contains data

Attributes

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

TABLE A–96 `timeout` Attributes

Attribute	Default	Description
`name`	none	Specifies the timeout input parameter, whose value is interpreted in seconds. The field's type must be `java.lang.Long` or `java.lang.Integer`.
`scope`	`request.attribute`	(optional) Specifies the scope from which the input parameter is retrieved. Allowed values are `context.attribute`, `request.header`, `request.parameter`, `request.cookie`, `request.attribute`, and `session.attribute`.

transport-config

Specifies the security transport information.

Superelements

"ior-security-config" on page 73 (`sun-ejb-jar.xml`)

Subelements

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

TABLE A–97 `transport-config` Subelements

Element	Required	Description
"integrity" on page 73	only one	Specifies if the target supports integrity-protected messages. The values are NONE, SUPPORTED, or REQUIRED.
"confidentiality" on page 55	only one	Specifies if the target supports privacy-protected messages. The values are NONE, SUPPORTED, or REQUIRED.
"establish-trust-in-target" on page 69	only one	Specifies if the target is capable of authenticating *to* a client. The values are NONE, SUPPORTED, or REQUIRED.
"establish-trust-in-client" on page 69	only one	Specifies if the target is capable of authenticating a client. The values are NONE, SUPPORTED, or REQUIRED.

transport-guarantee

Specifies that the communication between client and server is `NONE`, `INTEGRAL`, or `CONFIDENTIAL`.

- `NONE` means the application does not require any transport guarantees.
- `INTEGRAL` means the application requires that the data sent between client and server be sent in such a way that it can't be changed in transit.

- `CONFIDENTIAL` means the application requires that the data be transmitted in a fashion that prevents other entities from observing the contents of the transmission.

In most cases, a value of `INTEGRAL` or `CONFIDENTIAL` indicates that the use of SSL is required.

Superelements

"webservice-endpoint" on page 134 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

U

unique-id

Contains the unique ID for the application. This value is automatically updated each time the application is deployed or redeployed. Do not edit this value.

Superelements

"enterprise-beans" on page 67 (`sun-ejb-jar.xml`)

Subelements

none - contains data

url-pattern

Specifies a servlet URL pattern for which caching is enabled. See the Servlet 2.4 specification section SRV. 11.2 for applicable patterns.

Superelements

"cache-mapping" on page 49 (`sun-web.xml`)

Subelements

none - contains data

use-thread-pool-id

Specifies the thread pool from which threads are selected for remote invocations of this bean.

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

Superelements

"ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

none - contains data

V

value

Specifies the value of the entity.

Superelements

"call-property" on page 50, "stub-property" on page 122 (`sun-web.xml`, `sun-ejb-jar.xml`); "property (with subelements)" on page 102 (`sun-ejb-jar.xml`)

Subelements

none - contains data

victim-selection-policy

Specifies how stateful session beans are selected for passivation. Possible values are First In, First Out (`FIFO`), Least Recently Used (`LRU`), Not Recently Used (`NRU`). The default value is `NRU`, which is actually pseudo-LRU.

Note – You cannot plug in your own victim selection algorithm.

The victims are generally passivated into a backup store (typically a file system or database). This store is cleaned during startup, and also by a periodic background process that removes idle entries as specified by `removal-timeout-in-seconds`. The backup store is monitored by a background thread (or sweeper thread) to remove unwanted entries.

Applies to stateful session beans.

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

Superelements

"bean-cache" on page 44 (`sun-ejb-jar.xml`)

Subelements

none - contains data

Example

```
<victim-selection-policy>LRU</victim-selection-policy>
```

If both SSL2 and SSL3 are enabled, the server tries SSL3 encryption first. If that fails, the server tries SSL2 encryption. If both SSL2 and SSL3 are enabled for a virtual server, the server tries SSL3 encryption first. If that fails, the server tries SSL2 encryption.

W

webservice-description

Specifies a name and optional publish location for a web service.

Superelements

"sun-web-app" on page 124 (`sun-web.xml`), "enterprise-beans" on page 67 (`sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `webservice-description` element.

TABLE A–98 `webservice-description` subelements

Element	Required	Description
"webservice-description-name" on page 134	only one	Specifies a unique name for the web service within a web or EJB module.
"wsdl-publish-location" on page 136	zero or one	Specifies the URL of a directory to which a web service's WSDL is published during deployment.

webservice-description-name

Specifies a unique name for the web service within a web or EJB module.

Superelements

"webservice-description" on page 133 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

webservice-endpoint

Specifies information about a web service endpoint.

Superelements

"servlet" on page 117 (`sun-web.xml`), "ejb" on page 63 (`sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `webservice-endpoint` element.

TABLE A–99 `webservice-endpoint` subelements

Element	Required	Description
"port-component-name" on page 99	only one	Specifies a unique name for a port component within a web or EJB module.
"endpoint-address-uri" on page 67	zero or one	Specifies the automatically generated endpoint address.
"login-config" on page 84	zero or one	Specifies the authentication configuration for an EJB web service endpoint.
"message-security-binding" on page 91	zero or one	Specifies a custom authentication provider binding.
"transport-guarantee" on page 130	zero or one	Specifies that the communication between client and server is `NONE`, `INTEGRAL`, or `CONFIDENTIAL`.
"service-qname" on page 116	zero or one	Specifies the WSDL service element that is being referenced.
"tie-class" on page 129	zero or one	Specifies the automatically generated name of a tie implementation class for a port component.
"servlet-impl-class" on page 118	zero or one	Specifies the automatically generated name of the generated servlet implementation class.

TABLE A–99 `webservice-endpoint` subelements *(Continued)*

Element	Required	Description
"debugging-enabled" on page 60	zero or one	Specifies whether the debugging servlet is enabled for this web service endpoint. Allowed values are `true` and `false` (the default).
"property (with attributes)" on page 101 (`sun-web.xml`) "property (with subelements)" on page 102 (`sun-ejb-jar.xml`)	zero or more	Specifies a property, which has a name and a value.

wsdl-override

Specifies a valid URL pointing to a final WSDL document. If not specified, the WSDL document associated with the `service-ref` in the standard Java EE deployment descriptor is used.

Superelements

"service-ref" on page 116 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

Example

```
// available via HTTP
<wsdl-override>http://localhost:8000/myservice/myport?WSDL</wsdl-override>

// in a file
<wsdl-override>file:/home/user1/myfinalwsdl.wsdl</wsdl-override>
```

wsdl-port

Specifies the WSDL port.

Superelements

"port-info" on page 99 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

The following table describes subelements for the `wsdl-port` element.

TABLE A–100 `wsdl-port` subelements

Element	Required	Description
"namespaceURI" on page 95	only one	Specifies the namespace URI.
"localpart" on page 83	only one	Specifies the local part of a QNAME.

wsdl-publish-location

Specifies the URL of a directory to which a web service's WSDL is published during deployment. Any required files are published to this directory, preserving their location relative to the module-specific WSDL directory (`META-INF/wsdl` or `WEB-INF/wsdl`).

Superelements

"webservice-description" on page 133 (`sun-web.xml`, `sun-ejb-jar.xml`)

Subelements

none - contains data

Example

Suppose you have an `ejb.jar` file whose `webservices.xml` file's `wsdl-file` element contains the following reference:

```
META-INF/wsdl/a/Foo.wsdl
```

Suppose your `sun-ejb-jar` file contains the following element:

```
<wsdl-publish-location>file:/home/user1/publish</wsdl-publish-location>
```

The final WSDL is stored in `/home/user1/publish/a/Foo.wsdl`.

Index

D

E

M

N

O

P

Q

R

S

T

U

V

W

X