Eclipse GlassFish Upgrade Guide, Release 7

Eclipse GlassFish

Upgrade Guide

Release 7

This guide explains how to upgrade to Eclipse GlassFish 7 from previous Eclipse GlassFish and Sun GlassFish Enterprise Server product releases. Also included in this guide are instructions for upgrading configuration data and Jakarta EE applications from binary-compatible earlier versions of this software to work with Eclipse GlassFish 7. Finally, this guide describes compatibility issues that affect data and applications that are to be migrated.

Eclipse GlassFish Upgrade Guide, Release 7

Preface

This guide explains how to upgrade to Eclipse GlassFish from previous of Eclipse GlassFish and Sun GlassFish Enterprise Server product releases. This guide also includes instructions for upgrading configuration data and Jakarta EE applications from binary-compatible earlier versions of this software to work with Eclipse GlassFish. Finally, this guide describes compatibility issues that affect data and applications that are to be migrated.

1 Upgrading an Installation of Application Server or Eclipse GlassFish

This section explains how to upgrade to Eclipse GlassFish 7 from a previous version. The process involves replicating your existing configuration and applications in the new installation.

Key Topics

- Upgrade Paths
- Upgrade Procedure
- Special Cases (Clusters, Node Agents, NSS)
- Troubleshooting

Upgrade Paths

Choose one of the following upgrade paths:

Side-by-Side (Recommended)

Installs the new version in a separate directory, allowing you to test before switching to production.
Steps:

1. Install Eclipse GlassFish 7 in a new directory.
2. Manually copy the configuration from the old installation.
3. Test the new installation.
4. Update your production environment to use the new installation.

In-Place (Not Recommended)

Replaces the existing installation. Requires manual steps and significant downtime.
Steps:

1. Move the current installation to a backup directory.
2. Install Eclipse GlassFish 7 in the original location.
3. Manually copy the configuration from the backup.
4. Test the new installation. If issues arise, restore the backup.

Upgrade Procedure

Use the Upgrade Tool (`asadmin start-domain --upgrade`) to migrate configurations and applications.

Prerequisites

- Stop all domains on the source server.

- Install Eclipse GlassFish 7 (see Eclipse GlassFish Installation Guide).

- Copy custom/third-party libraries from the old installation to the new one.

Steps

1. **Install Eclipse GlassFish** 7 in a new directory.

2. **Copy configurations** from the old installation:

   - Domain directory (e.g., `glassfish/domains/domain1`)

   - Nodes directory (if applicable, e.g., `glassfish/nodes`)

3. **Run the Upgrade Tool**: `asadmin start-domain --upgrade``

4. **Start the upgraded domain**: `asadmin start-domain domain-name``

5. **Verify the upgrade** by logging into the Admin Console.

Special Cases

Upgrade of Clusters

When upgrading from a clustered configuration, the older cluster information is retained in a new `domain.xml` file in the Eclipse GlassFish 7 installation directories. However, it is still necessary to manually re-create the server instances that are contained in the clusters.

To do that:

1. Perform new (not upgrade) Eclipse GlassFish 7 installations on each node host. Eclipse GlassFish 7 installation instructions are provided in the [Eclipse GlassFish Installation Guide](#).

2. Correct the node configuration on the upgraded DAS, if necessary. This procedure is described in [Correcting Cluster configuration](#).

3. Start the upgraded DAS.

   ```
   asadmin> start-domain domain-name
   ```

   If the upgrade succeeded, the migrated cluster configuration exists and the `get-health` subcommand lists the status of the clustered instances as not running.

4. Confirm that the cluster configuration exists and contains all its instances.

   ```
   asadmin> get-health cluster-name
   ```

   For example, for the sample `cluster1` used in this procedure:

   ```
   asadmin> get-health cluster1
   instance1 not started
   instance2 not started
   ```

```
Command get-health executed successfully.
```

5. Re-create the clustered server instances on each instance host.
   The specific commands to use depend on your configuration.

   ◦ If remote hosts cannot contact the DAS, export and import the instances' configuration data,
     as explained in " To Resynchronize an Instance and the DAS Offline" in Eclipse GlassFish
     High Availability Administration Guide.

   ◦ If remote hosts can contact the DAS, create each instance individually and resynchronize the
     instance with the DAS, as explained in the following sections:

     ▪ " To Create an Instance Locally" in Eclipse GlassFish High Availability Administration
       Guide

     ▪ " To Resynchronize an Instance and the DAS Online" in Eclipse GlassFish High
       Availability Administration Guide
       Note that the node name matches that used for the node agent in the 2.x installation. If
       you get an error stating that some attributes do not match the values in the DAS
       configuration, follow the instructions in Correcting Cluster configuration.

6. After creating the instances, manually copy the instance-dir/imq directory for each instance
   from the older source installation to the target Eclipse GlassFish 7 installation.

7. If necessary, start the cluster.
   For example:

   ```
   asadmin> start-cluster cluster1
   ```

   This step may or may not be necessary, depending on the procedure you used to create the
   server instances for the cluster.

Example 2-3 Creating Two Local Instances

The following example shows how to create two local instances in a cluster.

```
host1$ asadmin --host dashost create-local-instance --node na1 --cluster cluster1
instance1
host2$ asadmin --host dashost create-local-instance --node na2 --cluster cluster1
instance2
```

dashost
   The name of the DAS host.

na1
   The name of the node host.

cluster1
   The name of the cluster.

instance1, instance2
    The names of the instances.

Correcting Cluster configuration

Eclipse GlassFish 7 does not support node agents. If you're upgrading from a version that supports them or you have issues in cluster configuration after an upgrade, after upgrading the DAS:

1. **Install Eclipse GlassFish 7 on each node host**.
2. **Correct node configurations** using `update-node-config` or `update-node-ssh`.
3. **Re-create clusters and instances**:
    - Use `create-local-instance` for each instance.
    - Copy the `imq` directory from the old installation.

Upgrading Installations Using NSS Cryptographic Tokens

Eclipse GlassFish 7 does not support NSS. If you're upgrading from a version that supports them, follow these steps:

1. **Prepare for Upgrade**:
    - Install Eclipse GlassFish 7 in a new directory.
    - Copy the source domain to the new installation.
    - Update `domain.xml` to remove NSS references and add JKS keystore paths.
2. **Perform Post-Upgrade Configuration**:
    - Migrate NSS keys to PKCS#12 using `keytool` and `certutil`.
    - Update the master password if needed.
3. **Upgrade PKCS#11 Hardware Tokens**:
    - Configure the token using JDK-JSSE mechanisms.
    - Update `domain.xml` to reference the hardware token.

Troubleshooting

Cluster Profile Security Setting

If upgrading from Application Server 9.1 or Enterprise Server v2, ensure the `admin-service` element in `domain.xml` has:

```
security-enabled=false
```

when `type=das-and-server`.

Common Issues

- **Cluster Profile Upgrade on Windows**: Ensure file paths use double backslashes (`\\`).

- **Upgrade Verification**: Check logs for errors and verify deployed applications.

Terminology

Source Domain Directory

Directory of the domain being upgraded (e.g., `c:\glassfish\domains\domain1`).

Target Root Domain's Directory

Directory where domains are created in the new installation (e.g., `c:\glassfish7\glassfish\domains`).

Master Password

SSL certificate database password (default: `changeit`).

2 Eclipse GlassFish Upgrade Compatibility Issues

This section describes some compatibility issues between Eclipse GlassFish 7 and earlier product releases. This section also describes some compatibility issues that affect Java applications that run on earlier product releases with which Eclipse GlassFish 7 is binary-compatible. When you upgrade to Eclipse GlassFish 7, you must address these issues.

The following topics are addressed here:

- Binary-Compatible Releases For Eclipse GlassFish 7
- New Default Installation Directory
- Changes to Group Management Service Settings
- Application Client Interoperability
- Node Agent Support
- HADB and `hadbm` Command Support
- Command Line Interface: The `asadmin` Command
- Applications That Use Java DB
- Applications That Use Persistence
- HTTP Service to Network Service Changes
- NSS Cryptographic Token Support

Binary-Compatible Releases For Eclipse GlassFish 7

Eclipse GlassFish 7 is NOT binary-compatible with the earlier releases of the software:

- Sun GlassFish Enterprise Server v2.1.1 (Enterprise and Developer Profiles)
- Sun GlassFish Enterprise Server v3
- GlassFish Server Open Source Edition 3.0.1
- GlassFish Server Open Source Edition 3.1
- GlassFish Server Open Source Edition 3.1.1
- GlassFish Server Open Source Edition 4.x
- GlassFish Server Open Source Edition 5.x
- Eclipse GlassFish 6.x

Java applications that run on these releases also work on Eclipse GlassFish 7 except for the compatibility issues that are listed in the remainder of this chapter.

> The compatibility issues that are listed in the remainder of this chapter do not affect Java applications that run on Sun GlassFish Enterprise Server v3 and Eclipse

> GlassFish 3.0.1. The differences between Eclipse GlassFish 7 and the Enterprise Server v3 releases do not affect applications and data.

New Default Installation Directory

The default Eclipse GlassFish 7 installation directories are as follows:

Solaris, Linux, and Mac OS X systems

```
user-home-directory/glassfish7
```

Windows systems

```
SystemDrive\glassfish7
```

Changes to Group Management Service Settings

The functionality of the Group Management Service (GMS) has not changed since Sun GlassFish Enterprise Server v2.1.1, but the names of GMS settings have been changed in the Administration Console to make them more understandable. These changes are made automatically during the upgrade process.

Changes to settings on the Edit Group Management Service page in the Administration Console are summarized in the following table.

Table 1-1 GMS Administration Console Settings Changes from 2.1.1 to 4.0

Old Setting Name	New Setting Name
Protocol Maximum Trial	Maximum Missed Heartbeats
Protocol Timeout	Heartbeat Frequency
Ping Timeout	Group Discovery Timeout
Verified Timeout	Failure Verification Wait Time

The Merge Protocol settings from Sun GlassFish Enterprise Server v2.1.1 are not supported and have been removed.

Application Client Interoperability

The Jakarta EE 10 platform specification imposes stricter requirements than Java EE 5 and older did on which JAR files can be visible to various modules within an EAR file. In particular, application clients must not have access to EJB JAR files or other JAR files in the EAR file unless they use a `Class-Path` header in the manifest file, or unless references use the standard Java SE mechanisms (extensions, for example), or use the Jakarta EE `library-directory` mechanism. Deployed Java EE 5 applications that are upgraded to Eclipse GlassFish 7 will have the `compatibility` property set to `v2`

and will run without change on Eclipse GlassFish 7. You may, however, want to consider modifying the applications to conform to Jakarta EE 10 requirements.

If your upgrade includes a deployed application with an application client, you will need to retrieve the client stubs using Eclipse GlassFish 7 in order to run the client. Use the `asadmin get-client-stubs` command.

If you try to run the application client before retrieving the client stubs, you will see the following error message:

```
Invalid or corrupt jarfile jar-file-name
```

If you commonly distribute application clients to remote systems from which users will run them, you must not only retrieve the client stubs, but you must also run the `package-appclient` utility for Eclipse GlassFish 7 to upgrade the Eclipse GlassFish system files. This utility creates a JAR file, which you can then expand on the remote systems.

Application clients use EJBs, web services, or other enterprise components that are in the application server (on the server side). The application client and the application server must use the same version and implementation of the RMI-IIOP protocol. Eclipse GlassFish 7 does not support communication between different versions of the protocol implementation. You cannot run application clients with one version of the application server runtime with a server that has a different version. Most often, this would happen if you upgraded the server but had not upgraded all the application client installations. If you run the `package-appclient` utility, this issue will not arise.

You can use the Java Web Start support to distribute and launch the application client. If the runtime on the server has changed since the end-user last used the application client, Java Web Start automatically retrieves the updated runtime. Java Web Start enables you to keep the clients and servers synchronized and using the same runtime.

Node Agent Support

Eclipse GlassFish 7 does not support node agents. When updating from installations of earlier product versions in which node agents were configured, the cluster definitions will be migrated, but the clustered instances themselves must be manually re-created. See Correcting Cluster configuration for more information.

HADB and `hadbm` Command Support

Eclipse GlassFish 7 does not support HADB or the `hadbm` management command.

Instead of HADB, Eclipse GlassFish 7 supports high availability clustering by means of in-memory session state replication and ActiveCache for GlassFish. See " High Availability in Eclipse GlassFish" in Eclipse GlassFish High Availability Administration Guide for more information.

Command Line Interface: The `asadmin` Command

The following sections describe changes to the command line utility `asadmin`:

- Deprecated `asadmin` Subcommands

- Deprecated, Unsupported, and Obsolete Options

For more information about `asadmin` and its subcommands, see the Eclipse GlassFish Reference Manual.

Deprecated `asadmin` Subcommands

In Eclipse GlassFish 7, it is recommended that utility options of the `asadmin` command precede the subcommand. Utility options are options that control the behavior of the `asadmin` utility, as distinguished from subcommand options. Use of the following options after the subcommand is deprecated and will be removed in Eclipse GlassFish 7.1.0.

- `--host`

- `--port`

- `--user`

- `--passwordfile`

- `--terse`

- `--secure`

- `--echo`

- `--interactive`

Deprecated, Unsupported, and Obsolete Options

Options in Table 1-2 are deprecated or no longer supported, or are obsolete and are ignored.

Table 1-2 Deprecated, Unsupported, and Obsolete Options for `asadmin` and Subcommands

Option	Affected Subcommands
`--acceptlang`	Unsupported for the `create-virtual-server` subcommand.
`--acls`	Unsupported for the `create-virtual-server` subcommand.
`--adminpassword`	Unsupported for all relevant subcommands. Use `--passwordfile` instead.
`--autoapplyenabled`	Obsolete for the `create-http-lb` subcommand.
`--autohadb`	Obsolete for the `create-cluster` subcommand.
`--autohadboverride`	Obsolete for the `start-cluster` subcommand and the `stop-cluster` subcommand
`--blockingenabled`	Unsupported for the `create-http-listener` subcommand.

Option	Affected Subcommands
`--configfile`	Unsupported for the `create-virtual-server` subcommand.
`--defaultobj`	Unsupported for the `create-virtual-server` subcommand.
`--defaultvs`	Deprecated for the `create-http-listener` subcommand. Use `--default-virtual-server` instead.
`--description`	Obsolete for the `restore-domain` subcommand.
`--devicesize`	Obsolete for the `create-cluster` subcommand.
`--haadminpassword`	Obsolete for the `create-cluster` subcommand.
`--haadminpasswordfile`	Obsolete for the `create-cluster` subcommand.
`--haagentport`	Obsolete for the `create-cluster` subcommand.
`--haproperty`	Obsolete for the `create-cluster` subcommand.
`--heartbeataddress`	Deprecated for the `create-cluster` subcommand. Use `--multicastaddress` instead.
`--heartbeatport`	Deprecated for the `create-cluster` subcommand. Use `--multicastport` instead.
`--hosts`	Obsolete for the `create-cluster` subcommand.
`--ignoreDescriptorItem`	Replaced by the all lowercase option `--ignoredescriptoritem` in the `set-web-context-param` subcommand and the `set-web-env-entry` subcommand.
`--mime`	Unsupported for the `create-virtual-server` subcommand.
`--password`	Unsupported for all remote subcommands. Use `--passwordfile` instead.
`--path`	Unsupported for the `create-domain` subcommand. Use `--domaindir` instead.
`--portbase`	Obsolete only for the `create-cluster` subcommand. This option is still valid in other subcommands such as `create-domain`, `create-instance`, and `create-local-instance`.
`--resourcetype`	Unsupported for all relevant subcommands. Use `--restype` instead.
`--retrievefile`	Obsolete for the `export-http-lb-config` subcommand.
`--setenv`	Obsolete for the `start-instance` subcommand.

Option	Affected Subcommands
`--target`	Obsolete only for the following subcommands: `create-connector-connection-pool` `create-resource-adapter-config` `delete-connector-connection-pool` `delete-connector-security-map` `delete-jdbc-connection-pool` `delete-resource-ref` Replaced by an operand in the `list-custom-resources` subcommand and the `list-jndi-entries` subcommand.

Applications That Use Java DB

The directory location of Java DB in Eclipse GlassFish 7 has changed from its location in previous installations. Suppose that you have deployed applications that use Java DB databases in your previous server installation, and you upgrade your existing installation to Eclipse GlassFish 7. If you run the `asadmin start-database` command and successfully start Java DB, you could run into problems while trying to run applications that were deployed on your previous server installation.

To solve this problem, you can copy the `databases` directory from your previous installation to as-install/`databases`. Make sure the database is not running when you do this.

Alternatively, you can perform these steps:

1. Use the `asadmin start-database` command with the `--dbhome` option pointing to the `databases` directory in the older version of Java DB. For example:

   ```
   asadmin start-database --dbhome c:\glassfish\databases
   ```

2. After upgrade, start Eclipse GlassFish 7.

Applications That Use Persistence

Eclipse GlassFish 7 and 3.0.1, and Sun GlassFish Enterprise Server v3 use the persistence provider EclipseLink, while earlier versions used TopLink Essentials.

An application that uses the container to create an `EntityManager` or `EntityManagerFactory` and that used Toplink Essentials as its provider will work in Eclipse GlassFish 7. The container creates an `EntityManager` if the application uses the `@PersistenceContext` annotation to inject an `EntityManager`, as in the following example:

```
@PersistenceContext
```

```
EntityManager em;
```

The container creates an `EntityManagerFactory` if the application uses the `@PersistenceUnit` annotation to inject an `EntityManagerFactory`, as in the following example:

```
@PersistenceUnit
EntityManagerFactory emf;

EntityManager em = emf.createEntityManager();
```

When the application is loaded, Eclipse GlassFish 7 will translate the provider to EclipseLink and will also translate `toplink.*` properties in the `persistence.xml` to corresponding EclipseLink properties. (The actual `persistence.xml` file remains unchanged.)

Under certain circumstances, however, you may have to modify the `persistence.xml` file or your code:

- If your application uses Java SE code to create the `EntityManagerFactory`, you will need to change your `persistence.xml` file for both the `provider` element and for any `toplink.*` properties to use the EclipseLink equivalents. An application uses Java SE code if it uses the `javax.persistence.Persistence` class to create the `EntityManagerFactory`, as in the following example:

  ```
  EntityManagerFactory emf =
      javax.persistence.Persistence.createEntityManagerFactory("Order");
  EntityManager em = emf.createEntityManager();
  ```

  In this case, change the `provider` element to specify the following:

  ```
  <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
  ```

- If the application itself contains any TopLink Essentials-specific code and therefore contains casts to `oracle.toplink.*`, you must change the code to cast to `org.eclipse.persistence.*`. You can use the package renamer tool described on the [Eclipse wiki](#) to do this. This tool is not provided with Eclipse GlassFish 7, however, so you must obtain it from the EclipseLink project download site.

HTTP Service to Network Service Changes

In Eclipse GlassFish 7, most HTTP Service settings are defined in the Network Service configuration that was introduced in Sun GlassFish Enterprise Server v3.

The changes are described in the following sections.

- [Changes to Dotted Names](#)

Changes to Dotted Names

The dotted name hierarchy for the HTTP Service configuration in Eclipse GlassFish 7 is shown below. Elements that are no longer supported are `request-processing`, `keep-alive`, `connection-pool`, `http-protocol`, `http-file-cache`, and `http-listener`. During the upgrade process, these discontinued elements are remapped to the new configuration automatically and then deleted.

```
config
    http-service
        access-log
        request-processing
        keep-alive
        connection-pool
        http-protocol
        http-file-cache
        http-listener
            ssl
            property
        virtual-server
            http-access-log
            property
        property
    thread-pools
        thread-pool
```

The dotted name hierarchy for the Eclipse GlassFish 7 Network Service and HTTP Service configurations is shown below. The `network-config` element and all its children are new except for `ssl`.

```
config
    network-config
        transports
            selection-key-handler
            transport
        protocols
            protocol
                http
                    file-cache
                port-unification
                    protocol-finder
                protocol-chain-instance-handler
                    protocol-chain
                protocol-filter
                ssl
```

```
        network-listeners
            network-listener
    http-service
        access-log
        virtual-server
            http-access-log
            property
        property
    thread-pools
        thread-pool
```

The following example compares the commands for setting a listener port for Sun GlassFish Enterprise Server v3 and Eclipse GlassFish 7. Note that the configuration for Enterprise Server v3 also applies to all earlier Enterprise Server 2.x releases.

- Command for Sun GlassFish Enterprise Server v3 and earlier:

  ```
  asadmin set server-config.http-service.http-listener.http-1.listenerport=4321
  ```

- Command for Eclipse GlassFish 7:

  ```
  asadmin set server-config.network-config.network-listeners.network-\
  listener.http-1.listenerport=4321
  ```

Changes to `asadmin` Subcommands

To accommodate the move of HTTP Service into the new Network Service configuration, `asadmin` subcommands are changed as follows:

- The `create-ssl` subcommand has a new `--type` parameter value, `network-listener`.

- The `create-virtual-server` SUBcommand has a new parameter, `--networklisteners`.

- The `create-http-listener` subcommand adds a `network-listener` element to the domain configuration. The syntax and options of this commands are unchanged.

Remapping of HTTP Service Attributes and Properties

The following tables describe how attributes and properties in the HTTP Service configuration for Eclipse GlassFish 7 are remapped to attributes in the Network Service configuration for older product releases. If you use a configuration from a Sun GlassFish Enterprise Server v2 or v3 release, this remapping happens automatically and then discontinued elements are deleted.

Table 1-3 `com.sun.grizzly` Property Remapping

`com.sun.grizzly` **Property**	**New Owning Element**	**New Attribute Name**
`selector.timeout`	`transport`	`selector-poll-timeout-millis`
`displayConfiguration`	`transport`	`display-configuration`

com.sun.grizzly **Property**	New Owning Element	New Attribute Name
enableSnoop	transport	snoop-enabled
readTimeout	transport	read-timeout-millis
writeTimeout	transport	write-timeout-millis

Table 1-4 connection-pool Attribute Remapping

connection-pool **Attribute**	New Owning Element	New Attribute Name
queue-size-in-bytes	thread-pool	max-queue-size
max-pending-count	transport	max-connections-count
receive-buffer-size-in- bytes	http	request-body-buffer-size- bytes
send-buffer-size-in-bytes	http	send-buffer-size-bytes

Table 1-5 http-file-cache Attribute Remapping

http-file-cache **Attribute**	New Owning Element	New Attribute Name
file-caching-enabled	file-cache	enabled
max-age-in-seconds	file-cache	max-age-seconds
medium-file-space-in-bytes	file-cache	max-cache-size-bytes
max-files-count	file-cache	max-files-count
globally-enabled	none	not supported
medium-file-size-limit-in-bytes	none	not supported
small-file-size-limit-in-bytes	none	not supported
small-file-space-in-bytes	none	not supported
file-transmission-enabled	none	not supported
hash-init-size	none	not supported

Table 1-6 http-listener Attribute Remapping

http-listener **Attribute**	New Owning Element	New Attribute Name
id	network-listener	name
address	network-listener	address
port	network-listener	port
enabled	network-listener	enabled
acceptor-threads	transport	acceptor-threads
security-enabled	protocol	security-enabled
default-virtual-server	http	default-virtual-server
server-name	http	server-name
redirect-port	http	redirect-port

http-listener **Attribute**	New Owning Element	New Attribute Name
xpowered-by	http	xpowered-by
external-port	none	not supported
family	none	not supported
blocking-enabled	none	not supported

Table 1-7 http-listener Property Remapping

http-listener **Property**	New Owning Element	New Attribute Name
maxKeepAliveRequests	http	max-connections
authPassthroughEnabled	http	auth-pass-through-enabled
compression	http	compression
compressableMimeType	http	compressable-mime-type
noCompressionUserAgents	http	no-compression-user-agents
compressionMinSize	http	compression-min-size-bytes
restrictedUserAgents	http	restricted-user-agents
cometSupport	http	comet-support-enabled
connectionUploadTimeout	http	connection-upload-timeout- millis
disableUploadTimeout	http	upload-timeout-enabled
chunkingDisabled	http	chunking-enabled
uriEncoding	http	uri-encoding
traceEnabled	http	trace-enabled
rcmSupport	http	rcm-support-enabled
jkEnabled	network-listener	jk-enabled
crlFile	ssl	crl-file
trustAlgorithm	ssl	trust-algorithm
trustMaxCertLength	ssl	trust-max-cert-length-bytes
tcpNoDelay	transport	tcp-no-delay
bufferSize	transport	buffer-size-bytes
use-nio-direct-bytebuffer	transport	byte-buffer-type
proxyHandler	none	not supported
proxiedProtocols	none	not supported
recycle-objects	none	not supported
reader-threads	none	not supported
acceptor-queue-length	none	not supported
reader-queue-length	none	not supported
connectionTimeout	none	not supported

http-listener **Property**	New Owning Element	New Attribute Name
monitoring-cache-enabled	none	not supported
monitoring-cache-refresh-in- millis	none	not supported
ssl-cache-entries	none	not supported
ssl3-session-timeout	none	not supported
ssl-session-timeout	none	not supported

Table 1-8 http-protocol Attribute Remapping

http-protocol **Attribute**	New Owning Element	New Attribute Name
version	http	version
forced-response-type	http	forced-response-type
default-response-type	http	default-response-type
dns-lookup-enabled	none	not supported
ssl-enabled	none	not supported

Table 1-9 http-service Property Remapping

http-service **Property**	New Owning Element	New Attribute or Property Name
accessLoggingEnabled	http-service, virtual-server	access-logging-enabled attribute
ssl-cache-entries	http-service	unchanged property
ssl3-session-timeout	http-service	unchanged property
ssl-session-timeout	http-service	unchanged property
proxyHandler	http-service	unchanged property
connectionTimeout	http-service	unchanged property
all other properties	none	not supported

Table 1-10 keep-alive Attribute Remapping

keep-alive **Attribute**	New Owning Element	New Attribute Name
max-connections	http	max-connections
timeout-in-seconds	http	timeout-seconds
thread-count	none	not supported

Table 1-11 request-processing Attribute Remapping

`request-processing` **Attribute**	New Owning Element	New Attribute Name
`thread-count`	`thread-pool`	`max-thread-pool-size`
`initial-thread-count`	`thread-pool`	`min-thread-pool-size`
`header-buffer-length-in-bytes`	`http`	`header-buffer-length-bytes`
`request-timeout-in-seconds`	`http`	`request-timeout-seconds`
`thread-increment`	none	not supported

Table 1-12 `ssl` Attribute Changes

Previous Attribute or Property	Previous Owning Element	New `ssl` Attribute
none	none	`key-store`
none	none	`trust-store`
`crlFile` property	`http-listener`	`crl-file`
`trustAlgorithm` property	`http-listener`	`trust-algorithm`
`trustMaxCertLength` property	`http-listener`	`trust-max-cert-length-bytes`
all other `ssl` attributes	`ssl`	unchanged

Table 1-13 `thread-pool` Attribute Changes

Previous Attribute	Previous Owning Element	New `thread-pool` Attribute
none	none	`classname`
none	none	`max-queue-size`
`thread-pool-id`	`thread-pool`	`name`
`idle-thread-timeout-in-seconds`	`thread-pool`	`idle-thread-timeout-seconds`
`num-work-queues`	`thread-pool`	not supported
all other `thread-pool` attributes	`thread-pool`	unchanged

Table 1-14 `virtual-server` Attribute Changes

Previous Attribute or Property	Previous Owning Element	New `virtual-server` Attribute
`http-listeners` attribute	`virtual-server`	`network-listeners`
`accessLoggingEnabled` property	`http-service`	`access-logging-enabled`
`sso-enabled` property	`virtual-server`	`sso-enabled`
`ssoCookieSecure` property	`virtual-server`	`sso-cookie-secure`
all other `virtual-server` attributes	`virtual-server`	unchanged

Previous Attribute or Property	Previous Owning Element	New `virtual-server` Attribute
all other `virtual-server` properties	`virtual-server`	unchanged, still properties

New Network Service Elements and Attributes

The following tables describe the Network Service elements and attributes that were introduced in Sun GlassFish Enterprise Server v3. For attributes and properties remapped from discontinued elements to new elements, see Remapping of HTTP Service Attributes and Properties.

The new `file-cache` element has no new attributes. All of its attributes are remapped from the `http-file-cache` element. For details, see Table 1-5.

Table 1-15 New `http` Attributes

Attribute	Default	Description
`adapter`	`com.sun.grizzly.tcp.StaticResourcesAdapter`	(Optional) Specifies the class name of the static resources adapter.
`max-post-size-bytes`	`2097152`	(Optional) Specifies the maximum size of `POST` actions.

For remapped `http` attributes, see Table 1-4, Table 1-6, Table 1-7, Table 1-8, Table 1-10, and Table 1-11.

Table 1-16 New `network-listener` Attributes

Attribute	Default	Description
`protocol`	none	Specifies the `name` of the `protocol` associated with this `network-listener`. Although this attribute is required, a `protocol` is automatically created with the same `name` as the `network-listener` when you use `asadmin create-http-listener` to create a `network-listener`.
`thread-pool`	none	(Optional) Specifies the `name` of the `thread-pool` associated with this `network-listener`.
`transport`	none	Specifies the `name` of the `transport` associated with this `network-listener`. Although this attribute is required, the default `transport` is used when you use `asadmin create-http-listener` to create a `network-listener`.

For remapped `network-listener` attributes, see Table 1-6.

Table 1-17 New `port-unification` Attributes

Attribute	Default	Description
`name`	none	Specifies a unique name for the `port-unification`.

Attribute	Default	Description
classname	none	Specifies the class name of the port-unification implementation.

Table 1-18 New protocol Attributes

Attribute	Default	Description
name	none	Specifies a unique name for the protocol.

For remapped protocol attributes, see Table 1-6.

Table 1-19 New protocol-chain Attributes

Attribute	Default	Description
name	none	Specifies a unique name for the protocol-chain.
classname	none	Specifies the class name of the protocol-chain implementation.
type	STATELESS	Specifies the type of protocol chain.

Table 1-20 New protocol-chain-instance-handler Attributes

Attribute	Default	Description
name	none	Specifies a unique name for the protocol-chain-instance-handler.
classname	none	Specifies the class name of the protocol-chain-instance-handler implementation.

Table 1-21 New protocol-filter Attributes

Attribute	Default	Description
name	none	Specifies a unique name for the protocol-filter.
classname	none	Specifies the class name of the protocol-filter implementation.

Table 1-22 New protocol-finder Attributes

Attribute	Default	Description
name	none	Specifies a unique name for the protocol-finder.
classname	none	Specifies the class name of the protocol-finder implementation.
protocol	none	Specifies the name of the protocol associated with this protocol-finder.

Table 1-23 New selection-key-handler Attributes

Attribute	Default	Description
`name`	none	Specifies a unique name for the `selection-key-handler`.
`classname`	none	Specifies the class name of the `selection-key-handler` implementation.

Table 1-24 New `ssl` Attributes

Attribute	Default	Description
`key-store`	none	(Optional) Specifies a key store.
`trust-store`	none	(Optional) Specifies a trust store.

For remapped `ssl` attributes, see Table 1-12.

Table 1-25 New `thread-pool` Attributes

Attribute	Default	Description
`classname`	`com.sun.grizzly.http.StatsThreadPool`	(Optional) Specifies the class name of the `thread-pool` implementation.
`max-queue-size`	`-1`	(Optional) Specifies the maximum number of messages that can be queued until threads are available to process them. A value of `-1` specifies no limit.

For remapped `thread-pool` attributes, see Table 1-4, Table 1-11, and Table 1-13.

Table 1-26 New `transport` Attributes

Attribute	Default	Description
`name`	none	Specifies a unique name for the `transport`.
`classname`	`com.sun.grizzly.TCPSelectorHandler`	(Optional) Specifies the class name of the `transport` implementation.
`selection-key-handler`	none	(Optional) Specifies the `name` of the `selection-key-handler` associated with this `transport`.
`idle-key-timeout-seconds`	`30`	(Optional) Specifies the idle key timeout.

For remapped `transport` attributes, see Table 1-3, Table 1-4, Table 1-6, and Table 1-7.

NSS Cryptographic Token Support

Eclipse GlassFish 7 does not support Network Security Services (NSS) cryptographic tokens. When upgrading to Eclipse GlassFish 7 from Enterprise Server v2.x, additional manual configuration steps

must be performed. These steps are explained later in this guide, in Upgrading Installations That Use NSS Cryptographic Tokens.

Appendix

This section contains information about and conventions for the entire Eclipse GlassFish (Eclipse GlassFish) documentation set.

Eclipse GlassFish 7 is developed through the GlassFish project open-source community at https://github.com/eclipse-ee4j/glassfish. The GlassFish project provides a structured process for developing the Eclipse GlassFish platform that makes the new features of the Jakarta EE platform available faster, while maintaining the most important feature of Jakarta EE: compatibility. It enables Java developers to access the Eclipse GlassFish source code and to contribute to the development of the Eclipse GlassFish.

The following topics are addressed here:

- Eclipse GlassFish Documentation Set
- Related Documentation
- Typographic Conventions
- Symbol Conventions
- Default Paths and File Names

Eclipse GlassFish Documentation Set

The Eclipse GlassFish documentation set describes deployment planning and system installation. For an introduction to Eclipse GlassFish, refer to the books in the order in which they are listed in the following table.

Book Title	Description
Release Notes	Provides late-breaking information about the software and the documentation and includes a comprehensive, table-based summary of the supported hardware, operating system, Java Development Kit (JDK), and database drivers.
Quick Start Guide	Explains how to get started with the Eclipse GlassFish product.
Installation Guide	Explains how to install the software and its components.
Upgrade Guide	Explains how to upgrade to the latest version of Eclipse GlassFish. This guide also describes differences between adjacent product releases and configuration options that can result in incompatibility with the product specifications.
Deployment Planning Guide	Explains how to build a production deployment of Eclipse GlassFish that meets the requirements of your system and enterprise.

Book Title	Description
Administration Guide	Explains how to configure, monitor, and manage Eclipse GlassFish 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.
Security Guide	Provides instructions for configuring and administering Eclipse GlassFish security.
Application Deployment Guide	Explains how to assemble and deploy applications to the Eclipse GlassFish and provides information about deployment descriptors.
Application Development Guide	Explains how to create and implement Java Platform, Enterprise Edition (Jakarta EE platform) applications that are intended to run on the Eclipse GlassFish. These applications follow the open Java standards model for Jakarta EE components and application programmer interfaces (APIs). This guide provides information about developer tools, security, and debugging.
Add-On Component Development Guide	Explains how to use published interfaces of Eclipse GlassFish to develop add-on components for Eclipse GlassFish. This document explains how to perform only those tasks that ensure that the add-on component is suitable for Eclipse GlassFish.
Embedded Server Guide	Explains how to run applications in embedded Eclipse GlassFish and to develop applications in which Eclipse GlassFish is embedded.
High Availability Administration Guide	Explains how to configure Eclipse GlassFish to provide higher availability and scalability through failover and load balancing.
Performance Tuning Guide	Explains how to optimize the performance of Eclipse GlassFish.
Troubleshooting Guide	Describes common problems that you might encounter when using Eclipse GlassFish and explains how to solve them.
Error Message Reference	Describes error messages that you might encounter when using Eclipse GlassFish.
Reference Manual	Provides reference information in man page format for Eclipse GlassFish administration commands, utility commands, and related concepts.
Message Queue Release Notes	Describes new features, compatibility issues, and existing bugs for Open Message Queue.
Message Queue Technical Overview	Provides an introduction to the technology, concepts, architecture, capabilities, and features of the Message Queue messaging service.
Message Queue Administration Guide	Explains how to set up and manage a Message Queue messaging system.

Book Title	Description
Message Queue Developer's Guide for JMX Clients	Describes the application programming interface in Message Queue for programmatically configuring and monitoring Message Queue resources in conformance with the Java Management Extensions (JMX).
Message Queue Developer's Guide for Java Clients	Provides information about concepts and procedures for developing Java messaging applications (Java clients) that work with Eclipse GlassFish.
Message Queue Developer's Guide for C Clients	Provides programming and reference information for developers working with Message Queue who want to use the C language binding to the Message Queue messaging service to send, receive, and process Message Queue messages.

Related Documentation

The following tutorials explain how to develop Jakarta EE applications:

- Your First Cup: An Introduction to the Jakarta EE Platform. For beginning Jakarta EE programmers, this short tutorial explains the entire process for developing a simple enterprise application. The sample application is a web application that consists of a component that is based on the Enterprise JavaBeans specification, a JAX-RS web service, and a JavaServer Faces component for the web front end.

- The Jakarta EE Tutorial. This comprehensive tutorial explains how to use Jakarta EE platform technologies and APIs to develop Jakarta EE applications.

Javadoc tool reference documentation for packages that are provided with Eclipse GlassFish is available as follows.

- The Jakarta EE specifications and API specification is located at https://jakarta.ee/specifications/.
- The API specification for Eclipse GlassFish 7, including Jakarta EE platform packages and non-platform packages that are specific to the Eclipse GlassFish product, is located at https://glassfish.org/documentation.html.

For information about creating enterprise applications in the NetBeans Integrated Development Environment (IDE), see the NetBeans Documentation, Training & Support page.

For information about the Apache Derby database for use with the Eclipse GlassFish, see the Derby page.

The Jakarta EE Examples project is a collection of code examples for Jakarta EE. It's available from the repository (`https://github.com/eclipse-ee4j/jakartaee-examples`).

The GlassFish Samples project is a collection of sample applications that demonstrate a broad range of Jakarta EE technologies. The GlassFish Samples are are available from the repository (`https://github.com/eclipse-ee4j/glassfish-samples`).

Typographic Conventions

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

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.

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.

Placeholder	Description	Default Value
as-install	Represents the base installation directory for Eclipse GlassFish. In configuration files, as-install is represented as follows: `${com.sun.aas.installRoot}`	- Installations on the Oracle Solaris operating system, Linux operating system, and Mac OS operating system: user's-home-directory`/glassfish7/glassfish` - Installations on the Windows operating system: SystemDrive`:\glassfish7\glassfish`
as-install-parent	Represents the parent of the base installation directory for Eclipse GlassFish.	- Installations on the Oracle Solaris operating system, Linux operating system, and Mac operating system: user's-home-directory`/glassfish7` - Installations on the Windows operating system: SystemDrive`:\glassfish7`
domain-root-dir	Represents the directory in which a domain is created by default.	as-install`/domains/`
domain-dir	Represents the directory in which a domain's configuration is stored. In configuration files, domain-dir is represented as follows: `${com.sun.aas.instanceRoot}`	domain-root-dir/domain-name
instance-dir	Represents the directory for a server instance.	domain-dir/instance-name

List of Tables