Eclipse GlassFish Upgrade Guide, Release 7

Eclipse GlassFish

Upgrade Guide

Release 7

Contributed 2018 - 2024

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.

Note: The main thrust of the Eclipse GlassFish 7 release is to provide an application server for developers to explore and begin exploiting the new and updated technologies in the Jakarta EE 10 platform. Thus, the upgrade feature of Eclipse GlassFish was not a focus of this release. The feature is included in the release, but it may not function properly with some of the new features added in support of the Jakarta EE 10 platform.

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. 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. Finally, this guide describes compatibility issues that affect data and applications that are to be migrated.

> The main thrust of the Eclipse GlassFish 7 release is to provide an application server for developers to explore and begin exploiting the new and updated technologies in the Jakarta EE 10 platform. Thus, the upgrade feature of Eclipse GlassFish was not a focus of this release. The feature is included in the release, but it may not function properly with some of the new features added in support of the Jakarta EE 10 platform.

This preface 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.

Book Title	Description
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.
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.

Book Title	Description
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.
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 nonplatform packages that are specific to the Eclipse GlassFish product, is located at https://glassfish.org/docs/.

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

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

The Jakarta EE Samples project is a collection of sample applications that demonstrate a broad range of Jakarta EE technologies. The Jakarta EE Samples are bundled with the Jakarta EE Software Development Kit (SDK) and are also 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

1 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
```

In Eclipse GlassFish 3.0.1 and Enterprise Server v3, the default installation root directory was `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 6 platform specification imposes stricter requirements than Jakarta EE 5 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 Jakarta 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 6 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 Upgrading Clusters and Node Agent Configurations 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.

- `--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

Option	Affected Subcommands
`--blockingenabled`	Unsupported for the `create-http-listener` subcommand.
`--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.

2 Upgrading an Installation of Application Server or Eclipse GlassFish

🔥 | This chapter is obsoleted and must be revided.

The Upgrade Tool that is bundled with Eclipse GlassFish 7 replicates the configuration of a previously installed server in the target installation. The Upgrade Tool assists in upgrading the configuration and applications from an earlier version of the Application Server or Eclipse GlassFish to Eclipse GlassFish 7.

The Upgrade Tool is explained later in this chapter.

To view a list of the older versions from which you can upgrade, see Supported Releases for Upgrade to Eclipse GlassFish 7.

The following topics are addressed here:

- Upgrade Overview
- Performing a Side-By-Side Upgrade With Upgrade Tool
- Upgrading Installations That Use NSS Cryptographic Tokens
- Upgrading Clusters and Node Agent Configurations
- Correcting Potential Upgrade Problems

Upgrade Overview

The subsections that follow provide information that you will need when you perform an upgrade.

The following topics are addressed here:

- Upgrade Paths
- Upgrade Terminology
- Summary of Upgrade Tools and Procedures
- Supported Releases for Upgrade to Eclipse GlassFish 7
- Upgrading From Version 8.x or Older Product Releases
- Upgrading Eclipse GlassFish Inside a Closed Network

Upgrade Paths

There are two general paths you can use when upgrading to Eclipse GlassFish 7:

Side-by-Side

A side-by-side upgrade means that the new Eclipse GlassFish release is installed in a different directory than the release from which you are upgrading.
In this scenario, you perform the following steps:

1. Perform a basic installation of Eclipse GlassFish 7 in a location other than the one being used for the older product.

2. Copy the configuration from old installation to the new one. While initially the Upgrade Tool would copy the configuration from the old installation to the new one, currently, this must be done manually.

3. Test the new Eclipse GlassFish installation to make sure everything is working properly.

4. When you are satisfied that the new installation works properly, modify your production environment to use the new installation.

The side-by-side upgrade path is typically used for live production environments because it allows you to thoroughly test the new Eclipse GlassFish installation before bringing it into production.

For a more detailed overview, see Summary of Upgrade Tools and Procedures.

Upgrade Terminology

The following are important terms related to the upgrade process.

Source Domain Directory

The directory of the server domain from which you are upgrading to the new version (for example, `c:\glassfish\domains\domain1`).

Target Root Domain's Directory

The directory where domains are created on the server to which you are upgrading (for example, `c:\glassfish7\glassfish\domains`).

Master Password

The SSL certificate database password used in operations such as Eclipse GlassFish startup. This term refers to the master password of the installation from which you want to upgrade. You need to specify this password if you have changed it from the default value of `changeit`.

Summary of Upgrade Tools and Procedures

There are several tools you can use to upgrade from an earlier Eclipse GlassFish or Enterprise Server installation to Eclipse GlassFish 7. The general procedures for upgrading to Eclipse GlassFish 7 vary depending on which tool you use and the product version from which you are upgrading.

The following topics are addressed here:

- Summary of Tools for Performing an Upgrade
- Summary of Procedure for Upgrading With Upgrade Tool

Summary of the Upgrade Tool Performing an Upgrade

Currently, There is only one tool you can use to perform an upgrade to Eclipse GlassFish 7 as described below.

- Upgrade Tool

Upgrade Tool

The Eclipse GlassFish Upgrade Tool is tended solely for performing side-by-side upgrades from any compatible older product version to Eclipse GlassFish 7.

Upgrade Tool provides a number of features that aid in the migration of older configurations and applications to a new Eclipse GlassFish 7 installation. These features are described in more detail in Upgrade Tool Functionality.

In Eclipse GlassFish 7, the term *Upgrade Tool* refers to the asadmin start-domain --upgrade command of the ASAdmin CLI tool, which is installed in as-install/bin directory.

> Upgrade Tool is the only tool you can use when upgrading to Eclipse GlassFish 7 from product versions prior to Eclipse GlassFish 3.0.1 or Enterprise Server v3.

See Summary of Procedure for Upgrading With Upgrade Tool for an overview of the general procedure for performing an upgrade with Upgrade Tool.

Summary of Procedure for Upgrading With Upgrade Tool

The general procedure for using Upgrade Tool to perform an upgrade to Eclipse GlassFish 7 from any compatible older version of Eclipse GlassFish or Enterprise Server comprises the following steps:

1. Download Eclipse GlassFish 7 and perform a Standard Installation, as described in " To Install Eclipse GlassFish Using the Self-Extracting File" in Eclipse GlassFish Installation Guide.

2. Copy any custom or third-party libraries from the older installation to their corresponding locations in the new Eclipse GlassFish 7 installation directories. Note that you should only copy custom or third-party libraries here. Do not copy an libraries from the actual domain that will be upgraded.

3. Copy the configuration from the old installation to the new one,

   ◦ copy GlassFish domain directory, e.g. `glassfish/domains/domain1`

   ◦ copy the nodes directory if it exists, e.g. `glassfish/nodes`

4. Run the `asadmin start-domain --upgrade` command from the new Eclipse GlassFish 7 as-install/`bin` directory.

5. Start the new Eclipse GlassFish 7 DAS with the `asadmin start-domain` subcommand.

This procedure is described in more detail in Performing a Side-By-Side Upgrade With Upgrade Tool.

Supported Releases for Upgrade to Eclipse GlassFish 7

Upgrades to Eclipse GlassFish 7 are supported from the following earlier Eclipse GlassFish product releases:

- Sun GlassFish Enterprise Server v2.1.1

- Sun GlassFish Enterprise Server v3

- Eclipse GlassFish 3.0.1

- Eclipse GlassFish 3.1

- Eclipse GlassFish 3.1.1

Upgrading From Version 8.x or Older Product Releases

It is not possible to upgrade to Eclipse GlassFish 7 directly from Sun GlassFish Enterprise Server 8.x or older product releases.

To upgrade from a product release that is older than any of those listed in Supported Releases for Upgrade to Eclipse GlassFish 7, you must first upgrade your older product release to one of the releases that are supported for upgrade to Eclipse GlassFish 7.

For example, to upgrade from any Enterprise Server 8.x release, you first need to upgrade that older release to Enterprise Server 2.1.1. That is, your upgrade path would be as follows:

Enterprise Server 8.x ⇒ Enterprise Server 2.1.1 ⇒ Eclipse GlassFish 7

Sun GlassFish Enterprise Server 2.1.1 is available for download from the GlassFish Community Downloads (`http://glassfish.java.net/public/downloadsindex.html`) page. Instructions for upgrading to Enterprise Server 2.1.1 are provided in Sun GlassFish Enterprise Server 2.1.1 Upgrade Guide (`http://docs.oracle.com/cd/E19879-01/821-0180/index.html`).

After upgrading your older Enterprise Server installation to Enterprise Server 2.1.1, you can proceed normally with the instructions in this guide to complete the upgrade to Eclipse GlassFish 7.

Upgrading Eclipse GlassFish Inside a Closed Network

For instructions on upgrading a Eclipse GlassFish installation in an environment where Internet access is not available, see "Extending and Updating Eclipse GlassFish Inside a Closed Network" in Eclipse GlassFish Administration Guide.

Performing a Side-By-Side Upgrade With Upgrade Tool

This section explains how to use Upgrade Tool to perform a side-by-side upgrade to Eclipse GlassFish 7 from any compatible older product release.

The following topics are addressed here:

- Upgrade Tool Summary

- Upgrade Tool Functionality

- To Upgrade From the Command Line Using Upgrade Tool

Upgrade Tool Summary

The Upgrade Tool upgrades your domain configurations and deployed applications. When you use

the Upgrade Tool, the source server and the target server are normally installed on the same machine, but under different install locations. Both server file systems must be accessible from the system on which you perform the upgrade.

To perform the upgrade, the user who runs the upgrade needs to have read permissions for the source and target directories and write permission for the target directory.

You can perform an upgrade using Upgrade Tool in the following ways:

- [To Upgrade From the Command Line Using Upgrade Tool](#)

Upgrade Tool Functionality

The Upgrade Tool migrates the configurations and deployed applications from an earlier version of Eclipse GlassFish to the current version. Database migrations or conversions are not part of this upgrade process.

Briefly, the Upgrade Tool performs the following steps:

- Copies the older source domain directory to the new target `domains` directory.
- Calls the `asadmin start-domain --upgrade` command to migrate the source configurations to the new target Eclipse GlassFish installation.
- Sends all `asadmin start-domain --upgrade` command output to the screen and to the `upgrade.log` file, and sends all server output to the `server.log` file.

Additional Upgrade Tool functions are explained in the following sections:

- [Migration of Deployed Applications](#)
- [Upgrade of Clusters](#)
- [Upgrade Verification](#)

Migration of Deployed Applications

Application archives (EAR files) and component archives (JAR, WAR, and RAR files) that are deployed in the source server do not require any modification to run on Eclipse GlassFish 7. Some components that may have incompatibilities are deployed on Eclipse GlassFish 7 with the `compatibility` property set to `v2` and will run without change on Eclipse GlassFish 7. It's, however, strongly encouraged to modify the applications to conform to Jakarta EE 10 requirements because the compatibility layer is not being maintained and will likely not cover all incompatible applications.

Jakarta EE 10 platform specification has stricter requirements than Java EE 5 and earlier version had 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. Setting the `library-directory` property to `v2` removes these restrictions in Eclipse GlassFish 7.

Applications and components that are deployed in the source server are deployed on the target

server during the upgrade. Applications that do not deploy successfully on the target server must be deployed manually on the target server by the user.

If a domain contains information about a deployed application and the installed application components do not agree with the configuration information, the configuration is migrated unchanged, without any attempt to reconfigure the incorrect configurations.

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. This procedure is explained in Upgrading Clusters and Node Agent Configurations.

Upgrade Verification

An upgrade log records the upgrade activity. The upgrade log file is named `upgrade.log` and is created in the working directory from which the Upgrade Tool is run. Additional information is recorded in the server log of the upgraded domain.

You can also use the `asadmin version` subcommand after starting the upgraded domain to verify the new Eclipse GlassFish product version; for example:

```
asadmin> version
Version = Eclipse GlassFish 7.0.0 (build 42)
Command version executed successfully.
```

To Upgrade From the Command Line Using Upgrade Tool

This procedure explains how to use the Upgrade Tool command line to upgrade to Eclipse GlassFish 7 from any supported older product release. See Supported Releases for Upgrade to Eclipse GlassFish 7 for a list of supported releases.

Before You Begin

Ensure that the domains on the source server from which you are upgrading are stopped before proceeding.

1. Download and install Eclipse GlassFish 7 using the Typical Installation path.
   See " Installing Eclipse GlassFish From a Self-Extracting Bundle" in Eclipse GlassFish Installation Guide for instructions.

2. Copy any custom or third-party libraries that may be located in the source as-install/`lib` directory to the target as-install/`lib` directory.
   Custom and third-party libraries should normally be located in the domain-dir/`lib` directory. This step is only necessary for custom or third-party libraries that may be located in the nonstandard as-install/`lib` directory.

3. Start Upgrade Tool from a command shell for your operating environment.

Use the Upgrade Tool that is located in the target Eclipse GlassFish 7 installation, not the older source installation.

```
asadmin start-domain --upgrade
[-s|--source source-domain-directory]
[-t|--target target-domain-directory]
[-f|--passwordfile password-file]
```

Explanations of these options are provided at the end of this procedure.

4. Follow the prompts to perform the upgrade.
   If a name used for an older domain that you are upgrading already exists in the new target domains directory, Upgrade Tool will ask if you want to rename the new directory so the old directory can be copied to the new installation.

   ◦ If you type `y` in response, the directory is renamed domain-name`.original`. If that name already exists, the directory will be renamed domain-name`.orginal.0`. For example, if the old domain directory is named `domain1`, it will be renamed `domain1.original`, or if that name already exists, `domain1.original.0`.

   ◦ If you type `n`, you are prompted to specify a different directory name or quit.

   The domain is upgraded and the results are output to the console.

5. Review the console output to verify that the upgrade proceeded correctly.
   This output is also written to the `output.log` file for later review.
   If there are any `SEVERE` or `WARNING` messages in the `server.log` file, the upgrade output will say `"Possible error encountered during upgrade. See server log after upgrade process completes."`

6. Start the upgraded Eclipse GlassFish 7 domain.

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

Log in to the Administration Console with the user name and password you used in the older server.

Eclipse GlassFish 7 does not support NSS authentication. If you are upgrading from a Enterprise Profile configuration that uses NSS authentication, follow the procedure in Upgrading Installations That Use NSS Cryptographic Tokens.

7. If you are upgrading a clustered configuration or a configuration in which node agents were used, proceed with the instructions in Upgrading Clusters and Node Agent Configurations.

Upgrading Installations That Use NSS Cryptographic Tokens

Eclipse GlassFish v2.x EE (Enterprise Edition) uses Network Security Services (NSS) for cryptographic software tokens. Eclipse GlassFish 7 does not support NSS, so when performing an upgrade from v2.x EE to 7 additional manual configuration steps must be performed.

The following topics are addressed here:

- To Prepare for the Upgrade
- To Perform Post-Upgrade Configuration
- To Upgrade PKCS#11 Hardware Tokens

To Prepare for the Upgrade

This procedure explains how to prepare for modifying an NSS-based Eclipse GlassFish 2.x installation when upgrading to Eclipse GlassFish 7.

1. Download and install Eclipse GlassFish 7 using the Typical Installation path.
   Ensure that you install the new Eclipse GlassFish 7 product in a directory that is different than the one used for the older installation from which you are upgrading.
   See "Installing Eclipse GlassFish From a Self-Extracting Bundle" in Eclipse GlassFish Installation Guide for instructions.

2. Rename the new Eclipse GlassFish 7 domain-dir (the default is as-install`/domains/domain1`) to a name of your choice.
   In this procedure, `31domain` is used for the renamed Eclipse GlassFish 7 domain.

3. Copy the older source domain to be upgraded to the new Eclipse GlassFish 7 as-install`/domains` directory.
   In this procedure, `domain1` is used for the older source domain that is copied to the new Eclipse GlassFish 7 installation.

   > ℹ️ The remaining steps in this procedure are performed on the copy of your source domain that you created in this step, rather than on your original source domain. It is strongly recommended that you perform the Eclipse GlassFish 7 upgrade on a copy of your old domain rather than on the original.

4. Copy the `server.policy`, `keystore.jks`, and `cacerts.jks` files from the renamed `./31domain/config` directory to the `./domain1/config` directory to be upgraded.
   For example:

```
cp as-install/domains/31domain/config/server.policy as-
install/domains/domain1/config
cp as-install/domains/31domain/config/keystore.jks as-
install/domains/domain1/config
cp as-install/domains/31domain/config/cacerts.jks as-install/domains/domain1/config
```

This will overwrite the master password for `./domain1` with the password used in the `./31domain`.

5. Modify the `domain.xml` file for `./domain1`.

   1. Add the following `jvm-options` under `server-config` and `default-config`:

      ```
      -Djavax.net.ssl.keyStore=${com.sun.aas.instanceRoot}/config/keystore.jks
      -Djavax.net.ssl.trustStore=${com.sun.aas.instanceRoot}/config/cacerts.jks
      ```

   2. Remove the following `jvm-option` under `server-config` and `default-config`:

      ```
      -Dcom.sun.appserv.nss.db=${com.sun.aas.instanceRoot}/config
      ```

6. Upgrade `./domain1` by starting the DAS in the new Eclipse GlassFish 7 installation with the `--upgrade` option.

   ```
   as-install/bin/asadmin start-domain --upgrade domain1
   ```

   This upgrades the domain and then shuts down the DAS.

7. Start the upgraded DAS normally.

   ```
   as-install/bin/asadmin start-domain domain1
   ```

To Perform Post-Upgrade Configuration

These instructions explain the post-upgrade configuration steps that must be performed when upgrading from an NSS-based installation to Eclipse GlassFish 7.

Before You Begin

Before proceeding with this procedure, complete the procedure explained in To Prepare for the Upgrade.

1. Start the Eclipse GlassFish 7 domain, if it is not already running, and open the Eclipse GlassFish Admin Console in a browser window.
   The default URL is https://localhost:4848
   As part of the To Prepare for the Upgrade procedure, the default keystore with a default self-signed key-certificate pair with an alias named `s1as` and a keystore password `changeit` was copied into the v2.x domain before the upgrade.

2. If your default server alias in the NSS v2.x domain is not `s1as`, you can delete this entry using the following command:

   ```
   keytool -delete -keystore keystore.jks -storepass changeit -alias s1as
   keytool -delete -keystore cacerts.jks -storepass changeit -alias s1as
   ```

3. If the master password for the v2.x domain is not the default password `changeit`, you need to change the new keystore password to match the v2.x master password.

```
keytool -storepasswd -new v2-master-password \
-keystore keystore.jks -storepass changeit
keytool -storepasswd -new v2-master-password \
-keystore cacerts.jks -storepass changeit
```

4. Take note of all the `KeyEntries` that exist in your NSS database.

   These entries must be migrated to the `keystore.jks` in the Eclipse GlassFish 7 domain. The following command can be used to list all the `KeyEntries` in the NSS database:

```
certutil -L -d $AS_NSS_DB
```

   `AS_NSS_DB` should point to the `${com.sun.aas.instanceRoot}/config` for the 7 instance into which the v2.x domain was copied. The listing with the attribute combinations `u,u,u` are the `KeyEntries`.
   For example:

```
s1as u,u,u
```

   > To run the `certutil` command, your `LD_LIBRARY_PATH` must point to the directory containing NSS library and DLLs.

5. For each `PrivateKey-Certificate` pair (`KeyEntry`) that exists in the v2.x NSS database, use the following commands to export them from the NSS database and import them into the newly created `keystore.jks` file.
   Make sure you use the same alias when importing the `KeyEntry` into the JKS keystore. For example, if s1as is the only alias present in the NSS database, the following command can be used:

```
> pk12util -o /tmp/s1as_pk.p12 -n s1as -d $AS_NSS_DB
>keytool -importkeystore -srckeystore /tmp/s1as_pk.p12 -destkeystore \
${com.sun.aas.instanceRoot}/config/keystore.jks -srcstoretype PKCS12 \
-deststoretype JKS -srcstorepass v2-master-password \
-deststorepass v3-master-password -srcalias s1as \
-destalias s1as -srckeypass v2-master-password \
-destkeypass v3-master-password
```

   > The reference to v3-master-password could be the same as v2-master-password if you intend to retain the same master password for the 7 domain after upgrading from v2.x.

6. If the `s1as` alias represents a `KeyEntry` with a self-signed certificate, the self-signed certificate must be copied to the `truststore`.

```
>certutil -L -n s1as -r -d $AS_NSS_DB> /tmp/s1as.der>keytool -import -keystore
cacerts.jks -storepass v3-master-password \
-file /tmp/s1as.der -alias s1as
```

7. There is a rare chance that the 2.x NSS database has some CA (Certificate Authority) certificates that are absent in the default created `truststore`. In such cases, all aliases that are missing in the `truststore` (`cacerts.jks`) need to collected.

   1. `certutil -L -d $AS_NSS_DB`
      Example output:

      ```
      verisignc1g1 T,c,c
      verisignc1g2 T,c,c
      verisignc1g3 T,c,c
      ```

   2. `keytool -list -keystore cacerts.jks -storepass` v3-master-password
      Example output:

      ```
      godaddyclass2ca, Jan 20, 2005, trustedCertEntry,
      Certificate fingerprint (MD5): 91:DE:06:25:AB:DA:FD:32:17:0C:BB:25:17:2A:84:67
      verisignclass1g3ca, Mar 26, 2004, trustedCertEntry,
      Certificate fingerprint (MD5): B1:47:BC:18:57 1:18:A0:78:2D:EC:71:E8:2A:95:73
      secomevrootca1, May 1, 2008, trustedCertEntry,
      Certificate fingerprint (MD5): 22:2D:A6:01:EA:7C:0A:F7:F0:6C:56:43:3F:77:76 3
      ```

8. For each of the aliases from the `certutil` output in the preceding step that are required but missing in the `truststore` listing, execute the following commands to export and import them into the 7 domain's `truststore`.

```
>certutil -L -n verisignc1g1 -r -d $AS_NSS_DB> /tmp/verisignc1g1.der>keytool
-import -keystore cacerts.jks -storepass v3-master-password \
-file /tmp/verisignc1g1.der -alias verisignc1g1
```

> ℹ️ Sometimes just the alias names that are used in the NSS database are different, and the same certificate is, in fact, present in the 7 default `truststore`.

To Upgrade PKCS#11 Hardware Tokens

If you are using Eclipse GlassFish v2.x Enterprise Edition with Hardware Tokens (for example, FIPS-140 compliant Sun Cryptographic Accelerator 6000 or other Sun Cryptographic Accelerators) configured by means of NSS-PKCS11, then the v2.x EE-to-7 upgrade solution is to directly configure the Hardware Token as a PKCS11 token using the JDK-JSSE supported mechanisms for configuring

PKCS#11 tokens.

1. Set the `javax.net.ssl.keyStoreType jvm-options` in Eclipse GlassFish 7 to PKCS11.

   ```
   <jvm-options>-Djavax.net.ssl.keyStoreType=PKCS11</jvm-options>
   ```

2. Set the `javax.net.ssl.keyStore` URL should be set to l since this is a hardware token.

   ```
   <jvm-options>-Djavax.net.ssl.keyStore=NONE</jvm-options>
   ```

3. Change the password for the `truststore` and the Eclipse GlassFish `MasterPassword` to match the PIN of your `HardwareToken`.

4. Since you are using a Hardware Token, you can delete the `keystore.jks` for the migrated domain.

5. Ensure the `token-alias` for the hardware token (private key) that you intend to use as the Server's Key for SSL is mentioned in every relevant place in the `domain.xml` for the domain.
   For example, the `cert-nickname` attribute for the `<ssl/>` element under the `protocol` configuration.

6. If the Hardware Token is to act as a `TrustStore` as well, remove the `cacerts.jks` file from the domain-dir/`config` directory.
   Ensure that the following two `jvm-options` are set in the `domain.xml` file:

   ```
   <jvm-options>-Djavax.net.ssl.trustStore=NONE</jvm-options>
   <jvm-options>-Djavax.net.ssl.trustStoreType=PKCS11</jvm-options>
   ```

Upgrading Clusters and Node Agent Configurations

This section explains additional steps you need to perform when upgrading cluster and node agent configurations from Application Server or Enterprise Server to Eclipse GlassFish 7.

Eclipse GlassFish 7 does not support node agents. As part of the upgrade process, any node agent elements in the older source configuration are transformed into `CONFIG` node elements in the `domain.xml` file for the upgraded DAS. If the source node agent configuration is incompatible with your Eclipse GlassFish 7 installation, you must correct the node configuration on the upgraded DAS.

In addition, although the source cluster configuration is retained in the `domain.xml` file for the upgraded DAS, it is still necessary to install Eclipse GlassFish 7 on each node host and manually re-create the server instances that are contained in the clusters.

The following topics are addressed here:

- Overview of Cluster and Node Agent Upgrade Procedures

- To Correct the Configuration of a Node After an Upgrade

- To Re-Create a Cluster

Overview of Cluster and Node Agent Upgrade Procedures

The general steps for upgrading a cluster and node agent configuration so it will work in Eclipse GlassFish 7 are as follows:

1. Perform a side-by-side upgrade of the DAS. This procedure is described in Performing a Side-By-Side Upgrade With Upgrade Tool.

2. 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.

3. Correct the node configuration on the upgraded DAS, if necessary. This procedure is described in To Correct the Configuration of a Node After an Upgrade.

4. Re-create the clusters and server instances on each Eclipse GlassFish 7 node host. This procedure is described in To Re-Create a Cluster.

To Correct the Configuration of a Node After an Upgrade

As part of the upgrade process, node agent elements in the DAS configuration are transformed into Eclipse GlassFish node elements of type `CONFIG`. This transformation does not affect the node agent directories for Eclipse GlassFish instances. To create the equivalent directories for Eclipse GlassFish instances after an upgrade, you must re-create the instances as explained in To Re-Create a Cluster.

The name of an upgraded node is the name of the node agent from which the node is transformed.

The host that the node represents is obtained from the configuration of the original node agent or, if not specified, is not set. If the configuration of the original node agent did not specify the name of the node host, you must update the node to specify the host that the node represents.

Default values are applied to the remainder of the node's configuration data.

The default values of the following items in a node's configuration data might not meet your requirements for the upgraded installation of Eclipse GlassFish:

- The parent of the base installation directory of the Eclipse GlassFish software on the host, for example, `/export/glassfish7`.
  The default is the parent of the default base installation directory of the Eclipse GlassFish 7 software on the DAS host. If the Eclipse GlassFish software is installed under a different directory on the node host, you must update the node's configuration to specify the correct directory.

- The directory that will contain the Eclipse GlassFish instances that are to reside on the node.
  The default is as-install`/nodes`, where as-install is the base installation directory of the Eclipse GlassFish software on the host. If you require the instances to be contained in a different directory, you must update the node's configuration to specify that directory.

If you are using secure shell (SSH) for centralized administration, you must also change the type of the node to `SSH` to enable the node for remote communication.

For more information about Eclipse GlassFish nodes, see "Administering Eclipse GlassFish Nodes" in Eclipse GlassFish High Availability Administration Guide.

Before You Begin

Ensure that the following prerequisites are met:

- A side-by-side upgrade on the DAS has been performed. For more information, see Performing a Side-By-Side Upgrade With Upgrade Tool.

- If you are changing the type of the node to SSH, ensure that SSH is configured on the host where the DAS is running and on the host that the node represents. For more information, see " Setting Up SSH for Centralized Administration" in Eclipse GlassFish High Availability Administration Guide.

- If you are upgrading from an Enterprise Profile configuration that uses NSS authentication, ensure that the procedure in Upgrading Installations That Use NSS Cryptographic Tokens has been performed. Eclipse GlassFish 7 does not support NSS authentication.

  1. Ensure that the DAS is running.
     Remote subcommands require a running server.

  2. Update the node's configuration data to specify the correct directories and, if necessary, change the type of the node.

     > **ⓘ** Only the options that are required to complete this task are provided in this step. For information about all the options for changing the node's configuration data, see the update-node-ssh(1) help page or the update-node-config(1) help page.

```
asadmin> node-update-subcommand [--installdir as-install-parent] [--nodedir node-dir]
[--nodehost node-host] node-name
```

node-update-subcommand

The subcommand to run to update the node.

- If you are leaving the type of the node as CONFIG, run the update-node-config subcommand on the node.

- If you are changing the type of the node to SSH, run the update-node-ssh subcommand on the node.

as-install-parent

The full path to the parent of the base installation directory of the Eclipse GlassFish software on the host, for example, /export/glassfish7.

node-dir

The path to the directory that will contain Eclipse GlassFish instances that are to reside on the node. If a relative path is specified, the path is relative to the as-install directory.

node-host

The name of the host that the node is to represent after the node is updated.

node-name

The name of the node to update. This name is the name of the node agent from which the node was transformed.

Example 2-2 Correcting the Configuration of a Node After an Upgrade

This example updates the path to the directory that will contain instances that are to reside on the node `xk01` to `/export/home/gf/nodes`. Because this node is transformed from a node agent, the type of the node is `CONFIG`. Therefore, type of the node is not changed.

```
asadmin> update-node-config --nodedir /export/home/gf/nodes xk01
Command update-node-config executed successfully.
```

Next Steps

Re-create the cluster configuration from the older source installation in the new Eclipse GlassFish 7 installation in as explained in To Re-Create a Cluster.

See Also

- " Setting Up SSH for Centralized Administration" in Eclipse GlassFish High Availability Administration Guide

- "Administering Eclipse GlassFish Nodes" in Eclipse GlassFish High Availability Administration Guide

- update-node-config(1)

- update-node-ssh(1)

To Re-Create a Cluster

This procedure explains how to re-create a clustered Eclipse GlassFish or Enterprise Server configuration for Eclipse GlassFish 7.

Before proceeding with these instructions, ensure that you have completed the following procedures:

- Perform the standard upgrade to Eclipse GlassFish 7 on the DAS, as described in Performing a Side-By-Side Upgrade With Upgrade Tool.

- Perform a new (not upgrade) installation of Eclipse GlassFish 7 on each node host. See the Eclipse GlassFish Installation Guide for instructions.

- Correct the upgraded node configuration, if necessary, as described To Correct the Configuration of a Node After an Upgrade.

1. 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.

2. 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.
```

3. 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 To Correct the Configuration of a Node After an Upgrade.

4. 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.

5. 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 Potential Upgrade Problems

This section addresses issues that can occur during an upgrade to Eclipse GlassFish 7.

The following topics are addressed here:

- Cluster Profile Security Setting
- Cluster Profile Upgrade on Windows

Cluster Profile Security Setting

When upgrading a clustered domain configuration from Application Server 9.1 or Enterprise Server v2 to Eclipse GlassFish 7, you may encounter problems if the `admin-service` element in the DAS `domain.xml` file sets both of the following attributes:

- `security-enabled=true`
- `type=das-and-server`

The `security-enabled` attribute must be set to `false` in the `admin-service` element for the DAS when `type` is set to `das-and-server`.

You can use the `get` subcommand to determine the values for these two attributes. For example:

- To display the value for the `security-enabled` attribute:

  ```
  asadmin> get configs.config.server-config.admin-service.jmx-
  connector.system.security-enabled
  ```

- To display the value for the type attribute:

```
asadmin> get configs.config.server-config.admin-service.type
```

If necessary, use the set subcommand to set security-enabled=false. For example:

```
asadmin> set configs.config.server-config.admin-service.jmx-connector.system.security-
enabled=false
```

Cluster Profile Upgrade on Windows

On Windows, when you upgrade cluster profile domains, you could encounter the following error:

```
Fatal error while backing up the domain directory
```

To resolve this error, look for and remove any hidden files in the source domain's directory and re-run Upgrade Tool.