Eclipse GlassFish Troubleshooting Guide, Release 7

Eclipse GlassFish

Troubleshooting Guide

Release 7

Contributed 2018 - 2024

This guide describes common problems that you might encounter when using Eclipse GlassFish and how to solve them.

Eclipse GlassFish Troubleshooting Guide, Release 7

Copyright © 2013, 2019 Oracle and/or its affiliates. All rights reserved.

This program and the accompanying materials are made available under the terms of the Eclipse Public License v. 2.0, which is available at http://www.eclipse.org/legal/epl-2.0.

SPDX-License-Identifier: EPL-2.0

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

Preface

> This documentation is part of the Java Enterprise Edition contribution to the Eclipse Foundation and is not intended for use in relation to Java Enterprise Edition or Orace GlassFish. The documentation is in the process of being revised to reflect the new Jakarta EE branding. Additional changes will be made as requirements and procedures evolve for Jakarta EE. Where applicable, references to Jakarta EE or Java Enterprise Edition should be considered references to Jakarta EE.
>
> Please see the Title page for additional license information.

This guide describes common problems that you might encounter when using Eclipse GlassFish and how to solve them.

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 Overview of Eclipse GlassFish Troubleshooting

This chapter describes some of the tools, methods, and resources available for troubleshooting Eclipse GlassFish. Guidelines for evaluating and investigating a problem are provided.

This chapter contains the following sections:

- Identifying the Problem
- Troubleshooting Tools
- Where to Go for More Information

Identifying the Problem

Application servers are typically deployed in complex operating environments and involve many technologies, products, and tools. Understanding and diagnosing problems in enterprise software with many components performing many tasks can be a challenge. This section describes how to get started , and contains the following topics:

- First Steps
- Gathering Information
- Examining Log Files
- Monitoring the System

First Steps

Sometimes the most obvious solutions are overlooked. As you begin your investigation, try the following steps first.

Verify System Requirements and Configuration

Ensure that your system meets the requirements listed in "Hardware and Software Requirements" in Eclipse GlassFish Release Notes. Problems are likely to arise if you attempt to install on a platform that is not supported or on a system that in some other way does not meet release requirements. Also see "Known Issues" in Eclipse GlassFish Release Notes for known issues related to installation.

Eclipse GlassFish requires JDK release 11 or newer. If necessary, download and install the required JDK software.

On Solaris, Linux, and Windows systems, JDK software is available from the Java SE downloads page (`https://www.oracle.com/java/technologies/downloads/`).

For Mac OS X systems, Eclipse GlassFish uses the JDK that is part of the Macintosh operating system. If necessary, obtain the required JDK version from the Mac OS X Updates site (`http://support.apple.com/downloads/`).

Also ensure that the `JAVA_HOME` environment variable on your system points to the JDK installation directory and not the Java Runtime Environment (JRE) software.

Set `JAVA_HOME` and `$JAVA_HOME/bin` in the `PATH` to point to the supported JDK version.

Review Known Issues

Your problem might be related to a known issue for the release. Refer to the Eclipse GlassFish Release Notes for the latest information regarding known issues and possible workarounds. Also search the GlassFish issues at `https://github.com/eclipse-ee4j/glassfish/issues`.

Search the Product Documentation

Eclipse GlassFish includes complete product documentation. Search the documentation to see if your problem is addressed. See Eclipse GlassFish Documentation Set for the books in the documentation set and a description of their contents. Also see the Administration Console online help for detailed information about performing administrative tasks using the graphical interface.

The following books will be especially helpful for troubleshooting:

- Eclipse GlassFish Release Notes, which provides the latest information regarding known issues and possible workarounds.
- Eclipse GlassFish Error Message Reference, which lists error messages you might encounter when using Eclipse GlassFish.

Use the product documentation to learn more about Eclipse GlassFish. The more you know about the product the easier it might be to figure out why something isn't working.

Search the GlassFish Mailing Lists and Forums

Lists and forums are extremely helpful resources, and are accessed as follows:

- GlassFish mailing lists: `https://accounts.eclipse.org/mailing-list/glassfish-dev`
- GlassFish user forum: `https://www.eclipse.org/forums/index.php/f/419/`

Gathering Information

Troubleshooting and analysis require information. The more information you have the easier it is to classify a problem and search for its solution. Detailed information will also be necessary should you need to contact others for support, either through a community mailing list or through more formal Sun support channels.

As you continue your investigation, consider the following questions.

When Does the Problem Occur?

- What do the logs show? What kind of error message are you getting? For more information about logs, see Examining Log Files. Logs are a good place to start your troubleshooting.
- What are you trying to do when the problem occurs and how are you doing it? What is the

sequence of events?

- Does the problem occur every time you perform the same type of operation, or does it seem random? Can you reproduce the issue?
- Have other events or problems occurred that could possibly be related, such as web sessions closing early, exceptions being thrown in your own application logic, components not being correctly deployed or undeployed?

What Is Your Environment?

- What version of Eclipse GlassFish are you using? What operating system and version? What JDK version? Many problems are caused simply because system requirements for the release are not met. Refer to the Eclipse GlassFish Release Notes for the latest information regarding system requirements, and make sure your system meets those requirements.
- What resources does your system have, such as memory, disk, and swap space? If system memory is an issue, look for ways to optimize your configuration and consider taking other actions such as adding memory or decreasing system load.
- Have any patches been applied? If so, what are the product and operating system patch numbers?
- What other products and technologies are installed and being used in the installation?
- How many application servers, web servers, and directory servers are installed? What are the products and versions?
- How is the web server connected to Eclipse GlassFish? On the same system?
- How is Eclipse GlassFish connected to the directory server?
- What JDBC driver is being used to access the database?

What Is Your System Configuration?

- What are your settings?
- On which port is Eclipse GlassFish configured — the default or something else?
- What defaults were changed during installation and what are the values?
- What other settings were changed from the defaults and what are their values?
- What are the parameters related to JVM heap, stack, and garbage collection set to?
- What are the JVM options?
- What is the permgen size? `OutOfMemoryError:PermGen space` errors are common and indicate that you need to increase the permanent generation space available.
- Is SSL enabled?
- What are your network settings (proxy, firewall, and so on)? What happens if you disable your firewall and attempt the task?

What Is Different?

- What is different that could have provoked the problem or triggered the event? Was something

new added or changed? Have any new applications been deployed? If changes have been made recently, consider backing them out and seeing what happens — does the problem still occur?

- Was the feature or functionality working correctly at one time? If so, what changed or happened between then and now?
- Is this working on another system? If so, what is different about that environment?

Examining Log Files

Logging is one of your most important troubleshooting tools. It is the process by which Eclipse GlassFish captures data about events that occur during server operation, such as configuration errors, security failures, or server malfunction. This data is recorded in log files, and is usually your first source of information when Enterprise Server problems occur. The primary purpose of log files is to provide troubleshooting information. Analyzing the log files can help determine the health of the server and identify problem areas.

By default, log information for each Eclipse GlassFish server instance is captured in a `server.log` file. That is, each instance, including the domain administration server (DAS), has an individual log file. By default, the log file for the DAS is located in domain-dir`/logs`, and the log file for each instance is located in instance-dir``/logs`.

In addition, for domains that use clustering, Eclipse GlassFish captures log information for each cluster instance in a `cluster.log` file. By default, the `cluster.log` file is also located in instance-dir`/logs`.

Oracle recommends using the Administration Console to view logging information. However, you can open a log file in a text editor and search for the module or message in which you are interested. Eclipse GlassFish also lets you collect log files into a ZIP file, which provides a convenient means to collect and view the log files for an instance or a domain even when it is not running.

You configure the Logging Service by setting attributes in the `logging.properties` file. Each server, configuration, instance, and cluster in the Eclipse GlassFish domain has an individual `logging.properties` file. The root directory in which these `logging.properties` files are located is the same directory as for the `domain.xml` file, typically domain-dir/`config`. The default target when configuring logging attributes is the DAS. However, you can optionally target a specific server, instance, or cluster. You can also target a configuration that is shared by one or more instances or clusters. The Logging Service can also be configured using the Administration Console.

Log levels such as `SEVERE`, `WARNING`, `INFO`, `CONFIG`, and others can be set to provide different types and amounts of information. The default setting is `INFO`. Each Eclipse GlassFish module has its own logger, and each logger has its own namespace. Log levels can be set globally for all loggers, or individually for module-specific loggers.

For information about using the Administration Console log viewer and logging functions, see the Administration Console online help. For information about using the command line for logging functions, see "Administering the Logging Service" in Eclipse GlassFish Administration Guide.

Monitoring the System

Monitoring is another helpful tool. It is the process of reviewing the statistics of a system to improve performance or solve problems. By monitoring the state of various components and services deployed in Eclipse GlassFish you can identify performance bottlenecks, predict failures, perform root cause analysis, and ensure that everything is functioning as expected. For more information about monitoring, including JConsole information, see "Administering the Monitoring Service" in Eclipse GlassFish Administration Guide.

Troubleshooting Tools

Several tools are available that can be used to collect information for troubleshooting purposes. This section provides basic information about some of them, and includes the following:

- Operating System Utilities
- Stack Traces and Thread Dumps
- VisualVM
- JVM Command-Line Tools

Operating System Utilities

Operating system utilities, such as `pkginfo` and `showrev` on Solaris and `rpm` on Linux, are helpful in gathering system information.

The `ps -ef` command provides helpful information about processes that are running, including their process identification numbers (PIDs).

Stack Traces and Thread Dumps

A stack trace is a user-friendly snapshot of the threads and monitors in a Virtual Machine for the Java platform (Java Virtual Machine or JVM machine). A thread dump shows what every thread in a JVM is doing at a given time and is useful in debugging. When the application server freezes, hangs, or becomes sluggish for no apparent reason, you should generate and analyze a thread dump.

This section explains how to obtain a thread dump for Eclipse GlassFish. More information about analyzing the information contained in a thread dump can be found in "An Introduction to Java Stack Traces" (`http://java.sun.com/developer/technicalArticles/Programming/Stacktrace`).

To Obtain a Server Thread Dump

Type the following command:

```
asadmin generate-jvm-report --type=thread
```

See Also

`generate-jvm-report(1)`

VisualVM

VisualVM is a Java troubleshooting tool that uses various technologies such as `jvmstat`, JMX, and Attach API to access monitored applications. VisualVM is a tool for visualizing data sources and by default visualizes the following types: applications, hosts, snapshots, core dumps, heap dumps, and thread dumps. These data sources are visualized in VisualVM so that they can be monitored for the purposes of analysis, management, and troubleshooting. VisualVM is commonly used to detect memory leaks.

VisualVM has a GlassFish plugin that enhances monitoring of hosted applications by adding specialized overview, a tab for monitoring the HTTP Service, and the ability to visually select and monitor any of the deployed web applications. You can experiment with VisualVM troubleshooting capabilities, but note that various features depend on the Java versions used in the client and server. Depending on your configuration, you might only get parts of the VisualVM features. For more information about VisualVM, see `http://visualvm.java.net`.

JVM Command-Line Tools

JVM command-line tools can be used to provide valuable information about hung Java processes or Java core dumps. These tools include the following:

- `jstack`: Prints Java stack traces of Java threads for a given Java process or core file or a remote debug server.
- `jinfo`: Prints Java configuration information for a given Java process or core file or a remote debug server.
- `jmap`: Prints shared object memory maps or heap memory details for a given process or core file or a remote debug server.
- `jsadebugd`: Attaches to a Java process or core file and acts as a debug server. Remote clients such as `jstack`, `jmap`, and `jinfo` can attach to the server using Java Remote Invocation Method (RMI).
- `jhat`: Enables Java heap dumps to be browsed using a web browser.
- `jstat`: Displays performance statistics for an instrumented HotSpot JVM.
- `jps`: Lists the instrumented HotSpot JVMs on the target system

Where to Go for More Information

These resources were mentioned throughout this chapter and are provided again here for easy reference.

- Release Notes: Eclipse GlassFish Release Notes
- Error messages: Eclipse GlassFish Error Message Reference
- Mailing lists and forums: See Search the GlassFish Mailing Lists and Forums for details about accessing lists and forums.
- GlassFish community site: `http://glassfish.java.net/`

2 Specific Issues

This chapter lists problems that you might encounter when using Eclipse GlassFish 7. The following topics are addressed:

- Cannot Access Local Server (`http://localhost:8080`)
- Cannot Access Remote Server
- Cannot Access the Administration Console
- Cannot Access a Server Application
- Administrator User Name or Password Not Known
- Experience Issues Related to the JDK
- Server Will Not Start on Windows (Port Conflict)
- Eclipse GlassFish Fails to Start in an OpenSolaris Zone
- Two Server Instances Bind to Same Port on Windows
- Cannot Produce a JVM Thread Dump After Server Crash
- Issues Related to Applications
- Issues Related to `asadmin`
- Issues Related to Installation
- Issues Related to Security

Cannot Access Local Server (`http://localhost:8080`)

When this error occurs, check the following:

- Did the Server Start?
- Was the Server Started at the Expected Port?
- Is a Personal Firewall Enabled?

Did the Server Start?

Description

If the console window is still open, the expected message indicates that the default domain was started successfully.

If the console window is already closed, check for messages in the log file. This is the default location:

```
domain-dir/logs/server.log
```

If startup was successful, the expected message is similar to that on the console, and appears at the

end of the log file.

For more information about starting a domain, see " Starting and Stopping a Domain" in Eclipse GlassFish Administration Guide. To easily determine if a domain is running, use the `asadmin list-domains` command.

Was the Server Started at the Expected Port?

Description

The server might be running at a different port number than expected, either because it was intentionally installed there, or because another server was already running on the default port when the server was installed.

To Determine Which Port Number the Server Is Using

Follow this procedure.

1. Examine the server's configuration file:

   ```
   domain-dir/config/domain.xml
   ```

2. Find the `network-listener` element.
3. Inspect the value of the `port` attribute.

   Be sure to enter the correct port number when invoking the server.

   > The server's default port number is `8080`, however, there are a number of ways in which the expected value can change:
   >
   > - A different port number was specified during installation.
   > - A previous installation exists.

Is a Personal Firewall Enabled?

Issues might result when personal firewalls are enabled. Disable your personal firewall and see if the server access problem still exists.

Cannot Access Remote Server

When attempting to open the start page of Eclipse GlassFish, the initial screen does not appear.

When this error occurs, check the following:

- Is the Server Available Locally?
- Is the Proxy Setting Causing a Problem?

Is the Server Available Locally?

Description

If the server cannot be accessed from the web, but it is running locally, then the server is actually running.

Solution

Verify that the server is running locally.

To Verify That the Server Is Running Locally

Follow this procedure.

1. Log on to the host where the server is running.
2. Go to the local web page. For example, if `8080` is the default port, go to:

   ```
   http://localhost:8080/
   ```

   If the start page does appear, the web connection is encountering a problem that prevents accessing the server remotely. If the start page does not appear, see Did the Server Start?.

Is the Proxy Setting Causing a Problem?

Description

The server should be accessible directly from the host on which it is running (`localhost`); for example, using the default port `8080`:

```
http://localhost:8080/
```

Solution

A server instance running on `localhost` might not be accessible if the server host machine is connected to the web through a proxy. To solve this problem, do one of the following:

- Set the browser to bypass the proxy server when accessing `localhost`. Refer to the browser's help system for information about how to do this.
- Use the fully-qualified host name or IP address of your system; for example:

  ```
  http://myhost.mydomain.com:8080/
  ```

- Create an entry in the system's hosts file (for example, pointing `127.0.0.1` to `localhost`; `127.0.0.1` is not proxied).

To determine the host name for the `localhost` machine, type `hostname` at the command prompt.

Cannot Access the Administration Console

The Administration Console provides a graphical interface for administrative functions. If the Administration Console is not accessible, check the following:

- Is the Application Server Running?
- Is the Administration Console Running on the Expected Port?

For more information about the Administration Console, see "Administration Console" in Eclipse GlassFish Administration Guide.

Is the Application Server Running?

Description

The server must be running before the Administration Console can be accessed.

Solution

Review the information in Did the Server Start? to determine if the server is running.

Is the Administration Console Running on the Expected Port?

Description

The default port number for the Administration Console is `4848`. However, it could be running on a different port number than expected, either because it was intentionally installed there, or because that port was in use when the server was started.

Solution

Refer to Was the Server Started at the Expected Port? for guidelines on verifying the port on which the Administration Console is running. Be sure to enter the correct port number and HTTP protocol when invoking the Administration Console.

Cannot Access a Server Application

If a particular application cannot be accessed through Eclipse GlassFish, check the following:

- Is the Application Server Running?
- Was Application Deployment Successful?

Is the Application Server Running?

Description

If Eclipse GlassFish is not running, applications are not accessible.

Solution

Review the information in Did the Server Start? to determine if the server is running. The server must be running before a server application can be accessed.

Was Application Deployment Successful?

Description

An application must be successfully deployed before it can be accessed.

Solution

Verify that the application was successfully deployed. There are several ways to do this:

- Check the server's log file for related entries:

  ```
  domain-dir/server.log
  ```

- Use the `asadmin list-applications` command to determine which applications are deployed.
- View the Applications page in the Administration Console, accessed by clicking the Applications node.

For more information about deploying applications, see "Deploying Applications" in Eclipse GlassFish Application Deployment Guide. Also see the Administration Console online help.

Administrator User Name or Password Not Known

If you have forgotten the administrator user name, you can find it by inspecting the domain-dir`/config/admin-keyfile` file, where domain-dir is the directory for the domain. In the default domain, `domain1`, the file to inspect is domain-dir`/config/admin-keyfile`. For a different domain, substitute its name in the path.

If you have forgotten the administrator password, one solution is to create a new domain with the admin username and password that you want, then copy the entry from the `config/admin-keyfile` file in that new domain to the other domain.

Experience Issues Related to the JDK

Description

You experience JDK-related issues in a variety of circumstances.

Solution

Eclipse GlassFish 7 requires JDK 11, so check your system for that dependency.

Ensure that the required JDK software is installed and that the `JAVA_HOME` environment variable points to the JDK installation directory, not the Java Runtime Environment (JRE) software.

Set `JAVA_HOME` and `$JAVA_HOME/bin` in the `PATH` to point to the supported JDK version.

Server Will Not Start on Windows (Port Conflict)

If a message similar to the following is displayed when starting Eclipse GlassFish on Microsoft Windows, a server port conflict has occurred:

```
Address already in use
```

This error occurs when another application is running on the Eclipse GlassFish port (default `8080`), or because a previous instance of Eclipse GlassFish did not shut down cleanly.

You might also check the following:

- Is Another Application Running on the Server's Port?
- Has an Ungraceful Shutdown Occurred on a Previously Running Server?

Is Another Application Running on the Server's Port?

If another application is using the server's port, stop the other application, then restart Eclipse GlassFish.

Has an Ungraceful Shutdown Occurred on a Previously Running Server?

Use the `asadmin stop-domain` command to stop the server, or explicitly kill the Java process and then restart Eclipse GlassFish.

Eclipse GlassFish Fails to Start in an OpenSolaris Zone

Description

If Eclipse GlassFish is installed in an OpenSolaris zone, an attempt to start a domain might fail with the following error message:

```
Waiting for DAS to start ..Error starting domain: domain.
The server exited prematurely with exit code 6.
Before it died, it produced the following output:

UTF ERROR ["../../../src/solaris/instrument/EncodingSupport_md.c":66]:
```

```
Failed to complete iconv_open() setup
```

The failure occurs because, by default, an OpenSolaris zone is installed without language and encoding support, which Eclipse GlassFish requires.

Resolution: To Enable Eclipse GlassFish to Run in an OpenSolaris Zone

1. Install the package that provides language and encoding support for your locale.

   ```
   $ pkg install package-name
   ```

   package-name

   The name of the package that provides language and encoding support for your locale. For example, the name of the package that provides language and encoding support for the US English locale is `SUNWlang-enUS`.

2. Set the `LANG` environment variable to the code for your locale.

   For example, if your locale is US English, set the `LANG` environment variable to `en_US.UTF-8`.

Two Server Instances Bind to Same Port on Windows

Description

This problem occurs on Windows XP systems with Eclipse GlassFish software, and is due to a known Windows security flaw rather than a problem with Eclipse GlassFish itself.

The problem occurs when two or more instances of Eclipse GlassFish are created using the same port number for the `instanceport` option; for example:

```
asadmin create-domain -adminport 5001 options -instanceport 6001 domain
asadmin create-domain -adminport 5002 options -instanceport 6001 domain
```

When the two domains are started on a UNIX or Linux system, a port conflict error is thrown and the second instance fails to start. However, when the two domains are started on Windows XP, no error is thrown, both server instances start, but only the first instance is accessible at the specified port. When that first server instance is subsequently shut down, the second instance then becomes accessible. Moreover, when both instances are running, the Windows `netstat` command only reports the first instance.

Solution

Be sure to use unique port numbers for all server instances on Windows systems.

Cannot Produce a JVM Thread Dump After Server Crash

Description

If Eclipse GlassFish crashes, the server dumps a core file and, by default, restarts with the `-Xrs` flag, which prevents the dump of a JVM thread dump.

Solution

To Obtain a Server Thread Dump

Type the following command:

```
asadmin generate-jvm-report --type=thread
```

See Also

`generate-jvm-report`(1)

Issues Related to Applications

Cannot Undeploy or Redeploy Application With Open Streams to `jar` Files (Windows)

Description

On Windows systems, after running an application, subsequent attempts to undeploy it or redeploy it throw exceptions about the server being unable to delete a file or rename a directory.

On Windows systems, an application may use `getClass().getResource` or `getResourceAsStream` methods to locate a resource inside the application, particularly in `jar` files that are in the application or accessible to it. If the streams remain open, subsequent attempts to redeploy or undeploy the application can fail. In addition, the Java runtime by default caches streams to `jar` files for performance reasons.

Solution

Be sure to close streams opened by your applications. Also, if an application needs to be redeployed or undeployed repeatedly, and also needs to obtain a resource from a `jar` file using `getResource` or `getResourceAsStream`, consider using `getClass().getResource`, which returns a URL object, then invoke the `url.setUseCaches` method to turn off caching for that `jar` file, and use `url.getInputStream()` to obtain the stream.

Although turning off caching for access to the `jar` file can slow performance, this approach does allow the application to be undeployed or redeployed. Note also that if the `getClass().getResourceAsStream` method is used instead, then the `jar` file in which the resource is

located is cached (this is the default Java runtime setting) and remains open until the server is stopped.

`MaxPermGen` Exception

Description

Application servers such as Eclipse GlassFish allow you to redeploy an application without restarting the server. Simply make the change in your source code, compile the source, and redeploy the application.

Each application is loaded using its own classloader. When you undeploy an application, its classloader is discarded with all the classes it loaded and is garbage collected sooner or later. However, if there's a reference from outside an application to an object in the application loaded by the application's classloader, that object can't be garbage collected. The reference holds the object in memory.

The memory in the Virtual Machine is divided into a number of regions. One of these regions is `PermGen`. It's an area of memory used to (among other things) load class files. The size of this memory region is fixed; it does not change when the VM is running. You can specify the size of this region with a command line switch: `-XX:MaxPermSize`. Setting the `-Xmx` parameter does not help: this parameter only specifies the total heap size and does not affect the size of the `PermGen` region.

If you keep loading new classes that can't be garbage collected because of references to them from outside the application, the VM runs out of space in the `PermGen` region, even if there's plenty of memory available. This is called a classloader leak. The resulting exception is `java.lang.OutOfMemoryError: PermGen space`.

The `java.lang.String.intern()` method also allocates memory in the `PermGen` region. If your application uses this method with strings and holds references to these strings, thereby making garbage collection impossible, your application may cause the same `PermGen space` exception.

Solution

Classloader leaks are difficult to diagnose. Most profilers list leaked objects but don't highlight the ones causing classloader leaks. Most profilers also stop tracing as soon as they reach a class object or classloader.

One diagnostic approach involves undeploying the application and triggering a memory dump using the JDK 11 `jcmd` tool. The simplest analysis is to list all instances of `java.lang.Class` and look for class objects that have many instances. This is a sign that the class has been loaded multiple times without being garbage collected.

See https://www.eclipse.org/openj9/docs/tool_jcmd/ for more information.

Issues Related to `asadmin`

`asadmin start-domain` Command Fails

The command `asadmin start-domain` fails with the following error:

```
There is more than one domain...
```

Description

When issued with no arguments, the command `asadmin start-domain` fails.

This error occurs when there is more than one domain in the domains directory, none of them is named `domain1`, and no domain is specified with the `start-domain` command.

Solution

Specify the domain when issuing the `start-domain` command:

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

For example:

```
asadmin start-domain mycustomdomain
```

Cannot Stop Domain Using `asadmin stop-domain`

Description

You cannot stop the domain using the `asadmin stop-domain` command.

Solution

Look for error messages that display in the console when you issue the command.

Search the `server.log` file for error messages related to your inability to stop the domain.

Issues Related to Installation

Installation Hangs During Update Tool Configuration

Description

Installation hangs more than five minutes during Update Tool configuration.

Solution

Cancel the installation and run the installation program again, but this time deselect the Install Update Tool check box. Update Tool can be installed later from as-install`/bin/`. For more

information about Update Tool, see "Update Tool" in Eclipse GlassFish Administration Guide. For general information about Eclipse GlassFishinstallation, see the Eclipse GlassFish Installation Guide.

> Update Tool differs from Upgrade Tool, which is used to migrate the configuration and deployed applications from an earlier version of Eclipse GlassFish to the current version. For more information about Upgrade Tool and upgrading, see the Eclipse GlassFish Upgrade Guide.

Eclipse GlassFish Components Not Removed During Uninstallation

Description

Not all Eclipse GlassFish directories are automatically removed by the uninstallation program. Some directories and files remain after uninstalling.

Solution

Examine the remaining directories and remove any files or directories that you do not want, including hidden directories prefixed with a dot. It is safe to remove uninstallation and installation log files after you have examined them.

For information related to uninstallation, see "Uninstalling GlassFish" in Eclipse GlassFish Installation Guide.

Issues Related to Security

`java.security.AccessControlException`: Access Denied Error

Description

The following error occurs from an application client, or appears in the `server.log` file:

```
java.security.AccessControlException: access denied
(java.util.PropertyPermission name write...)
```

There is a permissions issue in the policy files. Either the `client.policy` file for the application client or the `server.policy` file for server side components does not have permission to set the property.

Solution

Add the permission in `client.policy` (for the application client), or in `server.policy` (for web modules) for the application that needs to set the property. By default, applications only have read permission for properties.

For example, to grant read/write permission for all files in the codebase directory, add or append

the following to `client.policy` or `server.policy`:

```
grant codeBase "file:/.../build/sparc_SunOS/sec/-" {
   permission java.util.PropertyPermission "*", "read,write";
 };
```

Mutual Authentication Not Working With the Application Client

Description

This failure can occur when the keystore and truststore properties are not set properly.

Solution

Set the following properties on the JVM:

```
javax.net.ssl.keyStore=
<keystore-file-path>;javax.net.ssl.trustStore=<truststore-file-path>
```

To use the application client, set the environment variable `VMARGS` to the following value:

```
-Djavax.net.ssl.keyStore=${admin.domain.dir}/${admin.domain}/config/keystore.jks
-Djavax.net.ssl.trustStore=${admin.domain.dir}/${admin.domain}/config/cacerts.jks
```

3 Frequently Asked Questions

This chapter lists a few of the Frequently Asked Questions (FAQs) related to Eclipse GlassFish 7. For the latest FAQs, see the GlassFish wiki at `http://wikis.sun.com/display/GlassFish/GlassFishFAQIndex`.

This chapter contains the following:

- Administration FAQs
- Application FAQs
- Eclipse FAQs
- Extensibility FAQs
- Java Persistence FAQs
- Update Tool FAQs

Administration FAQs

Which `asadmin` Commands are Supported?

Use `asadmin list-commands` to determine which commands are supported and available for use. For related information, see "Subcommands for the asadmin Utility" in Eclipse GlassFish Administration Guide.

Application FAQs

How Do I Debug Applications?

See "Debugging Applications" in Eclipse GlassFish Application Development Guide.

How Do I Change Application Permissions?

See "Changing Permissions for an Application" in Eclipse GlassFish Application Development Guide.

What Are the Restrictions and Optimizations for Session Beans?

See "Session Bean Restrictions and Optimizations" in Eclipse GlassFish Application Development Guide.

Eclipse FAQs

Where Can I Find More Information About Eclipse and Eclipse GlassFish Integration?

Eclipse is a development environment that provides a framework, tools, and runtime for deploying

and managing software. The GlassFish plugin for Eclipse provides important changes to better support Eclipse GlassFish and to provide optimal redeployment of web applications. For more information, see the GlassFish Plugins page: `http://glassfishplugins.java.net`.

Extensibility FAQs

How Do I Develop Add-On Components?

See the Eclipse GlassFish Add-On Component Development Guide for complete details.

How Do I Add Containers?

Eclipse GlassFish has a highly modular architecture that enables you to add new types of containers that handle many application types, not just Jakarta EE. Eclipse GlassFish defines a service provider interface (SPI), which container developers implement. The Eclipse GlassFish framework then invokes the container's implementation of the SPI to detect, deploy, and manage applications of the new type.

For more information, see "Adding Container Capabilities" in Eclipse GlassFish Add-On Component Development Guide.

How Do I Extend the Administration Console?

See "Extending the Administration Console" in Eclipse GlassFish Add-On Component Development Guide.

Java Persistence FAQs

What Are the Restrictions and Optimizations for the Java Persistence API?

See "Restrictions and Optimizations" in Eclipse GlassFish Application Development Guide.

Update Tool FAQs

How Do I Use Update Tool to Extend My Eclipse GlassFish Installation?

Enterprise Server provides an administrative tool called Update Tool that enables you to install updates and add-on components to your existing Enterprise Server installation.

Update Tool can be accessed as a standalone graphical tool from the command line (using the `updatetool` command from as-install-parent/`bin`), or as a browser-based graphical tool from the Administration Console (using the Update Tool node). For more information about Update Tool, see "Update Tool" in Eclipse GlassFish Administration Guide.

To update or remove installed components, you must use the standalone graphical Update Tool, not the Administration Console Update Tool.

A command-line interface is also available for Update Tool. This interface uses the `pkg` command and enables you to perform most of the tasks provided by the standalone graphical version. For more information about the `pkg` command, see "Extending and Updating Eclipse GlassFish" in Eclipse GlassFish Administration Guide.

> Update Tool differs from Upgrade Tool, which is used to migrate the configuration and deployed applications from an earlier version of Eclipse GlassFish to the current version. For more information about Upgrade Tool and upgrading, see the Eclipse GlassFish Upgrade Guide.

How Do I Turn Off the Notifier?

Update Tool provides automatic notifications of available updates after installation. These notifications can be turned off if desired.

To Turn Off the Notifier

1. Launch the standalone graphical tool using the `updatetool` command:

   ```
   as-install-parent/bin/updatetool
   ```

2. Click Preferences.
3. Click the Updates tab.
4. Deselect Automatically Check for Updates and click OK.