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 GlassFish Server Open Source Edition 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 GlassFish Server Open Source Edition 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, GlassFish Server Open Source Edition 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. GlassFish Server Open Source Edition also lets you collect log files into a ZIP file, which is convenient for the management and archival of multiple log files in your environment.

You configure the Logging Service by setting attributes in the `logging.properties` file. Each server, configuration, instance, and cluster in the GlassFish Server Open Source Edition 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 GlassFish Server Open

Source Edition 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 Chapter 7, "Administering the Logging Service," in *Oracle GlassFish Server 3.1 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 GlassFish Server Open Source Edition 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 Chapter 8, "Administering the Monitoring Service," in *Oracle GlassFish Server 3.0.1 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" on page 16
- "Stack Traces and Thread Dumps" on page 16
- "VisualVM" on page 17
- "JVM Command-Line Tools" on page 17

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.