

◆ ◆ ◆ CHAPTER 7

Administering the Logging Service

This chapter provides instructions on how to configure logging and how to view log information in the GlassFish Server Open Source Edition 3.1 environment.

The following topics are addressed here:

- “About Logging” on page 19
- “Configuring the Logging Service” on page 24
- “Viewing Log Records” on page 35

Instructions for accomplishing these tasks and editing logging service properties in the Administration Console are available from the Administration Console online help.

About Logging

Logging is the process by which GlassFish Server Open Source Edition captures information about events that occur, such as configuration errors, security failures, or server malfunction. This data is recorded in log files and is usually the first source of information when problems occur. Analyzing the log files can help you to determine the health of the server.

Although application components can use the Apache Commons Logging Library to record messages, the platform standard JSR 047 API is recommended for better log configuration.

The following topics are addressed here:

- “Log Files” on page 20
- “Logger Namespaces” on page 22
- “Logging Targets” on page 23
- “Logging Properties” on page 23

Log Files

GlassFish Server Open Source Edition log records are captured in one of two general types of log files:

- Server log files, which capture information about the operation of a server instance running in the domain. Each instance, managed server instance (that is, each cluster member), and the domain administration server (DAS) has an individual server log file.
- Cluster log files, which capture information about the operation of a cluster instance, if one or more are configured in the domain. Each managed server instance that is a member of a cluster has a cluster log file in addition to a server log file. However, the contents of the cluster log file may differ from one instance to another depending on factors such as how instances are apportioned in the cluster, applications running on them, how load balancing is configured, and failover state.

The following topics are addressed here:

- [“Log File Names and Locations”](#) on page 20
- [“Log Records”](#) on page 21
- [“Log Rotation”](#) on page 22

Log File Names and Locations

In an GlassFish Server Open Source Edition domain, log files have the following names and locations by default:

Instance	Default Log File Name and Location
DAS	<i>domain-dir</i> /logs/server.log
Each server instance	<i>instance-dir</i> /logs/server.log
Cluster instance	<i>instance-dir</i> /logs/cluster.log

For example, in a domain hosted on a given machine that includes a cluster with two managed servers (ClusterServer1 and ClusterServer1) and a standalone instance (StandaloneServer), the log files might be arranged in the following directory structure. In this directory structure, the server.log file for the DAS is located in *domain-dir*/logs.

```

as-install-parent directory
  glassfish/
    domains/
      domain-name/
        logs/
          server.log
    nodes/
      hostname/
        ClusterServer1/

```

```

logs/
  server.log
  cluster.log
ClusterServer2/
  logs/
    server.log
    cluster.log
StandaloneServer/
  logs/
    server.log

```

You can change the default name or location of a log file by modifying the logging properties file for the corresponding instance, described in [“To Change the Name and Location of the Log File” on page 25](#).

Log Records

GlassFish Server Open Source Edition log records follow a uniform format:

```
[#|yyyy-mm-ddThh:mm:ss.SSS-Z|Log Level|ProductName-Version|LoggerName|Key Value Pairs|Message|#]
```

- [# and #] mark the beginning and end of the record.
- The vertical bar (|) separates the fields of the record.
- *yyyy-mm-ddThh:mm:ss.SSS-Z* represents the date and time that the record was created. For example: `2006-10-21T13:25:53.852-0400`
- *Log Level* represents the log level. You can set any of the following values: SEVERE, WARNING, INFO, CONFIG, FINE, FINER, and FINEST. The default is INFO.
- *ProductName-Version* represents the current version of the GlassFish Server Open Source Edition. For example: `glassfish`
- *LoggerName* represents a hierarchical logger namespace that identifies the source of the log module. For example: `javax.enterprise.system.core`
- *Key Value Pairs* represents pairs of key names and values, typically a thread ID. For example: `_ThreadID=14;`
- *Message* represents the text of the log message. For all GlassFish Server Open Source Edition SEVERE and WARNING messages and for many INFO messages, the message begins with a message ID that consists of a module code and a numerical value. For example: `CORE5004`

The following is an example of a log record:

```
[#|2006-10-21T13:25:53.852-0400|INFO|GlassFish10.0|javax.enterprise.
system.core|_ThreadID=13;|CORE5004: Resource Deployed:
[cr:jms/DurableConnectionFactory].|#]
```

The Administration Console presents log records in a more readable display.

Log Rotation

By default, when a log file grows to 2 MB, GlassFish Server Open Source Edition renames (rotates) the file to incorporate a timestamp and creates a new log file. The log file is renamed as *log-type.log_date*, where *log-type* represents either `server` or `cluster`, and *date* represents the time of rotation.

You can configure the logging service to change the default settings for log file rotation, as explained in [“Setting Log File Rotation” on page 30](#).

Logger Namespaces

GlassFish Server Open Source Edition provides a logger for each of its modules. The following list is an example of the logger namespaces in a server instance as they appear when using the `list-log-levels` subcommand.

```

java.util.logging.ConsoleHandler      <FINEST>
javax.enterprise.resource.corba       <INFO>
javax.enterprise.resource.javamail    <INFO>
javax.enterprise.resource.jdo         <INFO>
javax.enterprise.resource.jms         <INFO>
javax.enterprise.resource.jta         <INFO>
javax.enterprise.resource.resourceadapter <INFO>
javax.enterprise.resource.sqltrace    <INFO>
javax.enterprise.resource.webcontainer.jsf.application <INFO>
javax.enterprise.resource.webcontainer.jsf.config      <INFO>
javax.enterprise.resource.webcontainer.jsf.context    <INFO>
javax.enterprise.resource.webcontainer.jsf.facelets   <INFO>
javax.enterprise.resource.webcontainer.jsf.lifecycle  <INFO>
javax.enterprise.resource.webcontainer.jsf.managedbean <INFO>
javax.enterprise.resource.webcontainer.jsf.renderkit  <INFO>
javax.enterprise.resource.webcontainer.jsf.resource   <INFO>
javax.enterprise.resource.webcontainer.jsf.taglib     <INFO>
javax.enterprise.resource.webcontainer.jsf.timing     <INFO>
javax.enterprise.system.container.cmp      <INFO>
javax.enterprise.system.container.ejb      <INFO>
javax.enterprise.system.container.ejb.mdb  <INFO>
javax.enterprise.system.container.web     <INFO>
javax.enterprise.system.core.classloading  <INFO>
javax.enterprise.system.core.config       <INFO>
javax.enterprise.system.core.naming       <INFO>
javax.enterprise.system.core.security     <INFO>
javax.enterprise.system.core.selfmanagement <INFO>
javax.enterprise.system.core.transaction  <INFO>
javax.enterprise.system                  <INFO>
javax.enterprise.system.tools.admin       <INFO>
javax.enterprise.system.tools.backup      <INFO>
javax.enterprise.system.tools.deployment  <INFO>
javax.enterprise.system.util              <INFO>
javax.enterprise.system.webservices.registry <INFO>
javax.enterprise.system.webservices.rpc   <INFO>
javax.enterprise.system.webservices.saaj  <INFO>
javax                                    <INFO>

```

```
org.apache.catalina <INFO>
org.apache.coyote <INFO>
org.apache.jasper <INFO>
org.jvnet.hk2.osgiadapter <INFO>
```

For information about how to display logger namespaces and log levels, see [“To List Log Levels”](#) on page 27.

Logging Targets

Each instance in an GlassFish Server Open Source Edition domain has a dedicated log file, and each instance and cluster has its own logging properties file. To configure logging for an instance or a cluster, GlassFish Server Open Source Edition allows you *target* specific log files or logging properties files when you do the following:

- Set global or module-specific log levels
- Rotate log files or compress them into a ZIP archive
- Change logging property attributes
- List log levels or log attributes

The following subcommands optionally accept a target specification. A target can be a configuration name, server name, cluster name, or instance name, and is specified as either an operand or as a value passed using the `--target` option. If no target is specified when using any of these subcommands, the default target is the DAS.

Subcommand	Description	Target Specification
<code>collect-log-files(1)</code>	Collects all available log files into a ZIP archive.	<code>--target=target-name</code>
<code>list-log-attributes(1)</code>	Lists logging attributes in the logging properties file.	<i>target-name</i> operand
<code>list-log-levels(1)</code>	Lists the loggers in the logging properties file and their log levels.	<i>target-name</i> operand
<code>rotate-log(1)</code>	Rotates the log file by renaming it and creating a new log file to store new messages.	<code>--target=target-name</code>
<code>set-log-attributes(1)</code>	Sets the specified logging attributes in the logging properties file.	<code>--target=target-name</code>
<code>set-log-levels(1)</code>	Sets the log level for one or more loggers listed in the logging properties file.	<code>--target=target-name</code>

Logging Properties

The DAS as well as each configuration, instance, and cluster has its own set of logging properties that are maintained in individual configuration files. A logging properties file is named `logging.properties` and includes the following information:

- Log file name and location
- Logger names and levels
- Properties for custom handlers
- Log rotation and logger format properties

By default in an GlassFish Server Open Source Edition domain, logging properties files are created in the following locations:

Target	Default Location of Logging Properties File
DAS	<i>domain-dir/config/logging.properties</i>
A configuration	<i>domain-dir/config/config-name/logging.properties</i> , where <i>config-name</i> represents the name of a configuration that is shared by one or more instances or clusters.
An instance	<i>domain-dir/config/instance-name-config/logging.properties</i> , where <i>instance-name</i> represents the name of the instance.
A cluster	<i>domain-dir/config/cluster-name-config/logging.properties</i> , where <i>cluster-name</i> represents the name of the cluster.

For information about configuring logging properties, see [“Configuring the Logging Service” on page 24](#).

Configuring the Logging Service

This section contains the following topics:

- [“Changing the Name and Location of Logging Service Files” on page 24](#)
- [“Setting Log Levels” on page 27](#)
- [“Setting Log File Rotation” on page 30](#)
- [“Adding a Custom Logging Handler” on page 34](#)

Changing the Name and Location of Logging Service Files

This section explains how to change the name and location of the following logging service files:

- Log file
- Logging properties file

▼ To Change the Name and Location of the Log File

To change the name and location of the log file, first use the `list-log-attributes` subcommand to obtain the current log attribute setting for the log file name and location. Then use the `set-log-attributes` subcommand to specify the new name or location. The default target for these two subcommands is the DAS. However, you can optionally specify one of the following targets:

- Configuration name — to target all instances or clusters that share a specific configuration name.
- Server name — to target only a specific server.
- Instance name — to target only a specific instance.
- Cluster name — to target only a specific cluster.

1 Ensure that the target server or cluster is running.

Remote subcommands require a running server.

2 Use the `list-log-attributes(1)` subcommand in remote mode to obtain the current log attribute settings.

The name and location of the log file is set with the `com.sun.enterprise.server.logging.GFFileHandler.file` attribute of the logging properties file. Optionally you can target a configuration, server, instance, or cluster. If you do not specify a target, the log attribute settings for the DAS are displayed.

3 Use the `set-log-attributes(1)` subcommand in remote mode to define a custom name or location of the log file.

If you do not specify a target, the log file for the DAS is targeted by default. If you target a cluster, the name of the cluster log file for each member instance can be changed (the server log file name cannot).

Example 7-1 Changing the Name and Location of a Cluster's Log File

This example changes the name of the cluster log file for Cluster1 to `cluster1.log`. Cluster1 has two server instances: `ClusterServer1` and `ClusterServer2`.

```
asadmin> list-log-attributes Cluster1
com.sun.enterprise.server.logging.GFFileHandler.alarms <false>
com.sun.enterprise.server.logging.GFFileHandler.file
<${com.sun.aas.instanceRoot}/logs/server.log>
com.sun.enterprise.server.logging.GFFileHandler.flushFrequency <1>
.
.
log4j.logger.org.hibernate.validator.util.Version <warn>
Command list-log-attributes executed successfully.
asadmin> set-log-attributes --target Cluster1
```

```
com.sun.enterprise.server.logging.GFFileHandler.file=
${com.sun.aas.instanceRoot}/logs/cluster1.log
com.sun.enterprise.server.logging.GFFileHandler.file
logging attribute set with value ${com.sun.aas.instanceRoot}/logs/cluster1.log
These logging attributes are set for Cluster1.
ClusterServer1 :
com.sun.enterprise.server.logging.GFFileHandler.file
logging attribute set with value ${com.sun.aas.instanceRoot}/logs/cluster1.log
These logging attributes are set for Cluster1.

ClusterServer2 :
com.sun.enterprise.server.logging.GFFileHandler.file
logging attribute set with value ${com.sun.aas.instanceRoot}/logs/cluster1.log
These logging attributes are set for Cluster1.

Command set-log-attributes executed successfully.
```

See Also You can view the full syntax and options of these subcommands by typing `asadmin help list-log-levels` and `asadmin help set-log-attributes` at the command line.

▼ To Change the Name and Location of the Logging Properties File

You can set the name and location of the logging properties file by setting the `java.util.logging.config.file` system property. By setting this system property, you can have a single logging properties file that is used by all instances running on the same host.

Note – Setting the `java.util.logging.config.file` system property causes all other GlassFish Server Open Source Edition logging properties files on the host to be overridden.

1 Set the `java.util.logging.config.file` system property.

For example, you can use the following `java` command:

```
java -Djava.util.logging.config.file=properties_file
```

Alternatively, you can use the Administration Console to set this system property.

2 To apply your change, restart GlassFish Server Open Source Edition.

Example 7-2 Setting the `java.util.logging.config.file` System Property

The following example changes the location of the logging properties file to `/space/mylogging/logging.properties`:

```
java -Djava.util.logging.config.file=/space/mylogging/logging.properties
```


Setting Log Levels

The *log level* determines the granularity of the message that is logged, from error only (SEVERE) to detailed debug (FINEST). The following values apply: SEVERE, WARNING, INFO, CONFIG, FINE, FINER, and FINEST. These log levels are hierarchically inclusive, which means that if you set a particular log level, such as INFO, the messages that have log levels above that level (SEVERE and WARNING) are also included. If you set the log level to the lowest level, FINEST, your output includes all the messages in the file. The default setting is INFO.

You can list current log levels of all loggers specified in the `logging.properties` file. In some cases, loggers that have not been created by the respective containers will also appear in the list.

You can set log levels that are global or logger-specific. When you set a global log level, the log level goes into effect for all loggers. If you set the log level for a specific logger that is different from the global log level, the logger-specific setting takes precedence. In addition, when setting log levels, you can target a configuration, server, instance, or cluster.

Because setting log levels is a dynamic operation, you do not need to restart GlassFish Server Open Source Edition for changes to take effect.

Setting either global or logger-specific log levels is done by using the `set-log-levels` subcommand. Listing log levels is done by using the `list-log-levels` subcommand.

The following topics are addressed here:

- [“To List Log Levels” on page 27](#)
- [“To Set the Global Log Level” on page 28](#)
- [“To Set Module Log Levels” on page 29](#)

▼ To List Log Levels

GlassFish Server Open Source Edition provides the means to list all loggers and their log levels. Listing the loggers provides a convenient means to view current loggers and log levels either prior to or after making log level changes.

Use the `list-log-levels` subcommand in remote mode to list the modules and their current log levels. The default target for this subcommand is the DAS. However, you can optionally specify one of the following targets:

- Configuration name — to target all instances or clusters that share a specific configuration name.
- Server name — to target a specific server.
- Instance name — to target a specific instance.
- Cluster name — to target a specific cluster.

1 Ensure that the DAS is running.

Remote subcommands require a running server.

- 2 List the existing module loggers and log levels by using the `list-log-levels(1)` subcommand.

Example 7-3 Listing Logger Levels for Modules

This example shows a partial list of the existing loggers and their log levels in the DAS.

```
asadmin> list-log-levels
javax.enterprise.system.container.cmp <INFO>
javax.enterprise.system.tools.admin <INFO>
java.util.logging.ConsoleHandler <FINEST>
javax.enterprise.system.container.web <INFO>
javax.enterprise.system.util <INFO>
javax.enterprise.resource.webcontainer.jsf.timing <INFO>
javax <INFO>
javax.enterprise.resource.corba <INFO>
...
Command list-log-levels executed successfully.
```

Example 7-4 Listing Log Levels for an Instance

This example shows a partial list of the loggers and log levels for the instance MyServer2.

```
asadmin> list-log-levels MyServer2
java.util.logging.ConsoleHandler <FINEST>
javax.enterprise.resource.corba <INFO>
javax.enterprise.resource.javamail <INFO>
javax.enterprise.resource.jdo <INFO>
javax.enterprise.resource.jms <INFO>
javax.enterprise.resource.jta <INFO>
javax.enterprise.resource.resourceadapter <INFO>
javax.enterprise.resource.sqltrace <FINE>
...
Command list-log-levels executed successfully.
```

See Also You can view the full syntax and options of the subcommand by typing `asadmin help list-log-levels` at the command line.

▼ To Set the Global Log Level

The *global log level* specifies the events that are logged across all loggers. The default level for messages output to the console is `INFO` (which also includes `SEVERE` and `WARNING` messages).

Use the `set-log-levels` subcommand in remote mode to set the global log level. The default target for this subcommand is the DAS. However, you can optionally specify one of the following targets using the `--target` option:

- Configuration name — to target all instances or clusters that share a specific configuration name.
- Server name — to target a specific server.
- Instance name — to target a specific instance.

- Cluster name — to target a specific cluster.

- 1 **Ensure that the target server or cluster is running.**
- 2 **Set the global log level by using the `set-log-levels(1)` subcommand, specifying the log level of the `java.util.logging.ConsoleHandler` logger.**

The `ConsoleHandler` has a separate log level setting that limits the messages that are displayed. For example:

```
java.util.logging.ConsoleHandler <FINEST>
```

Example 7-5 Changing the Global Log Level for All Module Loggers

By setting the log level of the `ConsoleHandler`, you set the global log level for all loggers. This example sets the global log level in the DAS to `INFO`:

```
asadmin> set-log-levels java.util.logging.ConsoleHandler=INFO
java.util.logging.ConsoleHandler package set with log level INFO.
These logging levels are set for server.
```

Command `set-log-levels` executed successfully.

See Also You can view the full syntax and options of the subcommand by typing `asadmin help set-log-levels` at the command line.

▼ To Set Module Log Levels

A *module log level* specifies the events that are logged for a particular logger. The default level for messages output to the console is `INFO` (which also includes `SEVERE` and `WARNING` messages). The global log level is overridden by a module-specific log level.

By default, the module log level is set to `FINE`. The lines for the loggers in the logging properties file might look like this (the modules are indicated in bold):

```
javax.enterprise.system.tools.level=FINE
javax.enterprise.system.container.ejb.level=FINE
javax.enterprise.system.core.security.level=FINE
javax.enterprise.system.tools.admin.level=FINE
javax.enterprise.level=FINE
javax.enterprise.system.container.web.level=FINE
```

Because setting log levels is a dynamic operation, you do not need to restart GlassFish Server Open Source Edition for changes to take effect.

- 1 **Ensure that the target server or cluster is running.**
Remote subcommands require a running server.
- 2 **List the existing module loggers and their log levels by using the `list-log-levels(1)` subcommand.**

3 Set the log level for a module by using the `set-log-levels(1)` subcommand.

Your choices are SEVERE, WARNING, INFO, CONFIG, FINE, FINER, and FINEST.

Example 7-6 Setting the Log Level for a Module Logger

This example sets the log level for the web container logger to WARNING on the target instance `ManagedServer1`:

```
asadmin> set-log-levels --target ManagedServer1
javax.enterprise.system.container.web=WARNING
javax.enterprise.system.container.web package set with log level WARNING.
These logging levels are set for ManagedServer1.
ManagedServer1 :
javax.enterprise.system.container.web package set with log level WARNING.
These logging levels are set for ManagedServer1.

Command set-log-levels executed successfully.
```

Example 7-7 Setting Log Levels for Multiple Loggers

The following example sets the log level for security and web container loggers in the DAS.

```
asadmin> set-log-levels javax.enterprise.system.core.security=FINE:
javax.enterprise.system.container.web=WARNING
javax.enterprise.system.container.web package set with log level WARNING.
javax.enterprise.system.core.security package set with log level FINE.
These logging levels are set for server.

Command set-log-levels executed successfully.
```

See Also You can view the full syntax and options of the subcommand by typing `asadmin help set-log-levels` at the command line.

Setting Log File Rotation

As explained in “[Log Files](#)” on page 20, GlassFish Server Open Source Edition by default rotates log files when they reach 2 MB in size. However, you can change the default rotation settings. For example, you can change the file size at which the server rotates the log file or you can configure a server to rotate log files based on a time interval. In addition to changing when rotation occurs, you can also:

- Specify the maximum number of rotated files that can accumulate.
 - By default, GlassFish Server Open Source Edition does not limit the number of rotated log files that are retained. However, you can set a limit. After the number of log files reaches this limit, subsequent file rotations delete the oldest rotated log file.
- Rotate the log file manually.

A *manual rotation* forces the immediate rotation of the target log file.

Changing the default log rotation settings is done using the `set-log-attributes` subcommand, and rotating log files manually is done using the `rotate-log` subcommand, as explained in the following sections:

- “To Change the Rotation File Size” on page 31
- “To Change the File Rotation Interval” on page 32
- “To Change the Limit Number of Retained Files” on page 32
- “To Rotate Log Files Manually” on page 33

▼ To Change the Rotation File Size

Use the `set-log-attributes` subcommand in remote mode to change the log rotation file size. The default target of this subcommand is the DAS. Optionally, you can target a configuration, server, instance, or cluster. The minimum size that can be set is 500 KB.

- 1 Ensure that the target server or cluster is running.
- 2 Change the rotation file size limit by using the `set-log-attributes(1)` subcommand, specifying the following attribute and the desired limit in bytes:
`com.sun.enterprise.server.logging.GFFileHandler.rotationLimitInBytes=bytes`
- 3 To apply your change, restart GlassFish Server Open Source Edition.

Example 7-8 Changing the Rotation Size

The following example sets the log file rotation size to 1 MB for the standalone instance `ManagedServer1`:

```
asadmin> set-log-attributes --target ManagedServer1
com.sun.enterprise.server.logging.GFFileHandler.rotationLimitInBytes=1000000
com.sun.enterprise.server.logging.GFFileHandler.rotationLimitInBytes
logging attribute set with value 1000000.
These logging attributes are set for ManagedServer1.
ManagedServer1 :
com.sun.enterprise.server.logging.GFFileHandler.rotationLimitInBytes
logging attribute set with value 1000000.
These logging attributes are set for ManagedServer1.

Command set-log-attributes executed successfully.
```

See Also You can view the full syntax and options of the subcommand by typing `asadmin help set-log-attributes` at the command line.

▼ To Change the File Rotation Interval

Use the `set-log-attributes` subcommand in remote mode to change the log file rotation time limit interval. The default target of this subcommand is the DAS. Optionally, you can target a configuration, server, instance, or cluster. The default value is 0.

- 1 Ensure that the target server or cluster is running.
- 2 Change the rotation time limit by using the `set-log-attributes(1)` subcommand, specifying the following attribute and the desired limit in minutes:
`com.sun.enterprise.server.logging.GFFileHandler.rotationTimelimitInMinutes=minutes`
- 3 To apply your change, restart GlassFish Server Open Source Edition.

Example 7-9 Changing the Rotation Interval

The following example sets the log file rotation time limit for the cluster `Cluster1`, which has the instances `ClusterServer1` and `ClusterServer2`.

```
asadmin> set-log-attributes --target Cluster1
com.sun.enterprise.server.logging.GFFileHandler.rotationTimelimitInMinutes=10
com.sun.enterprise.server.logging.GFFileHandler.rotationTimelimitInMinutes
logging attribute set with value 10.
These logging attributes are set for Cluster1.
ClusterServer1 :
com.sun.enterprise.server.logging.GFFileHandler.rotationTimelimitInMinutes
logging attribute set with value 10.
These logging attributes are set for Cluster1.

ClusterServer2 :
com.sun.enterprise.server.logging.GFFileHandler.rotationTimelimitInMinutes
logging attribute set with value 10.
These logging attributes are set for Cluster1.

Command set-log-attributes executed successfully.
```

See Also You can view the full syntax and options of the subcommand by typing `asadmin help set-log-attributes` at the command line.

▼ To Change the Limit Number of Retained Files

Use the `set-log-attributes` subcommand in remote mode to change the limit on the number of log files that the server creates to store old log messages. The default target of this subcommand is the DAS. Optionally, you can target a configuration, server, instance, or cluster. The default limit value is 0, which results in no limit placed on the number of rotated log files that are retained.

- 1 Ensure that the target server or cluster is running.

- 2 **Change the limit number of retained log files by using the `set-log-attributes(1)` subcommand, specifying the following attribute and the desired file limit number:**

```
com.sun.enterprise.server.logging.GFFileHandler.maxHistoryFiles=minutes
```

The behavior of the

`com.sun.enterprise.server.logging.GFFileHandler.maxHistoryFiles` attribute is as follows:

- If the property is not set, GlassFish Server Open Source Edition keeps a maximum of 10 rotated log files.
- If the property is set to an invalid number or null, GlassFish Server Open Source Edition keeps a maximum of 10 rotated log files.
- If the property is set to 0, GlassFish Server Open Source Edition retains all rotated log files (that is, no sets no maximum).

- 3 **To apply your change, restart GlassFish Server Open Source Edition.**

Example 7-10 Changing the Limit Number of Retained Files

The following example sets the log limit number of retained log files for the DAS to 10.

```
asadmin> set-log-attributes
com.sun.enterprise.server.logging.GFFileHandler.maxHistoryFiles=10
com.sun.enterprise.server.logging.GFFileHandler.maxHistoryFiles
logging attribute set with value 10.
These logging attributes are set for server.
Command set-log-attributes executed successfully.
```

See Also You can view the full syntax and options of the subcommand by typing `asadmin help set-log-attributes` at the command line.

▼ To Rotate Log Files Manually

You can rotate log files manually by using the `rotate-log` subcommand in remote mode. The default target of this subcommand is the DAS. Optionally, you can target a configuration, server, instance, or cluster. When you use this subcommand, the target log file is immediately moved to a new time-stamped file and a new log file is created.

Because log rotation is a dynamic operation, you do not need to restart GlassFish Server Open Source Edition for changes to take effect.

- 1 **Ensure that the target server or cluster is running.**
- 2 **Rotate log files by using the `rotate-log(1)` subcommand.**

Example 7-11 Rotating Log Files Manually

The following example rotates the `server.log` file for `ManagedServer2` to `server.log_yyyy-mm-ddThh-mm-ss`, where `yyyy-mm-ddThh-mm-ss` represents the time when the file is rotated, and creates a new `server.log` file in the default location.

```
asadmin> rotate-log --target ManagedServer2
Command rotate-log executed successfully.
```

See Also You can view the full syntax and options of the subcommand by typing `asadmin help rotate-log` at the command line.

Adding a Custom Logging Handler

By default, GlassFish Server Open Source Edition log records are captured in a server log file using the format described in “[Log Records](#)” on page 21. However, you may find that you want to log messages to a different location, such as a database or a remote server, or log messages from specific loggers to your own file. This can be done by implementing a custom log handler. This section explains how to add a custom log handler to the GlassFish Server Open Source Edition logging service.

▼ To Add a Custom Log Handler

A comma-separated list of log handlers is installed during startup of the Java Virtual Machine (JVM) host. The default log handler that is provided in the `logging.properties` file, `ConsoleHandler`, is configured as follows:

```
handlers=java.util.logging.ConsoleHandler
```

In GlassFish Server Open Source Edition, the best approach to developing a custom handler is to define a Hundred-Kilobyte Kernel (HK2) component that implements the handler contract. GlassFish Server Open Source Edition registers this handler automatically because it is an HK2 component. There is no task required of the administrator.

To configure a custom handler that is not developed as an HK2 component, add the new handler to the `logging.properties` file after the developer has put the custom handler JAR file into the `domain-dir/lib/ext` directory.

Before You Begin If you set a handler by setting the `handlers` attribute in the logging properties file, the class that extends `java.util.logging.Handler` must be in the server classpath.

1 Ensure that the target server or cluster is running.

Remote subcommands require a running server.

- 2 Use the **set-log-attributes(1)** subcommand to add the handler to the **handlers** attribute.
The default target of this subcommand is the DAS. Optionally you can target a configuration, server, instance, or cluster.
- 3 To apply your changes, restart GlassFish Server Open Source Edition.
See “To Restart a Domain” on page .

Example 7–12 Adding a New Log Handler

This example adds the custom logger `com.example.logging.MyHandler` to the logging properties file of the DAS.

```
asadmin> set-log-attributes
handlers=java.util.logging.ConsoleHandler,com.example.logging.MyHandler
handlers logging attribute set with value
java.util.logging.ConsoleHandler,com.example.logging.MyHandler.
These logging attributes are set for server.
Command set-log-attributes executed successfully.
```

See Also You can view the full syntax and options of the subcommand by typing `asadmin help set-log-attributes` at the command line.

Viewing Log Records

The recommended means for general viewing of logging information is to use the Log Viewer in the Administration Console. The Log Viewer simplifies reading, searching, and filtering log file contents. For instructions, see the Administration Console online help.

GlassFish Server Open Source Edition also allows you to collect log files into a ZIP archive, which provides the means to obtain and view log files for an instance or cluster even when it is not currently running. The following section explains how to collect all available log files for an instance or cluster and compile them into a single ZIP archive, which is done by using the `collect-log-files` subcommand.

▼ To Collect Log Files into a ZIP Archive

Use the `collect-log-files` subcommand in remote mode to collect log files into a ZIP archive. The default target of this subcommand is the DAS. Optionally you can target a configuration, server, instance, or cluster.

- 1 Ensure that the target server or cluster is running.
Remote subcommands require a running server.

2 Use the `collect-log-files(1)` subcommand to create the ZIP archive.

The default location in which the ZIP archive is created is the *domain-dir/collected-logs* directory. The `collect-log-files` subcommand allows you to specify a nondefault directory in which the ZIP archive is to be created by using the `--retrieve` option set to `true`, followed by the directory name.

The name of the ZIP file contains the timestamp, as follows:

```
log_yyyy-mm-dd_hh-min-sec.zip
```

Example 7-13 Creating a ZIP Archive

This example shows collecting the log files for the cluster `MyCluster` and compiling them into a ZIP archive in the `/space/output` directory.

```
asadmin> collect-log-files --target MyCluster
--retrieve true /space/output
Log files are downloaded for ClusterServer1.
Log files are downloaded for ClusterServer2.
Created Zip file under /space/output/log_2011-02-10_13-35-25.zip.
Command collect-log-files executed successfully.
```

When the ZIP file created by the preceding command is uncompressed, the following directory structure is created:

```
as-install-parent/
  glassfish/
    domains/
      domain-name/
        collected_logs/
          logs/
            ClusterServer1/
              server.log
            ClusterServer2/
              server.log
```

See Also You can view the full syntax and options of the subcommand by typing `asadmin help collect-log-files` at the command line.

P A R T I I

Resources and Services Administration

