One Pager: GlassFish v3 Logging

Table of Contents

1. Introduction

1.1. Project/Component Working Name

Logging

1.2. Name(s) and e-mail address of Document Author(s)/Supplier

Carla Mott, carla.mott@sun.com
Jerome Dochez, jerome.dochez@sun.com

1.3. Date of This Document

Initial Draft: Wed Dec 3 12:55:07 PST 2008

Review complete: Mon Mar 9 17:37:22 PDT 2009

2. Project Summary

2.1. Project Description

Provide a logging facility for GlassFish application server.

2.2. Risks and Assumptions

The logging facility for GlassFish V3 will leverage the JDK logging utility as much as possible. This means that there will be a new property file, logging.properties, created which will contain configuration information.

3. Problem Summary

3.1. Problem Area

Developers and adminstrators need a way of determining the health of the server. One way to do so is by analyzing the server logs. The logging facility logs message from the application server to a file or set of files that can then be analysed used tools or scripts. The facility allows for specifying the granularity or level of messages to be logged for the different modules, specifying the name of the file where the messages will be stored and specifying the time or size limit that will cause the facility to rotate the log file. Logging to syslog is also supported on Solaris.

3.2. Justification

Simplify the logging code to leaverage the JDK logging utility as much as possible. Carry forward existing v2 functionality for logging.

4. Technical Description

4.1. Details

Logging has long been an important part of GlassFish. In v3 the logging facility has been rewritten to address some performance issues and to leverage the JDK logging utility as much as possible. In the new implementation, reconfiguration is done using the JDK logging APIs and facilities where possible to simplify the code. As part of this effort there is a new configuration file called logging.properties which is described by the JDK logging utility. The main motivation behind adding the logging.properties file in GlassFish V3 is to leverage the JDK logging facility as much as possible. Many developers, administrators and users are familiar with the JDK logging utility and the format of the logging.properties file. The JDK logging utility is similar to other logging facilities such as log4j which are popular with many Java developers and the format and concepts are well known and understood. Using the logging.properties file will allow developers to easily add handlers, formatters and filters to enhance or customize the GlassFish logging facility. There is a fair amount of existing documentation and examples that describe how to implment these features and are applicable in GlassFish v3.

Although most of the code in v3 logging is new, the implementation does include a small amount of code that is present in previous releases of GlassFish. Specifically the code that rotates the log file based on size or time is preserved because GlassFish allows users to rotate log files based on a time value in addition to number of bytes. A significant change in the new code base is that all logger level configuration is now handled through the logging.properties file and dependencies on domain.xml will be removed. Configuration properties for rotating the log files based on size of the file or the elapsed time are added to the logging.properties file as well all other configuration properties found under the log-service element in domain.xml. See 4.5 Interfaces section below for more information.

The v3 logging facility is implemented as a service and therefore implments the HK2 service APIs. The service is an init service which it is initialized as one of the first services at startup.. At that time any user or system defined handlers are added to the root logger, the server log file is created and standard out and standard err are redirected to a logger. In addition, properties that previously were in domain.xml are "moved" to logging.properties file if needed and the logging facility is initialized using the logging.properties file. Updating the log level on the loggers is done dynamically with the aid of the file monitoring functionality. The log service is notified when there is a change to the logging.properties file (usually file has been saved) and will lookup the known loggers in the file and set the log level to the level specified in the file. All GlassFish loggers are set the level INFO by default.

The description of properties found in the logging.properties file is found in section 4.5 Interfaces. The default logger uses those properties to configure the logging facility. Only the log levels for the loggers can changed dynamically. That is no server restart is required for the new values to take affect. All changes to other properties do require the server to be restarted.

In v2 users use the admin GUI or cli to configure the log file name or change the logger levels. That functionality remains for v3 from the users point of view. Because logging.properties is used to maintain the configuration, any logging related elements that are in the domain.xml file for v2 will be moved to the logging.properties file for v3. Updates to the GlassFish loggers through the Admin GUI and the CLI will result in a change to the logging.properties file. Updating the logging.properties file is the responsibility of the logging facility and done with the aid of service object and new API. See section 4.5 Interfaces for details on the new API.

To help improve logging performance, log messages are now written to a queue and that queue is emptied to the log file as time permits. The size of the queue is configurable through the properties file but may fill up if the user has requested log level FINEST and there is a lot of activity on the server. The code does detect if the queue is full and blocks until messages can been written out to the file. Initial tests show that messages can be processed reasonably although heavy load on the server can result in a bottleneck.

4.2. Bug/RFE Number(s)

6668, 6148 specifically.

4.3. In Scope

Software needed to produce logs files that are suitable for analysis by end users, system administrators, developers ro field service engineers. The logs capture information such as security failures, configuration errors, diagnostic messages and/or bugs in the server.

4.4. Out of Scope

Actual implementation of the tools that allow users to modify the logging properties are outside of the scope of this document. That includes the Admin GUI, Admin cli commands and the Log Viewer graphical interface. It is also the responisbility of the various modules to use the logging facility to report status or failures using the log level guidelines of GlassFish v3.

4.5. Interfaces

4.5.1 Exported Interfaces

Interface	Stability	Former Stability (if changing)	Class Name	Comments
public static Logger getLogger(final Class clazz, final String name)	Committed		com.sun.logging.LogDomains	All modules in GlassFish use this interface to create loggers. clazz is used to locate the resource bundle for the module and the logger name specifies which logger to return.
String setLoggingProperty(String propertyName, String propertyValue)	Committed		com.sun.logging.LoggingConfigImpl	set specified logging property name to the specified value
Map updateLoggingProperties(Map properties)	Committed		com.sun.logging.LoggingConfigImpl	set specified logging properties in the hashtable with corresponding values
Map getLoggingProperties()	Committed		com.sun.logging.LoggingConfigImpl	get all the properties in the logging properties
void removeLoggingProperties(Set properties)	Committed		com.sun.logging.LoggingConfigImpl	remove the properties listed in properties from the logging.properties file

4.5.2 Imported interfaces

Interface	Stability	Exporting Project: Name, Specification or other Link.	Comments
HK2	Committed	org.jvnet.hk2.*	logging is implemented as a HK2 service, server-config for notifications on changes from admin

4.5.3 Other interfaces (Optional)

Interface	Stability	Exporting Project: Name, Specification or other Link.	Comments
logging.properties file	Committed	external	Used internally to maintain current log configuration. Updated by logging facility

The logging.properties file contains all properties related to logging. This file contains all the logger names and levels, as well as the properties to add custom handlers and properties to specify log file rotation. The properties that need to be moved from domain.xml to logging.properteis are listed here.

domain.xml name under log-service	logging.properties name	default value
file	com.sun.enterprise.server.logging.GFFileHandler.file	logs/server.log
use-system-logging	com.sun.enterprise.server.logging.SyslogHandler.useSystemLogging	false
log-handler	handlers	java.util.logging.ConsoleHandler
log-filter	com.sun.enterprise.server.logging.GFFileHandler.logFilter	N/A
log-to-console	com.sun.enterprise.server.logging.GFFileHandler.logtoConsole	false
log-rotation-limit-in-bytes	com.sun.enterprise.server.logging.GFFileHandler.rotationLimitInBytes	2000000
log-rotation-timelimit-in-minutes	com.sun.enterprise.server.logging.GFFileHandler.rotationTimelimitInMinutes	0
retain-error-statistics-for-hours	com.sun.enterprise.server.logging.GFFileHandler.retainErrorsStasticsForHours	0

Custom handlers are added to the root logger and available to all applications running on the server. Users can also specify a custom filter which is added to each logger known to the system at startup.

The following properties are new with the v3 release.

Interface	Stability	Default Value	logging.properties name	Comments
			Number of entries the log	

max-queue-size	Committed	5000	queue will hold. When full the system will block until a record is written to the log file. The server must be restarted when this number is changed.
flush-frequency	Committed	1	This is used to specify the max number of messages to be written from the queue out to the file at once. The default value is 1

4.6. Doc Impact

None.

4.7. Admin/Config Impact

The configuration properties are now stored in the logging.properties file. Admin console and cli will need to use the new API to update the property value. There are a few new properties that should be supported by the tools. Specifically, **MaxQueueSize** and **FlushFrequency**

4.8. HA Impact

None.

4.9. I18N/L10N Impact

No new requirements.

4.10. Packaging & Delivery

None.

4.11. Security Impact

None.

4.12. Compatibility Impact

The new version of the logging code uses the logging.properties file as described in sections 4.1 and 4.5.3.

4.13. Dependencies

None.

5. Reference Documents

// List of related documents, if any (BugID's, RFP's, papers, Blogs).
// Explain how/where to obtain the documents, and what each
// contains, not just their titles.

6. Schedule

Per GlassFish v3 schedule.