Framework Management Layer User's Guide

Logging Functions

5/3/2025

Logging Functions

Contents

The Management Layer collects Genesys application logs of defined levels and stores them in a centralized database.

Log Levels

Genesys applications can report log events at five levels of detail: *Alarm*, *Standard*, *Interaction*, *Trace*, and *Debug*. Only the first four are intended for on-site analysis by a user. Log events of the Alarm, Standard, Interaction, and Trace levels feature the same unified log record format and can be stored in the Centralized Log Database.

Logging During Normal Operation

For complete specifications of log events reported at the Alarm, Standard, Interaction, and Trace levels, see *Framework Combined Log Events Help*.

Alarm Log Level

The Alarm-level logs contain only log records of the Alarm level. SCS generates Alarm log events on behalf of other applications when receiving from them log events configured as Detection Events in Alarm Conditions. Using this level, Solution Control Server reports the occurrence and removal of all alarms to the Centralized Log Database.

This level contains the Management Layer translations of log events of other levels into Alarms.

Standard Log Level

Permanently enable *only* the Standard level of logging during solution operation in regular production mode. It contains high-level events that report both major problems and normal operations of in-service solutions.

An event is reported at the Standard level if it satisfies one of these criteria:

- Indicates that an attempt to perform any external operation has failed
- Indicates that the latest attempt to perform an external operation that previously failed has succeeded
- Indicates detection of a condition that has a negative impact on operations, actual or projected
- Indicates that a previously detected condition, which had a negative impact on operations, no longer exists
- Indicates a security violation of any kind
- Indicates a high-level data exchange that cannot be recognized or does not follow the expected logical sequence
- Indicates inability to process an external request
- Indicates successful completion of a logical step in an initialization process
- Indicates a transition of an application from one operational mode to another
- Indicates that the value of a parameter associated with a configurable threshold has exceeded that

threshold

- Indicates that the value of a parameter associated with a configurable threshold that earlier exceeded the threshold has returned to its normal range

Interaction Log Level

The Interaction-level logs report the details of an interaction processed by solution components that handle interactions. The log contains information about the processing steps for each interaction by each solution component.

An event is reported at the Interaction level if it:

- Is a recognizable high-level data exchange with another application about an interaction.
- Indicates a change in real-time state of an interaction handled by the application (unless such a change is visible from the high-level data exchange).

The specific criteria depend on a particular component and its role in interaction processing.

Use the Interaction-level log records to analyze and troubleshoot new interaction-processing scenarios, especially when you introduce new solutions or enable new functions within existing solutions. Note that Interaction-level records contain the interaction attributes, such as Interaction ID, that you can use to search for log events related to the same interaction but generated by different applications.

> **Warning**
> Using the Interaction level generates a higher number of loggingevents on the network and that may adversely affect the performance of the DBMS, Message Servers, and interaction-processing components.

Trace Log Level

The Trace-level logs report the details of communications between the various solution components. The log contains information about the processing steps for each interaction by each solution component.

An event is reported at the Trace level if it satisfies one of these criteria:

- It is a recognizable high-level data exchange with another application.
- It is a recognizable high-level data exchange with an external system.
- It indicates a change in real-time state of user-level objects that the application handles (unless such a change can be seen from the high-level data exchange).

Use the Trace-level log records to analyze and troubleshoot new interaction-processing scenarios, especially when you introduce new solutions or enable new functions within existing solutions.

> **Warning**
> Using the Trace level generates a higher number of logging events on the network and that may adversely affect performance of the DBMS, Message Servers, and interaction-processing components.

Logging During Irregular Operation

Standard-level, Interaction-level, and Trace-level log events do not contain all the details you or someone else may need to analyze and troubleshoot solutions malfunctions. That is why Genesys Customer Care might ask you to provide relevant Debug-level logs when you request their assistance.

Because the Debug-level log events do not have a unified format, are not documented, and can only be stored in a text file, they are only useful to Genesys Customer Care. Logging at this level is likely to adversely affect application performance. Enable this log level only when a Genesys representative requests it. Keep in mind that running Genesys servers with the Debug level of logging is highly resource-intensive and, as such, is not recommended when you are in production mode.

Before you set a logging level more detailed than the Standard level, carefully consider whether a situation (such as the initial deployment or first signs of technical problems) really calls for it. Then, test for how more-detailed logging affects the network loads in a lab or controlled environment.

Note that changing the log level of a running application does not interrupt solution operations.

In addition to asking for Debug-level log records when you report a problem to them, Genesys Customer Care might also request that you reproduce the problem because:

- During regular operations, many contact-center systems, such as DBMSs, IVRs, and switches, do not employ logging at the level of detail required to diagnose serious technical issues.
- Reasons other than application failure can contribute to interaction-handling errors. For example, a call can be misrouted (delivered to a wrong DN) despite the fact that applications are functioning properly.

Reliability Logging

Starting in release 8.5, Genesys software supports a level of reliability logging, most notably to provide information for estimating product availability. This information is available from some log events, to which appropriate information has been added or made available. At the Management Framework level, three common log events (`00-05090`, `00-05091`, and `00-05064`) have been enhanced with the addition of application name, type, and DBID as extended attributes. Refer to *Framework Combined Log Events Help* for full descriptions of these log events.

Centralized Logging

The centralized logging function provides a number of advantages over the more traditional logging

to a text file:

- Keeping log records of all applications in one place and presenting them in the unified log record format provides for a comprehensive view of the solutions' operations history.
- Using a relational DBMS such as the central log storage enables quick access to the required records and allows for advanced record selections, which you can base on a variety of search criteria.
- Viewing, via Genesys Administrator, the logs stored in a Centralized Log Database gives you an integrated view of the solutions' maintenance history and complements the solution-control and alarming capabilities.
- Deleting the obsolete logs or logs of a particular solution, host, or application makes the Log Database management more convenient.

Given these advantages, Genesys recommends using the centralized logging as the primary method for storing Standard-level log events of all applications. When enabling the log output for Interaction and Trace logs, store log events of both levels in the Centralized Log Database in addition to log events of the Standard level. Genesys does not recommend the simultaneous use of both local and centralized logging options except for some special, temporary purpose.

The centralized logging system consists of:

- One or more Message Servers that collect log events from applications.
- One or more Log Databases.
- Genesys Administrator.

Provided that the Standard level of log output is routinely used under normal production conditions, always limit the centralized logging system to one Message Server and one Log Database for all but large and geographically distributed interaction management networks.

If any part of the centralized logging system becomes unavailable, the log outputs of the affected applications are temporarily redirected to local binary files. Upon restoration of normal functioning, the applications automatically resume logging to the Centralized Log Database. The log records accumulated in the local binary files are automatically transferred to the Log Database.

If the connection between the Message Server and the Log Database is unavailable, messages are stored in a queue. When the connection is restored, the messages in the queue are written to the Log Database. See the "Message Server" section of the *Framework Configuration Options Reference Manual* for more information.

The format of records kept in the Log Database is specified in Log Formats.

Viewing Log Database Entries

Although you can use any general-purpose DBMS client to make advanced selections from the Log Database, Genesys Administrator's log-viewing capabilities may actually meet your needs just as well. In either interface, you can view an entire log. They also provide a number of predefined selections from the Log Database, which are based on the most typical maintenance-selection criteria:

- Records generated by components of a selected solution.
- Records generated by applications located on a selected host.

- Records of a specified output level.
- Records containing a specified combination of symbols in text.
- Records generated within a specified time interval.
- Records containing specified values of certain extended attributes.

You can use these predefined selection criteria in any combination.

To delete obsolete log records, you can use Genesys Administrator.

Logging and Application Performance

Follow these recommendations to increase an application's performance while enabling the application's logging:

- Always enable buffering of the log output when sending logs to a log file. Refer to the "Common Log Options" chapter in the *Framework Configuration Options Reference Manual*.
- Store log files on the local disk of the computer running the application rather than storing them using network file systems. Such systems may not perform very well and the added network traffic for this storage can affect application performance.
- Configure only log events of the Standard level to be sent to the Log Database. For log events of other levels, consider using the memory output as the safest output in terms of application performance.
- Directing log output to the console (by using the `stdout` or `stderr` settings) can affect application performance. Avoid using these log output settings in a production environment.

Logging Resilience

Starting in release 8.5, Genesys logging incorporates additional functionality specifically aimed at maintaining the integrity and usefulness of the logging system, including the Centralized Log, without causing input/output or run-time logging issues from affecting normal operations or causing the application to terminate unexpectedly.

Without this feature, logs are written to output by the application's main thread. Any delay or bottleneck in the log output system takes up processing time and space for non-log operations using that thread.

This problem is greatly resolved, if not eliminated, by the logging resilience feature. The feature is configured at the application level, and consists of two elements:

- An internal log queue, for which a dedicated thread processes log messages for output.
- A throttling mechanism, which only when a performance problem is detected, changes the verbose level of the log system (as specified by the **verbose** option in the `log` section). This element cannot be activated unless the dedicated process thread is active.

When logging resilience is activated (as specified by **enable-thread=true** in the `log` section), all log messages generated by the application are moved to an internal queue, from which they are written

to output using the dedicated thread. This negates output problems backing up into regular processing.

Once the internal queue is established, the throttling mechanism can start (if configured to do so). This throttling mechanism works by monitoring the size of the internal log queue, and changing the value of the verbose option to increase or decrease the number of logs being generated, as follows:

- When the queue size approaches a configured threshold (specified by a non-zero value of the **throttle-threshold** option in the `log` section), the value of the **verbose** option is reduced by one level to reduce the number of log messages generated.
- When the queue size decreases to 50% or less of the threshold, the value of the **verbose** option is increased by one level to increase the number of log messages generated.

The verbose level is maintained at this level until both of the following conditions occur:

- A configured period of time (called the throttle period, and specified by the **throttle-period** option in the `log` section) expires. This timer is reset each time that throttling (up or down) occurs.
- One of:
  - The queue size increases and approaches the threshold, at which point the verbose level is throttled again. This can only occur until **verbose=none**, at which point there are no logs being processed for output.
  - The queue size drops to 50% or less of the threshold, at which point the verbose level is increased by one level. This can only occur until the value of verbose is back to its originally configured value.

Throttling is an optional part of the Logging Resilience feature, it can be disabled or stopped without interfering with the internal log queue.

For detailed descriptions of the three options used for and by this log resiliency functionality, refer to the “Common Configuration Options” chapter of the *Framework Configuration Options Reference Manual*.

Alarms

The Management Layer uses the Centralized Log Database to store detailed and structured information about Alarm activation and clearance. (See Alarm-Signaling Functions for more information about how alarms are generated.) Solution Control Server generates alarm-related information as log events of the Alarm level for each Alarm activation and clearance event. Solution Control Server attaches a set of extended attributes to each Alarm log event; in particular, the ID attribute uniquely identifies each Alarm.

For complete specifications of Alarm log events that SCS and Message Server report, and for information about extended attributes for each log event, see *Framework Configuration Options Reference Manual*.

Audit Trail

The Management Layer also uses the Centralized Log Database to store Audit-Trail records (from here on referred to as Audit records) that Framework components (in particular, Configuration Server and SCS) generate for configuration changes and control actions performed over processes, solutions, and alarms. Starting in release 8.1, the Audit records also include the name of the client application and details about the host on which the client application resides. This is to ensure compliance to Payment Card Industry Data Security Standard (PCI-DSS) 10.3.

Framework components generate Audit records as log events and, if available, attach extended attributes to Audit log events.

For information about setting up an audit trail and viewing the Audit logs, see How to Set Up and Use an Audit Trail.

For complete specifications of Audit log events that Framework components report and for information about extended attributes for each log event, see *Framework Configuration Options Reference Manual*.

History of Configuration Changes

Starting in release 8.5, Configuration Server keeps a history of configuration changes, including a record of new and previous values. For more information, refer to the *Framework Deployment Guide*.

Interaction Tracing

You can configure Framework components to send Interaction-level log events to the Centralized Log Database. You can later retrieve from the database all records related to a certain interaction, enabling you to trace its progress in the contact center.

> **Important**
> Storing Interaction-level log events in the Log Database might affect application performance, so Genesys does not recommend it in production environments.

Framework components might attach a set of extended attributes to each Interaction log event. In particular, each such event contains a unique identifier of the contact center interaction in the IID extended attribute.

> **Important**
> The set of extended attributes for Interaction-level log events may vary depending on a particular interaction's properties and the component that generates the log event.

For complete specifications of Interaction-level log events that Framework components report, and for information about extended attributes for each log event, see Framework Combined Log Events Help, available on the Genesys Documentation wiki.

Use Genesys Administrator to display all Interaction-level records from the Centralized Log Database. Use predefined selection criteria to search for all records with a particular Interaction ID.

For information about viewing Audit records with Genesys Administrator, see Framework Genesys Administrator Help.