Framework Management Layer User's Guide

How to Manage Log Records

5/4/2025

How to Manage Log Records

Contents

You can manage records in the Log Database by:

- Using Genesys Administrator.
- Creating your own scripts for automated database purging.

> **Important**
>
> Log records also contain alarm history. Be careful not to delete current alarm history records when you remove log records from the Log Database.

The log-managing function requires the installation of the same components as for the log-viewing function.

Using the Log Database Maintenance Wizard

You can use the Management Layer logging function to manage log records stored in the Centralized Log Database. With the Log Database Maintenance Wizard available in Genesys Administrator, you can specify criteria—through a custom SQL statement or individual selections—for the search and removal of log records from the database. Available criteria categories include log type, log level, log generation time, log source, or extended attributes. Log source can be a particular application, applications that belong to a particular solution, or applications that run on a particular host computer.

Launch the Log Maintenance Wizard from **Environment > Centralized Log** under the **Monitoring** tab, and follow the instructions for each step. *Genesys Administrator Help* contains more information about the Wizard.

Automating the Purging Functionality

This section describes how to automate the removal of obsolete log records from the Log Database. Database purging involves the periodic, automated execution of appropriate SQL statements within your SQL server.

To enable automated purging:

1. Prepare SQL statements that remove log records. **[+] Show more information**

   As you create SQL statements that delete records from the Log Database tables, keep in mind that these SQL statements must contain one or more criteria for selecting the log records you want to remove. You can base the selection criteria on the values of the log record fields, such as log record generation time, application name, host name, and so forth. For example, you might remove older log messages from the G_LOG_MESSAGES table and their corresponding attributes, if any, from the G_LOG_ATTRS table with the following SQL statements (in the order specified):

   ```
   DELETE FROM G_LOG_ATTRS
   WHERE LRID IN (SELECT G_LOG_MESSAGES.ID
   ```

```
  FROM G_LOG_MESSAGES
  WHERE (TIMEGENERATED > <start datetime>)
  AND (TIMEGENERATED < <end datetime>)
)
DELETE FROM G_LOG_MESSAGES
  WHERE (TIMEGENERATED > <start datetime>)
  AND (TIMEGENERATED < <end datetime>)
```

Refer to the structure description in Log Format for more information about Log Database tables and fields. Combine the selection criteria to achieve the level of purging that suits your environment.

Check the Log Database Maintenance Wizard in Genesys Administrator, for examples of the records-removal SQL statements that the Wizard prepares. The Wizard provides the graphical interface through which you specify various log-records selection criteria, and it displays the resulting SQL statements.

2. Schedule automated execution of the SQL statements. **[+] Show more information**

To enable automated purging of log records, schedule the periodic, automatic execution of the SQL statements you have prepared (for example, once a week). The simplest way to do this is to use either SQL server utilities or operating system services.

<tabber> Using SQL Server Utilities= If you decide to use SQL Server utilities, refer to your SQL Server documentation to determine whether that server provides tools for automatic execution of SQL statements. |-| Using OS Services= If you decide to use scheduling tools available in your operating system, you should:

1. Prepare a command (either an executable file or a batch/shell file) that executes your SQL statements.

2. Use an operating system tool that enables you to schedule the specified command for execution.

To prepare a command that executes your SQL statement(s), use either a batch file or shell script. A command like this usually calls an SQL Server–specific tool to execute command-line SQL statements and passes an SQL statement to this tool as a parameter. For example, you can use the following tools to execute command-line SQL statements:

*   **isql.exe** (a Microsoft SQL tool)

*   **sqlplus** (an Oracle tool)

To schedule a specified command for execution with the required frequency, consider using these tools:

*   `cron` on UNIX platforms

*   `at` on Windows platforms