Framework Management Layer User's Guide

How to Manage Third-Party Applications

5/2/2025

How to Manage Third-Party Applications

Contents

In Genesys terms, a third-party application is an application not instrumented with Genesys libraries. This chapter describes which Management Layer functions you can use with third-party applications and how the Management Layer processes the related commands. It also lists the software prerequisites for and describes how to configure these applications.

Prerequisites

In Genesys terms, a third-party application is an application not instrumented with Genesys libraries. The Management Layer can monitor, start, and stop a third-party application as long as that application:

- Supports startup from a command line.
- Starts if the computer it runs on is unattended (for instance, on a Windows computer with no user logged in); however, this is not mandatory.
- Works without a console window on Windows; however, this is not mandatory.
- Is registered in the Configuration Database as an Application of the Third Party Server type.
- Runs on an operating system that Genesys supports.

Important

You cannot perform the centralized logging and alarm-signaling functions (including switchover) over a third-party application because they require built-in support on the application side.

Required Components

If you have configured third-party applications in the Genesys Configuration Database, Management Layer can control, monitor, start, and stop them. Even if you do not use the Management Layer to start a particular application, the application's runtime status is displayed. This functionality is also supported for:

- Third-party applications installed as Windows Services
- Third-party applications started with a script.

Managing third-party applications requires the installation of:

- Solution Control Server
- An instance of Local Control Agent (LCA) for each host computer running third-party applications
- Genesys Administrator

The monitoring views and control commands are available through Genesys Administrator, just as they are for managing Genesys applications. *Framework Genesys Administrator Help* provides detailed instructions for viewing the applications, and starting and stopping them.

Configuring Third-Party Applications

You must create an Application object for each third-party application that you want to use, and configure it appropriately. **[+] Show steps**

Purpose

To create and configure an Application object for a third-party application using Genesys Administrator.

Prerequisites

- Management Layer components are installed and running.
- The third-party application is installed.
- You are logged in to Genesys Administrator.

Start of procedure

1. Register the third-party application in the Configuration Database as an Application object of the Third Party Server type.

2. In the **Server Info** section of the Application object's **Configuration** tab, specify the following:

   - **Working Directory**— The full path to the directory from which the application starts.

   - **Command Line**— The command line used for starting the application; usually, it is the name of the executable file.

3. (Optional) If you want to start and stop the third-party application using the Management Layer, do the following:

   a. In the **Server Info** section of the Application object's **Configuration** tab, in the **Command Line Arguments** field, specify any additional parameters used to start the application.

   b. If the application is started with a batch file or script, specify the name of the command used for launching that file or script. In the Annex list of the Application object's **Options** tab, create a section named **[start_stop]** and an option named **start_command**.

   c. As a value for this option, specify the command that launches the batch file or script, including the full path to the executed file or script. (The

start_command option may also contain the command to start the executable file.)

d. If the application is stopped with a batch file or script that performs the correct shutdown of the application, specify the name of the command used for launching that file or script. In the Annex list of the Application object's **Options** tab, create (or open) a section named **[start_stop]** and create an option named **stop_command**. For its value, specify the command that launches the batch file or script, including the full path to the executed file or script.

Monitoring Third-Party Applications

Monitoring functionality is provisioned by the ability of Local Control Agent to determine whether:

- A third-party application is started.
- A third-party application is stopped.

Determining Whether Applications Are Started

LCA uses the so-called command–line matching mechanism to determine if a third-party application is started. This means that LCA periodically retrieves a list of all currently running processes, and then compares command lines of all processes from that list with possible command lines of the third-party applications being checked.

LCA uses the following command–line matching rules for this comparison:

- The command line of a process is equal to the set of these elements:
    Working Directory + Command Line [+ Command Line Arguments]
  where **Working Directory**, **Command Line**, and **Command Line Arguments** are the properties of the Application object in the Configuration Database. If the **Command Line Arguments** property is empty, it is not used.
  The processes started with the full path specification are evaluated based on this rule.

- The command line of a process is equal to the set of these elements:
    Command Line [+ Command Line Arguments]
  where **Command Line** and **Command Line Arguments** are the properties of the Application object in the Configuration Database. If the **Command Line Arguments** property is empty, it is not used.
  The processes started without the full path specification are evaluated based on this rule.

- For Windows operating systems only, the command line of a process is equal to the set of these elements:
    + Working Directory + Command Line [+ Command Line Arguments]
  where **Working Directory**, **Command Line**, and **Command Line Arguments** are the properties of the Application object in the Configuration Database. If the property **Command Line Arguments** is empty, it is not used.
  The processes started with the full path specification are evaluated based on this rule when the path contains spaces.

If LCA finds a process whose command line matches that of a third-party application, LCA assumes that the application has started and then:

1. Stores the PID (process identifier) for that application.

2. Sets the application status to Started.

3. Sends a notification to SCS.

Determining Whether Applications Are Stopped

LCA uses the so-called PID-check mechanism to determine if a third-party application is stopped. This means that LCA tracks the PIDs (process identifiers) for all currently running processes. Using relevant operating system commands, LCA determines if a process with a particular PID is running. If not, LCA considers the corresponding third-party application stopped and:

1. Sets the application status to Stopped.

2. Sends a notification to SCS.

Starting Third-Party Applications

You can start a third-party application in the following ways:

- User command from Genesys Administrator

- Alarm Reaction

- Automatically at host start-up

- Auto-Restart

When the Management Layer receives a request to start a particular third-party application, SCS generates a command line and passes it to LCA, which executes the required operating system function.

SCS generates the command line based on the parameters you configured for a specific third-party Application object in the Configuration Database, which may include:

- **Working Directory**—as specified in the Application object's properties.

- **Command Line**—as specified in the Application object's properties.

- **Command Line Arguments**—as specified in the Application object's properties.

- Start Command—as specified by the **start_command** option of the Application object's Annex.

If you have not specified values for the first three listed parameters or have not created a **start_command** option and provided a value for it, the Management Layer cannot start the application. For more information about these parameters, refer to Configuring Third-Party Applications.

Solution Control Server forms the command line as follows:

- If you have specified the **start_command** option, SCS uses its value to form the command line and ignores the other parameters.

- If you have not specified the start_command option, SCS uses the values of the **Command Line** and **Command Line Arguments** to form the command line, while LCA executes an appropriate operating system function in the directory specified as the **Working Directory** for this application.

LCA passes all required parameters to the operating system function (`CreateProcess` on Windows or execvp on UNIX) and calls the function, after which two scenarios can occur:

- The operating system function returns an error. In this case, LCA passes the error to SCS, which retains the Stopped status for the third-party application.

- The operating system function does not return an error. In this case, LCA determines the status of the third-party application and passes the status to SCS. (See Determining Application Status for a description of the methods LCA uses to determine the application status.)

Whichever scenario occurs, the startup process is then considered finished.

Starting Third-Party Applications Automatically

Management Layer supports the automatic start-up of third-party applications. You must be aware of, and correct if necessary, the configuration of the startup timeout for automatically started third-party applications.

The Management Layer must know the correct status of the third-party application at the exact moment when determining if the application is stopped and is to be started, that is, when the startup timeout for the third-party application expires.

Incorrect configuration of the automatic third-party application start-up configuration can cause multiple instances of the same application to be started.

Restarting Third-Party Applications Automatically

The Management Layer also supports the automatic restart of third-party applications following an unexpected termination. You must select **Auto-Restart** (in the Application's properties) when configuring the application. Note that the startup timeout is not used when restarting third-party applications.

Determining Application Status

The method LCA uses to determine the current status of a third-party application depends on the method SCS uses for forming the startup command line:

- If you have not configured the **start_command** option, SCS uses the values of the **Command Line** and **Command Line Arguments** to form the command line. In this case, LCA stores the PID returned by the operating system function and immediately passes the Started status to SCS.

- If you have configured the **start_command** option, SCS uses the value of this option to form the command line. In this case, LCA passes the Pending status to SCS and determines if the application has started successfully, as described in Determining Whether Applications Are Started.

Ensuring Command Line Correctness

If you want to monitor a third-party application, use the running process and its arguments as a model for the command line and command line arguments in Genesys Administrator. Follow these steps:

1. Use a system tool (for example, the UNIX tool ps) to display the running process and its arguments.

2. In the third-party Application object's properties, do the following:

    a. In the **Command Line** field, enter the exact value of the running process.

    b. In the **Command Line Arguments** field, enter the exact value of the running process arguments.

Stopping Third-Party Applications

You can stop a third-party application in the following ways:

- User command from Genesys Administrator
- Alarm Reaction

Important
The Shutdown Timeout does not apply to third-party applications.

The Management Layer can only stop a third-party application that has a status of Started; that is, LCA knows the PID for this application.

When the Management Layer receives a request to stop a particular third-party application, SCS passes it to LCA, which executes the required operating system function.

LCA processes the request as follows:

- If you have not configured the **stop_command** option and the application runs on UNIX, LCA sends the SIGINT signal to the process with the PID corresponding to the third-party application. Then, LCA sets the application status to Stopped and notifies SCS.

- If you have not configured the **stop_command** option and the application runs on Windows, LCA calls the TerminateProcess function for the process with the PID corresponding to the third-party application. Then, LCA sets the application status to Stopped and notifies SCS.

- If you have configured the **stop_command** option, LCA either executes the specified operating system command or launches the specified script or batch file. LCA sets the status of the third-party application to Pending and then determines the actual status of the application, which, when determined, it passes to SCS. (See Determining Whether Applications Are Stopped for a description of the methods LCA uses to determine application status.)

At this point, the process of stopping a third-party application is considered finished.

Example

To demonstrate how you can use the Management Layer to control a third-party application, such as License Manager, running on a Windows-based computer, do the following:

1. Install License Manager to directory **d:\flexlm**. Use the License Manager installation procedure for Windows described in the *Genesys Licensing Guide*.

2. Create two *.bat files, one (named **lmgrd_run.bat**) for starting License Manager and the other (named **lmgrd_stop.bat**) for shutting down the application. The file content should be as described in Start Script and Stop Script Content.

3. Save both files to the **d:\flexlm** directory.

4. Create an Application object of the Third Party Server type and name it **FLEXlm**. Refer to the configuration procedures in Configuring Third-Party Applications.

5. In the third-party Application object's **Startup Info** section, set the parameters as follows:

   - Specify `d:\flexlm` as the value for **Working Directory**.

   - Specify `lmgrd` as the value for **Command Line**.

   - Specify `-c d:\flexlm\license.dat` as the value for **Command Line Arguments**.

   > ### Important
   > Make sure that the combined string
   >
   > *<Working directory> + <Command Line> + <space> + <Command Line Arguments>*
   > matches the command line in the **lmgrd_run.bat** file, which is
   >
   > `d:\flexlm\lmgrd -c d:\flexlm\license.dat.`

6. Start **lmgrd_run.bat** manually. After 20 or so seconds, check the Application object's status. Its status should be Started.

7. In the FLEXlm Application object's Annex:

   a. Create a section called `start_stop`.

   b. Create two options, **start_command** and **stop_command**, in this new section.

   c. Specify full paths to the appropriate *.bat files as the option values:
      start_command = d:\flexlm\lmgrd_run.bat
      stop_command = d:\flexlm\lmgrd_stop.bat

   d. Save configuration changes.

8. Try to stop and start the FLEXlm application using appropriate commands in Genesys Administrator.

Start Script and Stop Script Content

The content of the **lmgrd_run.bat** and **lmgrd_stop.bat** files depends on whether you run License Manager as a regular application or as a Windows Service.

For a regular application, the file content should be as follows:

- **lmgrd_run.bat** `@echo "Starting FLEXlm License Manager" d:\flexlm\lmgrd -c d:\flexlm\license.dat`

- **lmgrd_stop.bat** `@echo "Stopping FLEXlm License Manager" d:\flexlm\lmutil lmdown -q -c d:\flexlm\license.dat`

For a Windows Service, the file content should be as follows:

- **lmgrd_run.bat** `net start <FLEXlm Service Name>`

- **lmgrd_stop.bat** `net stop <FLEXlm Service Name>`