Framework Management Layer User's Guide

Fault Management Functions

5/2/2025

Contents

Fault Management Functions

Faults—accidental and unplanned events causing a system to fail—present the biggest challenge to solution availability. The functions that detect, isolate, and correct various types of faults are partly incorporated into every Genesys component and partly implemented in the Management Layer of the Genesys Framework. The role of the Management Layer in application failure management is described in detail in this section.

Application Failures

A complete application failure may be a result of either an internal defect (for example, an infinite loop) or an external event (for example, a power failure). It may manifest as either a process non-response or termination. Typically, if a solution component stops working, the solution is no longer available to process customer interactions.

Since the application that fails cannot perform any functions, you must employ an external mechanism for both detection and correction of faults of this type. The Management Layer serves as such a mechanism. To detect an application failure, the Management Layer uses a simple monitoring component called Local Control Agent (LCA), which continuously maintains a connection with the application, confirming both its existence and ability to communicate. To make sure an application failure is never confused with a connection failure, the LCA that monitors a specific application always resides on the computer on which the application itself is running.

LCA is installed on a one-per-host basis and can connect to all Genesys applications located on the host. When a connection is broken, LCA generates a message to Solution Control Server, where an appropriate recovery action is chosen and executed according to the system configuration. SCS uses the Advanced Disconnect Detection Protocol (ADDP) to recognize a loss of connection with LCA. A loss of connection is interpreted as a failure of the host (that is, as failures of all Genesys components running on that host).

ADDP is, by default, enabled for the connection between SCS and LCA, with the ADDP timeout set to 9 seconds. With the default settings, SCS can detect and handle application failures in 20 seconds or less.

> ### Important
> However, if there is a particular risk of network delays, Genesys recommends setting ADDP timeouts to values equal to or greater than 10 seconds, rather than relying on default values to avoid false detection of disconnection. You can modify ADDP parameters for the connection between SCS and LCA in the Host object of the computer that runs LCA. For more information about these settings, refer to the *Framework Configuration Options Reference Manual*. For more information about ADDP, refer to the *Framework Deployment Guide*.

If you have not configured a backup application for the failed component, the correction procedure

normally consists of attempts to restart the failed process, if so configured. The same LCA component that detects application failures starts any Genesys application located on the host upon a command from SCS. If the application in question is a server, the clients automatically reconnect to this server once it is restarted and initialized.

> **Tip**
>
> Genesys recommends that you configure an automatic application restart procedure for all daemon applications.

> **Warning**
>
> Stopping an application via the Management Layer is not considered an application failure. Therefore, the Management Layer does not restart applications that it has stopped unless you have configured an appropriate alarm condition and alarm reaction for them.

If a backup application is configured and started, the Management Layer automatically switches operations over to that application, given that you have a *high-availability* (HA) license (see the following notes). If the application is a server, the clients automatically connect to the backup server.

Notes:

- HA licenses are required only for those components that support automatic switchover.
- HA licenses are not required for manual switchover, either via the application-specific menu, or by stopping the primary server. All applications can be switched over manually.

The Management Layer currently supports warm standby between redundant components within the layer. It also supports switchovers between redundant client applications, regardless of the redundancy type specified by those applications. You must have an HA license to support either type of redundancy.

> **Warning**
>
> The Management Layer does not support Cold Standby redundancy type.

The Management Layer also provides more robust switchover capabilities, and, in particular, allows detection of situations when a running application is unable to provide service and treats this situation as an application failure. The Service Unavailable application status serves this purpose.

When an application reports that its status has changed to Service Unavailable, and a backup server for this application is configured and started, the Management Layer automatically switches operations over to the backup server. Respectively, when both primary and backup applications are running with the Service Unavailable status, the backup application may report that it can now

provide the service (that is, the backup application status changes to Started). In this case, the Management Layer automatically switches operations over to the backup application. As with a switchover resulting from an application failure, you must have an HA license to perform a switchover related to service unavailability.

> **Important**
>
> While some applications support the Service Unavailable status and report it under appropriate circumstances, others do not. (For instance, when T-Server loses its connection to the CTI Link, T-Server changes its status to Service Unavailable). The Management Layer operates based on the information supplied by an application and cannot automatically detect an application's inability to provide service. Refer to application-specific documentation to determine if the Service Unavailable status is supported on the application side.

Warm Standby Redundancy Type

Genesys uses the term `Warm Standby` to describe the redundancy type with which a backup server application remains initialized and ready to take over the operations of the primary server. Inability to process interactions that may have originated during the time it took to detect the failure is reduced to a minimum. Warm Standby redundancy type also eliminates the need to bring a standby server online, thereby increasing solution availability.

The standby server recognizes its role as a backup and does not process client requests until its role is changed to primary by the Management Layer. When a connection is broken between the primary server and the LCA running on the same host, a failure of the primary process is reported. As a result, the Management Layer instructs the standby process to change its role from standby to primary, and the former standby starts processing all new requests for service.

> **Important**
>
> To switch to primary mode, the backup Configuration Server must have an active connection to the Configuration Database during the failure of the primary Configuration Server.

While normal operations are restored as soon as the standby process takes over, the fault management effort continues. It consists of repeated attempts to restart the process that failed. Once successfully restarted, the process is assigned the standby role.

If Solution Control Server detects a loss of connection with the LCA of a host, SCS performs switchover for all applications located on the host, if backup applications are configured. There are two exceptions to this:

- A Configuration Server in backup mode ignores the switchover command if it detects another Configuration Server in primary mode. In other words, if the LCA residing on a host with a Configuration Server in primary mode goes down, the SCS requests that a Configuration Server in backup mode, on another host with an available LCA, switch over to primary mode. When the request is received, this Configuration Server checks whether the Configuration Server in primary mode is down, as indicated by

a lost connection between the two Configuration Servers. The Configuration Server in backup mode switches over to primary mode only if this connection is down. If the connection still exists, no switchover occurs.

- An SCS in backup mode does not try to switch itself over if it can still detect the SCS that is in primary mode. In other words, if an SCS in backup mode loses its connection to an LCA residing on a remote host with an SCS in primary mode—either because the LCA went down or a network timeout caused the SCS to drop its connection—the SCS in backup mode checks whether it is still connected to the remote SCS in primary mode. If that connection is also lost, the SCS switches over and runs in primary mode.

Hot Standby Redundancy Type

Genesys uses the term *Hot Standby* to describe the redundancy type with which a backup server application remains initialized, clients connect to both the primary and the backup servers at startup, and the backup server data is synchronized from the primary server. Data synchronization and existing client connections to the backup guarantee higher availability of a component. Configuration Layer and Management Layer components do not support Hot Standby between pairs of redundant components. They do support switchover between client applications configured with this type.

Hot Standby redundancy type with client connection support is only implemented in T-Servers for most types of switches and is not implemented in applications of other types. For a complete description of the hot standby redundancy type, refer to the deployment guide for your specific T-Server.

Hang-up Detection

Starting in release 8.0, LCA can use hang-up detection to detect unresponsive Genesys applications supporting this functionality. Users can then configure appropriate actions, including alarms if required.

To enable hang-up detection, use the configuration options **heartbeat-period** and **heartbeat-period-thread-class-<n>** to set the time interval in which a heartbeat message must be received before the application itself, or a thread of the application, is considered to be unresponsive for each application. A third option, **hangup-restart**, can be used to set the action that LCA takes when it deems the application to be non-responsive, either automatically restarting the application or just generating a notification of the situation.

For more information about these options, refer to the *Framework Configuration Options Reference Manual*.

> ### Warning
> Use this functionality with great care, and only with those applications for which support of this functionality has been announced. Failure to use it properly could result in unexpected behavior, from ignoring the options to an unexpected restart of the application.

To determine if your application supports hang-up detection, refer to application-specific documentation. Support by Management Framework components is indicated in the following table.

Component	Application Level	Thread Level	Hangup Thread	
Class	Name			
Configuration Server	Yes	Yes	1	`auth thread`
	Yes	Yes	1	`confserv_thread`
Solution Control Server	Yes	Yes	1	`mailer thread`
DB Server [a]	Yes	No	N/A	N/A
Message Server	Yes	Yes	1	`dbthread`
SNMP Master Agent	Yes	No	N/A	N/A

a. DB Server is no longer required by Management Framework starting in release 8.5.1, but can be used in place of the new internal database access system implemented in 8.5.x if desired. Refer to the *Framework Database Connectivity Reference Guide* for more information.

Important

- The option **hangup-restart** does not apply to Solution Control Server.
- The option **heartbeat-period-thread-class-<n>** does not apply to SNMP Master Agent.
- Solution Control Server must be running if the **hangup-restart** option is used.

Split Brain or No Brain State Detection

Split brain occurs when the state of both the primary and backup applications in an HA pair controlled by Solution Control Server (SCS) changes to primary, and no brain state can happen when one or both applications become unavailable or change state to backup. SCS generates the following log events to indicate the state of applications:

- 20621 - Indicates that an application is in "no brain" state.
- 20622 - Indicates that "split brain" condition has occurred for an application.
- 20623 - Indicates that an application is back to the normal state.

Important

These alarms might not work as expected if SCS is in the split brain state. Also, during switchover, the alarm is generated when the applications are either in primary/primary or backup/backup mode and is cleared automatically when the applications complete switchover.

You can create alarms using these log events to detect split brain or no brain state of applications. For more information on creating alarms using log events, see Using Log Events for Alarm Detection.

Detecting split-brain state of Solution Control Servers

Perform the following steps to detect if Solution Control Server (SCS) has entered the split-brain state:

1. Check if both primary and backup SCS are running.
2. Check if the following line is available in the logs of backup SCS:

   ```
   "SelfServer: Internal Change RUNMODE from 'BACKUP' to 'PRIMARY'".
   ```

   This log message indicates that the state of SCS has most likely entered the split brain state.

> **Important**
>
> Starting from Solution Control Server 8.5.100.38, the Log Event "10329" is generated when primary SCS disconnects from the backup SCS, indicating a possible SCS split brain state. The Log Event "10330" is generated when SCS returns to the normal state.

Perform the following checks to recover SCS from the split-brain state:

- Check and resolve if there are network connectivity issues between primary and backup SCS machines.
- Check whether DNS resolves quickly from the host where the primary SCS runs. If there is a delay in DNS resolution, troubleshoot the issue.
- Check whether there is any addp timeout triggering SCS to connect to host or resolve an FQDN which takes more time. In these cases, check the reason for addp timeout and increase the timeout value accordingly.

Remote Site Failures with Distributed Solution Control Servers

Starting in release 8.0, any Solution Control Server in the distributed environment can also detect the failure of a remote site controlled by another Solution Control Server in the environment and generate an appropriate log message.

Solution Control Server considers a remote site to have failed if it stops receiving polling messages from the Solution Control Server (or primary and backup Solution Control Servers, if configured) controlling the remote site within a specified time period. The time period is specified by the configuration option **alive_timeout**, configured on each Solution Control Server. Refer to the *Framework Configuration Options Reference Manual* for a detailed description of this option. In this

case, the primary Solution Control Server generates log event 43-20600.

If the remote site recovers and the Solution Control Server (or primary and backup Solution Control Servers, if configured) controlling that site starts to send polling messages, the primary Solution Control Server generates log event 43-20601 to indicate that the remote site is back in service.

For additional information, refer to the following documents:

- *Framework Deployment Guide* for more information about distributed Solution Control Servers, and for detailed instructions for configuring them
- *Framework Configuration Options Reference Manual* for a detailed description of the option **alive_timeout**
- *Framework Combined Log Events Help* for a full description of Solution Control Server log events 43-20600 and 43-20601.

Site Failure (Disaster Recovery)

Starting in release 8.5, Genesys software provides some Disaster Recovery/Business Continuity functionality, which enables you to continue operations and prevent data loss if the main site fails because of a natural, man-made, or unintended situation.

When the main site fails, the active Log Database is automatically replicated into the dormant Log Database installed at the remote site. The dormant Log Message Server is started and connects to that now-active Log Database. Another dormant Message Server is started and provides communication between the SCS at the failed site and the SCS at the remote site.

Refer to the *Framework Deployment Guide* for more information about this functionality.