



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Framework Management Layer User's Guide

How to Handle Stuck Calls

5/5/2025

# How to Handle Stuck Calls

## Contents

- **1 How to Handle Stuck Calls**
  - 1.1 Which Method To Use?
  - 1.2 Prerequisites
  - 1.3 Using T-Server
  - 1.4 Using the SNMP Interface
  - 1.5 Using the gstuckcalls Utility

This topic describes the procedures related to detecting and ways of dealing with calls that appear to be stuck.

A stuck call occurs when information about the release of a call in the communication system fails to reach one or more of the components of a CTI solution. One possible cause is the temporary loss of communication between CTI applications and devices, such as switching and interactive voice response systems, in the communication infrastructure.

Having missed the call release information, CTI applications continue to treat such calls as active, which results in less efficient operation and inaccurate reporting.

Because T-Servers are directly involved in communications with the switching systems, they play a critical role in detecting and handling stuck calls.

## Which Method To Use?

Stuck calls can be handled by any of three methods:

1. Using **T-Server switch-specific** functionality
2. Using the **SNMP interface** in the Management Layer
3. Using the **gstuckcalls utility** in the Management Layer

### T-Server Switch-Specific Functionality

This method offers stuck calls detection and cleanup built-in to T-Server.

This is the basic form of using the stuck call feature in T-Server that provides for minimal customization and provisioning from the user's perspective.

Characteristics:

- A simple, timeout-based detection mechanism is used internally in T-Server.
- This method does not utilize Management Layer capabilities—no automatic reactions to be executed upon detection of stuck calls.
- You must set up and manage each T-Server manually and individually, using Genesys Administrator.
- Unwieldy for managing multiple T-Servers.

See **Using T-Server**.

### SNMP Interface in the Management Layer

This method provides an SNMP interface, good for SNMP-based installations and in an SNMP-oriented application.

### Characteristics:

- Relies on external SNMP-aware applications (such as SNMP Perl scripts) to monitor and detect stuck calls in T-Server.
- Stuck call detection logic is highly customizable; information such as filters and timestamp properties lies in the new SNMP tables.
- The script provides for a central point of management, and can be tailored to manage a single or multiple T-Server applications.
- Does not utilize the Management Layer capability to monitor and react to alarm events; the script must do it.
- Requires SNMP licensing.

See [Using the SNMP Interface](#).

### gstuckcalls Utility

This method has the advantages of the second method but does not require SNMP.

### Characteristics:

- The stuck call detection logic is highly customizable.
- This method is integrated with the Management Layer. A stuck calls event can be configured and reacted upon when corresponding log messages are received by SCS.
- This method does not require an SNMP license.
- The scripts in this utility generate an XML file with a summary of all calls retrieved from T-Server, which makes it useful as a quick-look diagnostic tool.

The **gstuckcalls** utility uses the **logmsg** utility, which allows sending specified log messages to trigger and clear Management Layer alarms based on conditions external to the Management Layer.

See [Using the gstuckcalls Utility](#).

### Prerequisites

The Stuck Calls functionality requires that Perl be installed on the SCS host computer. The following table lists the names and minimum versions of Perl extension modules required. Users may need to install some or all of them, depending on their current Perl installation. **[+] Show table**

Perl Module	Recommended Version
HTML::Parser	3.25 and higher
SOAP::Lite	0.60 and higher
XML::DOM	1.43 and higher

Perl Module	Recommended Version
XML::Parser	2.34 and higher
XML::SAX	0.12 and higher
XML::NamespaceSupport	1.08 and higher
XML::RegExp	0.03 and higher
HTTP::Cookies	1.39 and higher

These modules are available from the Comprehensive Perl Archive Network (CPAN) website (<http://www.CPAN.org>).

The Framework stuck calls functionality—including the Perl scripts **GStuckCallsDetect.pl** and **GStuckCallsClear.pl**, and the above modules—were tested using Perl version 5.6.1.

## Using T-Server

To support stuck calls handling in T-Server, a set of configuration options has been introduced. These options control stuck call detection, notification, and automatic cleanup. For more information, refer to the “T-Server Common Configuration Options” chapter in the latest version of the Deployment Guide for your **T-Server**.

To support stuck calls handling in the Management Layer, a set of log messages have been added to the T-Server Common Part. Download the *Framework Combined Log Events Help* for more information.

To support stuck calls handling in client applications of T-Server, a new property has been added to the T-Server events that define the end of the call (EventReleased and EventAbandoned). See the *Genesys Events and Models Reference Manual* for more information.

Based on a specified timeout, T-Server waits for call information being updated. After the timeout is expired, T-Server considers a call as a stuck call and reports a standard log message.

Processing of timeouts and notifications is implemented in the T-Server Common Part, but the actual call cleanup involves interaction with the switch-dependent part for each T-Server.

## Configuration Options Summary

Three new options must be configured in the section call-cleanup.

- **notify-idle-tout:** This option specifies the time interval that T-Server waits for a call being updated from its last update. After this time elapses, if no new events about the call are received, T-Server reports this call as a stuck call.
- **cleanup-idle-tout:** This option specifies the time interval that T-Server waits for a call being updated from its last update. After this time elapses, if no new events about the call are received, T-Server clears this call as a stuck call either by querying the switch (if a CTI link provides such capabilities), or by deleting call information from memory unconditionally. The option description in the Deployment Guide for each **T-Server** reflects the actual implementation for that particular T-Server.
- **periodic-check-tout:** This option specifies the time interval for periodic checks for stuck calls (affects both notification and cleanup functionality) by checking the T-Server’s own call information with call

information available in the switch. For performance reasons, T-Server does not verify whether the **notify-idle-tout** or **cleanup-idle-tout** option has expired before performing this checking.

### T-Server Common Log Events

Three T-Server common log events support stuck calls management: 01-20020, 01-20021, and 01-20022. Refer to the latest *Framework Combined Log Events Help* for detailed specifications of these log events.

### EventReleased on special DN

The value of the **TReliability** parameter indicates that the update was forced by external request:

- TReliabilityExternal = 3
- TReliabilityExternal - cleared by an external SNMP request

### Using the SNMP Interface

The following tables in the T-Server-Specific SNMP Objects support management of stuck calls using the SNMP Interface in the Management Layer. These tables allow you to retrieve only those call instances that were defined by the filters in the tsCallFilterTable table thus reducing network traffic and increasing application performance.

- **tsCallFilterTable** provides the interface for setting call filter criteria for the tsCallInfoTable table. It also provides the interface for clearing calls by the call's Connection ID.
- **tsCallInfoTable** stores the latest snapshot of active calls from a given T-Server, and contains information about active calls filtered by conditions set in the tsCallFilterTable.

### Using the gstuckcalls Utility

The **gstuckcalls** utility contains two stuck calls management scripts, **GStuckCallsDetect.pl** and **GStuckCallsClear.pl**, which support the detection and automatic clearing of stuck calls. Both scripts use the **gstuckcallsscript.cfg** configuration file, also part of the utility.

#### Important

When using the **gstuckcalls** utility, the password is saved as plain text in the configuration file and this file is usually not encrypted. Therefore, Genesys recommends to carefully guard the configuration file containing the passwords.

### How to Install the Utility

If you installed Solution Control Server 8.5.100.41 and earlier, the **gstuckcalls** utility containing the

---

stuck calls scripts and configuration file is already installed, and is located in the same folder in which Solution Control Server was installed.

Starting with SCS 8.5.100.42 release, the **gstuckcalls** utility is not installed by default with Solution Control Server. You can install it separately by following the instructions in the section [Installing the Solution Control Server Utilities](#) of the Framework Deployment Guide. After the utilities are installed, the **gstuckcalls** utility is stored in the location that you specified during the installation.

Starting in 8.1.2, you can install the Solution Control Server utilities without installing Solution Control Server itself.

## How to Run the Utility

To start the stuck calls utility:

- In Windows, enter **gstuckcalls.exe** on the command line.
- On UNIX, enter **gstuckcalls** on the command line.

The utility will automatically run the scripts.

### GStuckCallsDetect.pl Script

The **GStuckCallsDetect.pl** script performs these functions:

1. Retrieves all the information about all T-Servers from the configuration.
2. Queries each T-Server for stuck calls according to the specified filter using the **gstuckcalls** utility.
3. If stuck calls are found, sends log message 0095000 on behalf of the T-Server.
4. If stuck calls are not found, sends log message 095010 on behalf of the T-Server.

If you require alarming for stuck calls, schedule this script for periodic execution by using tools provided by your operating system, such as Scheduled Tasks for Windows and Cron for UNIX.

### GStuckCallsClear.pl Script

The **GStuckCallsClear.pl** script clears stuck calls in the specified T-Server. If you need to clear stuck calls automatically, use this script as an alarm reaction for the active alarm Stuck Calls Detected. This script performs the following:

1. Connects to the specified T-Server.
2. Uses the **gstuckcalls** utility to clear all stuck calls according to the specified filter.

### gstuckcallsscript.cfg Configuration File

The **GStuckCallsDetect.pl** and **GStuckCallsClear.pl** scripts use this configuration file. It has the following format:

```
[cfgserver]
```

```
host=<host>
port=<port>
username = <username>
password = <password>
[msgserver]
host=<host>
port=<port>
[filter]
createdbefore=<seconds>
createdafter=<seconds>
updatedbefore=<seconds>
updatedafter=<seconds>
```

### Stuck Calls Alarm Log Messages

The following alarm log messages have been added to support detection and automatic clearance of stuck calls:

- **09500:** Stuck calls detected
- **09501:** Stuck calls not detected

### Configuring the Alarm Condition

To enable automatic stuck calls detection, configure the corresponding Alarm Condition with the following settings:

For Detect Event:

- Log Event ID set to 9500
- Selection Mode set to Select By Application Type
- Type set to T-Server

For Cancel Event:

- Log Event ID set to 9501

See [Using Log Events for Alarm Detection](#) for more information.

The active alarm Stuck Calls Detected is communicated when log message 9500 is received. This happens when the **GStuckCallsDetect.pl** script detects stuck calls in a T-Server. Scheduling the **GStuckCallsDetect.pl** script for periodic execution (for instance, once per day) ensures automatic stuck calls detection and alarming.

To clear stuck calls automatically, follow these steps:

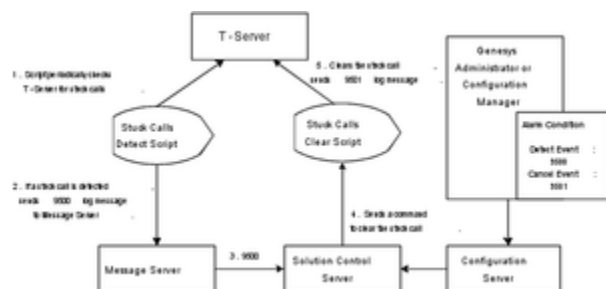
1. Configure the alarm reaction type Execute OS Command for the Alarm Condition Stuck Calls Detected.
2. Configure this alarm reaction to execute the **GStuckCallsClear.pl** script.

The script clears stuck calls at the corresponding T-Server and sends log message 9501 to the Management Layer, which then clears the active alarm Stuck Calls Detected.



### Stuck Calls Scripts Flow Chart

The following figure provides a flow chart to help you better understand the scripts.



Stuck Calls Script Flowchart