Framework 8.1

SIP Server

Deployment Guide

About Genesys

Every year, Genesys® delivers more than 70 billion remarkable customer experiences for organizations in over 100 countries. Through the power of the cloud and AI, our technology connects every customer moment across marketing, sales and service on any channel, while also improving employee experiences. Genesys pioneered Experience as a Service. so organizations of any size can provide true personalization at scale, interact with empathy, and foster customer trust and loyalty. This is enabled by Genesys Cloud™, an all-in-one solution and the world's leading public cloud contact center platform, designed for rapid innovation, scalability and flexibility. Go to www.genesys.com for more information.

Each product has its own documentation for online viewing at the Genesys Documentation website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Genesys and the Genesys logo are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other company names and logos may be trademarks or registered trademarks of their respective holders.

The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Customer Care from Genesys

If you have purchased support directly from Genesys, please contact Genesys Customer Care. Before contacting Customer Care, please refer to the ***Genesys Care Support Guide for On-Premises*** for complete contact information and procedures.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the ***Genesys Licensing Guide.***

Released by

Genesys Telecommunications Laboratories, Inc. www.genesys.com

Document Version: 81fr_dep-sip_10-2021_v8.1.101.70

GENESYS

Table of Contents

GENESYS

List of Procedures

GENESYS

Preface

Welcome to the *Framework 8.1 SIP Server Deployment Guide*. This document introduces you to the concepts, terminology, and procedures relevant to T-Servers® in general and provides detailed reference information about SIP Server. The information includes, but is not limited to, configuration options, limitations, and switch-specific functionality. You must configure the configuration objects and options described in this document in the Framework Configuration Layer.

Use this document only after you have read through the *Framework Deployment Guide,* and the Release Note for SIP Server.

This document is valid only for the 8.1 release of this product.

Note: For versions of this document created for other releases of this product, visit the Genesys Documentation website, or request the Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesys.com.

This preface contains the following sections:

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on page 765.

About SIP Server

SIP Server is the Genesys software component that provides an interface between your telephony hardware and the rest of the Genesys software components in your enterprise. It translates and keeps track of events and requests that come from, and are sent to the telephony device. SIP Server is a TCP/IP-based server that can also act as a messaging interface between SIP Server clients. It is the critical point in allowing your Genesys solution to facilitate and track the contacts that flow through your enterprise.

Intended Audience

This guide is intended primarily for system administrators, both those who are new to SIP Server and those who are familiar with it.

- If you are new to SIP Server, read the *Framework 8.1 SIP Server Deployment Guide* and the Release Note mentioned earlier, and then read all of the sections of this document that apply to your software and its accompanying components. Refer back to the *Framework Deployment Guide* as needed.
- If you are an experienced SIP Server user—someone with computer expertise, who is used to installing, configuring, testing, or maintaining Genesys software—you may find it more time efficient to go to the Index to see what is new or different in SIP Server release 8.1. If you take that approach, please also read Release Notes and refer to other related resources, such as the *Genesys Events and Models Reference Manual*.

In general, this document assumes that you have a basic understanding of, and familiarity with:

- Network design and operation.
- Your own network configurations.
- Your telephony hardware and software.
- Genesys Framework architecture and functions.
- Genesys Administrator Extension (GAX) interface and object-managing operations.

Based on your specific contact center environment and your responsibilities in it, you may need to be familiar with a much wider range of issues as you deploy SIP Server.

Reading Prerequisites

You must read the *Framework Deployment Guide* before using this *SIP Server Deployment Guide*. That book contains information about the Genesys software you must deploy before deploying SIP Server.

Making Comments on This Document

If you especially like or dislike anything about this document, feel free to e-mail your comments to Techpubs.webadmin@genesys.com.

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document. Please limit your comments to the scope of this document only and to the way in which the information is presented. Contact your Genesys Account

Representative or Genesys Customer Care if you have suggestions about the product itself.

When you send us comments, you grant Genesys a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

Contacting Genesys Customer Care

If you have purchased support directly from Genesys, please contact Genesys Customer Care.

Before contacting Customer Care, please refer to the *Genesys Care Support Guide for On-Premises* for complete contact information and procedures.

Document Change History

This section lists topics that are new or that have changed significantly since the first release of this document.

New in Document Version 8.1.101.70

- Added "Strict SIP Endpoint Registration" on page 366.
- Added "Treating Incoming Calls As Inbound Calls" on page 369.
- Added a known limitation to Mapping SIP Headers and SDP Messages, "From T-Library Messages to SIP Messages" on page 272.
- Added the following configuration options:
  - `clearcall-sip-reject-code` (Application level)
  - `enforce-1pcc-inbound` (Application level)
  - `inbound-trunk-hint-sip-field` (Application level)
  - `inbound-trunk-hint` (DN level)
  - `internal-call-domains` (Application level)
  - `no-login-on-presence` (Application level)
  - `reuse-tls-conn` (Application level)
  - `sip-registrar-allowlist` (Application level)
  - `sip-registrar-allowlist-origin` (Application level)
  - `sip-registrar-reject-code` (Application level)
  - `sip-transfer-complete-timeout` (Application level)
  - `update-ctrl-party` (Application level)
  - `userdata-map-filter-mode` (Application level)
- Updated configuration options:
  - `sip-ring-tone-mode` (Application level)

- sip-ring-tone-mode (DN level)
- sip-transfer-complete-message (DN level)
- userdata-map-filter (DN level)

New in Document Version 8.1.101.60

The following topics have been added or changed since the previous release of this document:

- Configuration Manager and Genesys Administrator are replaced with Genesys Administration Extension (GAX).
- Added "Passing Extended Recording Metadata to GVP" on page 403.
- Added "Enhanced Handling of XS Requests" on page 214.
- Added "Asynchronous DNS Resolution" on page 219.
- Added "Antivirus Guidelines" on page 55.
- Updated links to the Framework Deployment Guide.
- Updated "Remote Agents Support" on page 342.
- Updated "Endpoint Sends PUBLISH Requests to SIP Server" on page 327.
- Added the following configuration options:
  - backup-init-check (Application level)
  - backup-init-check-timeout (Application level)
  - backup-sip-port-check (Application level)
  - enable-async-fqdn-resolve (DN level)
  - enable-enhanced-dialplan-handling (Application level)
  - enhanced-pending-acw (Application level)
  - ha-max-calls-sync-at-once (Application level)
  - make-call-cpd-merged-userdata (DN level)
  - record-metadata-prefix (Application level)
  - reset-acw-persistent-reasons (Application level)
  - sip-enable-replaces (DN level)
  - sip-continue-treatment-on-call-reject (Application level)
  - sip-enable-strict-auth (Application and DN levels)
  - sip-error-overflow (DN level)
  - sip-reinvite-action (DN level)
  - sip-progress-response-code (DN level)
  - sip-retry-after (Application level)
  - switchover-on-msml-oos (Application level)
  - switchover-on-trunks-oos (Application level)
  - switchover-on-xs-oos (Application level)
  - t-library-stats-enabled (Application level)
  - time-before-switchover-on-xs-oos (Application level)

  - trunk-stats-enabled (Application level)
  - x-sip-mask-sensitive-data (Application level, log section)
  - x-sip-unmask-headers (Application level, log section)
  - x-sip-unmask-headers-default (Application level, log section)
  - xs-heartbeat-interval (Application and DN levels)
  - xs-heartbeat-timeout (DN level)
  - xs-missed-heartbeat-threshold (DN level)
  - xs-pool-size (Application and DN levels)
  - xs-post-timeout (DN level)
  - xs-request-timeout (DN level)
- Updated the following configuration options:
  - dual-dialog-enabled (DN level)
  - enable-retransmit-on-oos-transport (Application and DN levels)
  - password (DN level, AuthClient section)
  - sip-tls-sec-protocol (Application level)
  - username (DN level, AuthClient section)

New in Document Version 8.1.101.55

The following topics have been added or changed since the previous release of this document:

- Added the BusinessCallType attribute extension.
- Updated "Preview Interactions" on page 335.
- Updated "Video Support" on page 382.
- Updated the TEXT parameter in Table 70.
- Updated "Agent Login and State Update on SIP Phones" on page 329.
- Updated "Nailed-Up Connections for Agents" on page 287.
- Updated "DTMF Tones Generation on Media Server" on page 222.
- Updated "Mapping SIP Headers and SDP Messages" on page 261.
- Removed the Outbound dialing rules section.
- Removed the Stream Manager component.
- Removed "Asterisk Voice Mail Integration."
- Added the following configuration options:
  - agent-allow-empty-password (Application level)
  - enable-retransmit-on-oos-transport (Application level)
  - enable-retransmit-on-oos-transport (DN level)
  - override-domain-ruri (DN level)
  - sip-add-via (DN level)
  - sip-call-id-suffix (Application level)
  - sip-contact-user (DN level)

  - sip-disable-unreliable-sdp (DN level)
  - sip-wait-ack-timeout (Application level)
- Updated the following configuration options:
  - acw-persistent-reasons (Application level)
  - agent-strict-id (Application level)
  - contact (DN level)
  - enable-async-dns (Application level)
  - info-pass-through (Application level)
  - info-pass-through (DN level)
  - logout-on-out-of-service (Application level)
  - no-answer-overflow (DN level)
  - public-contact (DN level)
  - sip-address-srv (Application level)
  - sip-dtmf-send-rtp (Application level)
  - sip-<SIP_error_code> (Application level)
  - sip-error-conversion (DN level)
  - wrap-up-time (Application level)
- Removed the out-rule-<n> configuration option.

New in Document Version 8.1.101.50

The following topics have been added or changed since the previous release of this document:

- "Secure SIP Signaling" on page 348.
- "Configuring Remote Agents with Non-provisioned Phone Numbers" on page 345.
- "Mapping SIP Headers and SDP Messages" on page 261.
- New configuration options:
  - enable-outbound-ext-dial-plan
  - ipo-tout
  - recording-failure-alarm-timeout
  - report-error-on-routing-end
  - sip-elin-timeout
  - subscription-max-body-size
  - unknown-gateway-reject-code
- Updated configuration options:
  - extensions-<n>
  - userdata-<n>
  - override-to-on-divert
  - rp-use-dial-plan
  - sip-ring-tone-mode

- subscription-event-allowed

New in Document Version 8.1.101.45

The following topics have been added or changed since the previous release of this document:

- observing-routing-point option has been updated.

New in Document Version 8.1.101.44

The following topics have been added or changed since the previous release of this document:

- Added "SRV Address Support in Contact and Record-Route Headers" on page 365.
- Added "Masking Sensitive Data in SIP Messages" on page 276.
- Added the following configuration options:
  - sip-disable-via-srv
  - sip-enable-x-genesys-route
  - sip-recovery-allow-userdata
  - sip-response-msml-oos
  - x-sip-mask-sensitive-data
- Updated the following configuration options:
  - extensions-<n>
  - session-refresh-enforced

New in Document Version 8.1.101.40

The following topics have been added or changed since the previous release of this document:

- Added "Recording an Agent Greeting" on page 324.
- Added "Controlling Early Media with a Routing Strategy" on page 233.
- Added "HTTP Live Streaming" on page 244.
- Added "SIP Feature Server Log Messages" on page 213.
- Added "Setting SIP INVITE Timeout for Individual DNs" on page 108.
- Revised "Customizing Music on Hold and in Queue" on page 179.
- Added "CPU Usage Overload Control" on page 316.
- Added the following configuration options:
  - enable-oosp-alarm (DN level)
  - find-outbound-msml-by-location (Application level)
  - hide-msml-location (Application level)
  - log-reduce-cpu-threshold (Application level)
  - msml-enable-record-extensions (Application level)

  - msml-oos-recover-enabled (Application level)
  - music-on-hold (DN level)
  - record-agent-greeting (Application level)
  - sip-enable-ivr-metadata (Application level)
  - sip-enable-ivr-metadata (DN level)
  - sip-enhance-diversion (Application level)
  - sip-trying-timeout (DN level)
- Updated the following configuration options:
  - sip-error-conversion (DN level)
  - sip-error-conversion (Application level)
- Removed the following configuration option:
  - sip-save-rejected-sdp

New in Document Version 8.1.101.35

The following topics have been added or changed since the previous release of this document:

- Added "Instant Messaging For Multi-Site Calls" on page 255.
- Added "Defining After Routing Timeout Action" on page 303.
- Updated "Caller Information Delivery Content for AT&T Trunks" on page 117 to include support for GVP.
- Added "Modifying the From Header in SIP INVITE" on page 279.
- Added "Muting/Unmuting a Party in a Conference" on page 167.
- Added "HTTP Monitoring Interface" on page 245.
- Updated "Dial Plan" on page 195 to include support for the SIP Feature Server dial plan.
- Added the following configuration options:
  - acw-persistent-reasons (Application level)
  - after-routing-timeout-action (Application level)
  - cid-enable-on-vtp (Application level)
  - cpn-digits-to-both-legs (DN level)
  - cpn-dnis (DN level)
  - cpn-self (DN level)
  - enable-iscc-dial-plan (Application level)
  - enable-legacy-reporting (Application level)
  - enforce-rfc3455 (DN level)
  - greeting-stops-no-answer-timeout (Application level)
  - http-port (Application level)
  - msml-mute-type (Application level)
  - msml-record-metadata-support (Application level)

  - rp-use-dial-plan (Application level)
  - sip-add-local-contact-user (Application level)
  - sip-enable-two-party-mute (Application level)
  - sip-pass-body (DN level)
  - sip-route-active-transport (DN level)
  - sip-tls-sec-protocol (Application level)
  - summary-stat-timeout (Application level)
  - transaction-state (Application level)
- Updated the following configuration options:
  - audio-codecs (Application level)
  - audio-codecs (DN level)
  - convert-otherdn (Application level)
  - enable-strict-location-match (Application level)
  - prefix (DN level)
  - replace-prefix (DN level)

New in Document Version 8.1.101.30

The following topics have been added or changed since the previous release of this document:

- Added "Find Me Follow Me" on page 243.
- Added "Customer-on-Hold Privacy" on page 143.
- Added "Caller Information Delivery Content for AT&T Trunks" on page 117.
- Added "DTMF Clamping in a Conference" on page 220.
- Added "Private Conversations During Conference" on page 166.
- Added "Remote Agents Support" on page 342.
- Added "Sending Outgoing INVITEs with Multipart Body" on page 350.
- Added "Providing Origination DN Name and Location in EventRinging" on page 338.
- Added reference to "HTTP Monitoring Interface" on page 245.
- Updated "Alternate Routing for Unresponsive URS/ORS" on page 109.
- Updated "Providing Call Participant Info" on page 336.
- Updated "ISCC/Call Overflow Feature" on page 677 to include SIP Server support of ANI matching.
- Updated the following configuration options:
  - default-route-point
  - dr-forward
  - operational-stat-timeout
  - sip-enable-call-info

- Added the following configuration options:
  - alternate-route-profile (Application level)
  - clamp-dtmf-allowed (Application level)
  - clamp-dtmf-enabled (DN level)
  - control-remote-vip-scripts (Application level)
  - control-vip-scripts (Application level)
  - default-route-point-order (Application level)
  - dr-oosp-transfer-enabled (DN level)
  - dr-peer-location (Application level)
  - fmfm-confirmation-digit (Application level)
  - fmfm-confirmation-timeout (Application level)
  - fmfm-prompt-file (Application level)
  - fmfm-trunk-group (Application level)
  - graceful-shutdown-sip-timeout (Application level)
  - monitor-party-on-hold (Application level)
  - music-listen-disconnect (Application level)
  - network-monitoring-timeout (Application level)
  - resolve-internal-rp-by-host (Application level)
  - sip-accept-body (DN level)
  - sip-enable-call-info-extended (Application level)
  - sip-iptakeover-monitoring (Application level)
  - sip-nic-address (Application level)
  - sip-nic-monitoring (Application level)
  - sip-release-call-on-disable-dn (Application level)
  - sip-vip-script-down (Application level)
  - sip-vip-script-up (Application level)
  - tlib-nic-monitoring (Application level)
  - vip-state-change-timeout (Application level)

New in Document Version 8.1.101.25

The following topics have been added or changed since the previous release of this document:

- Added “Shared Call Appearance” on page 357.
- Added “Disabling Media Before Greeting” on page 324.
- Added “Geo-Location Support by GVP” on page 392.
- Added “Deleting Party From Conference in Multi-site Deployments” on page 165.
- Updated “VXML Support for Agent Greetings” on page 321.
- Updated “Geo-Location for MSML-Based Services: Strict Matching” on page 393.

- Updated "Quality of Service" on page 341.
- Updated "Genesys Media Server Integration" on page 94.
- Updated "Nailed-Up Connections for Agents" on page 287.
- Added the following configuration options:
  - `agent-reject-route-point` (DN level)
  - `connect-nailedup-on-login` (Application level)
  - `connect-nailedup-on-login` (DN level)
  - `disable-media-before-greeting` (Application level)
  - `disable-media-before-greeting` (DN level)
  - `disconnect-nailedup-timeout` (Application level)
  - `disconnect-nailedup-timeout` (DN level)
  - `msml-location-alarm-timeout` (Application level)
  - `overflow-location-map` (Application level)
  - `predictive-timerb-enabled` (DN level)
  - `shared-line` (DN-level)
  - `shared-line-capacity` (DN level)
  - `shared-line-number` (DN level)
  - `sip-remote-del-from-conf` (Application level)

New in Document Version 8.1.101.22

The following topics have been added or changed since the previous release of this document:

- Added "Switching Between Supervision Modes" on page 142.
- Added "Keep Alive for TCP Connections" on page 259.
- Added "Geo-Location for MSML-Based Services: Strict Matching" on page 393.
- Added "ISCC Path Optimization" on page 378 and page 695.
- Added "Logging To Remote Location" on page 282.
- Updated the following configuration options:
  - `overload-ctrl-call-tapplytreatment-requests-rate` (Application level)
  - `overload-ctrl-call-tupdateuserdata-requests-rate` (Application level)
  - `overload-ctrl-call-trequests-rate` (Application level)
  - `overload-ctrl-trequests-rate` (Application level)
  - `sip-enable-call-info` (Application level)
- Added the following configuration options:
  - `auto-answer-after` (DN level)
  - `dr-forward` (DN level)
  - `enable-strict-location-match` (Application level)

  - oos-error-check (DN level)
  - session-refresh-enforced (Application level)
  - session-refresh-enforced (DN level)
  - sip-3pcc-from-pass-through (Application level)
  - sip-enable-tcp-keep-alive (Application level)
  - sip-pass-xfer-params-enabled (DN level)
  - sip-resubscribe-on-nonotify (Application level)
  - userdata-map-invite-after-refer (Application level)
- Removed the following configuration option:
  - sip-call-retain-timeout

New in Document Version 8.1.101.18

The following topics have been added or changed since the previous release of this document:

- Updated Hunt Groups to support sequential ringing. See “Hunt Groups” on page 245.
- Added “VXML Support for Agent Greetings” on page 321.
- Added “Routed Calls as Originating or Terminating” on page 248.
- Updated “Music and Announcements” on page 283.
- Updated “Nailed-Up Connections for Agents” on page 287.
- Updated “Call Recording—NETANN-Based” on page 121.
- Updated “Call Recording—MSML-Based” on page 125.
- Updated “Overload Control” on page 313.
- Updated “Providing Call Participant Info” on page 336.
- Updated “Smart OtherDN Handling” on page 363.
- Updated “SIP Traffic Monitoring” on page 355.
- Updated “User to User Information (UUI)” on page 379.
- Updated “Agent Login and State Update on SIP Phones” on page 329.
- Updated “Multi-Threaded Versus Single-Threaded Mode” on page 50.
- Added information about a number of client connections that SIP Server supports for Windows and Linux operating systems. See “Client Connections” on page 657.
- Added the following keys to AttributeExtensions:
  - agent-greeting-type
  - LCTParty<n>_location
- Added the following configuration options:
  - ims-use-term-legs-for-routing (Application level)
  - peer-proxy-contact (DN level)

- Updated the following configuration options:
  - display-name (DN level)
  - dr-forward (Application level)
  - dual-dialog-enabled (DN level)
  - feature-code-park (Application level)
  - feature-code-pickup (Application level)
  - feature-code-retrieve (Application level)
  - hg-noanswer-timeout (DN level)
  - hg-queue-limit (DN level)
  - hg-queue-timeout (DN level)
  - hg-type (DN level)
  - overload-ctrl-threshold (Application level)
  - sip-link-type (Application level)
  - sip-alert-info (DN level)
  - sip-alert-info-external (DN level)
  - sip-alert-info-consult (DN level)
  - use-display-name (DN level)

New in Document Version 8.1.101.14

The following topics have been added or changed since the previous release of this document:

- Updated "Trunk Capacity Control" on page 372.
- Added "Video Blocking" on page 381.
- Added "User to User Information (UUI)" on page 379.
- Updated "Dial Plan" on page 195.
- Added "Removal Overdialed Digits From DNIS" on page 198.
- Added "DN Recording Override" on page 132.
- Added the following configuration options:
  - capacity-sip-error-code (Application level)
  - capacity-tlib-error-code (Application level)
  - capacity-limit-inbound (DN level)
  - init-dnis-by-ruri (Application level)
  - mwi-subscribe-vmb (Application level)
  - resolve-external-contact (Application level)
  - sip-filter-media (Application level)
  - sip-filter-media (DN level)
  - sip-rel-200-retransmit (Application level)
- Updated the following configuration options:
  - agent-emu-login-on-call (Application level)
  - logout-on-disconnect (Application level)

- subscription-timeout (Application level)
- userdata-map-format (DN level)

New in Document Version 8.1.101.10

The following topics have been added or changed since the previous release of this document:

- Updated "Network Asserted Identity" on page 292.
- Added "Providing AoC Notifications for Established Calls" on page 102.
- Added support for the original-dialplan-digits extension key to provide the original destination number before the dial plan is applied. See "Dial Plan" on page 195.
- Added "Recording Calls Without Music-on-Hold Treatment" on page 128.
- Added "Call Recording Alarms" on page 130.
- Added "SDP Message Mapping" on page 275.
- Added support for CPD performed by the Genesys Media Server. See "Outbound IP Solution Integration" on page 306.
- Added "Referred-By Header Support" on page 172.
- Added "Filter Greetings By Call Type" on page 319.
- Removed Wizard deployment procedures.

New in Document Version 8.1.101.05

The following topics have been added or changed since the previous release of this document:

- Updated the Active-Active Resource Managers support. See "Active-Active Resource Managers" on page 396.
- Added "SIP Proxy Support" on page 354.
- Added the T-Request overload control section and related configuration options for this functionality. See "Overload Control" on page 313 for details.
- Updated "Guidelines for Deploying SIP Server on Various Operating Systems" on page 54.
- Added "Selecting SIP Call Flows from the Routing Strategy" on page 163.
- Updated the following configuration options:
  - after-call-divert-destination (DN level)
  - call-monitor-acw (Application level)
  - override-domain-oosp (DN level)
  - reuse-sdp-on-reinvite (DN level)
  - replace-prefix (DN level)
  - sip-link-type (Application level)

- sip-respect-privacy (Application level)
- userdata-map-format (DN level)

New in Document Version 8.1.101.00

This document has been updated to support SIP Server release 8.1.1.

- See "New in Release 8.1.1" on page 34 for information about new supported features.

New in Document Version 8.1.003.08

The following topics have been added or changed since the previous release of this document:

- Changed several Application and DN-level options that were incorrectly described as changes to the options taking effect immediately. Changes at the DN-level typically do not take place until the next call. The following options were changed:
  - agent-greeting (DN level)
  - audio-codecs (DN level)
  - cos (Application level)
  - cpn (DN level)
  - customer-greeting (DN level)
  - default-dn (DN level)
  - display-name (DN level)
  - dual-dialog-enabled (DN level)
  - geo-location (DN level)
  - info-pass-through (DN level)
  - line-type (DN level)
  - oosp-transfer-enabled (DN level)
  - reject-call-incall (DN level)
  - reject-call-notready (DN level)
  - sip-cti-control (DN level)
  - sip-enable-sdp-codec-filter (DN level)
  - reuse-sdp-on-reinvite (DN level)
  - sip-hold-rfc3264 (DN level)
  - sip-replaces-mode (DN level)
  - transfer-complete-by-refer (DN level)
  - use-contact-as-dn (DN level)
  - use-display-name (DN level)
  - override-domain (DN level)
  - override-domain-oosp (DN level)
  - override-domain-from (DN level)

  - override-call-type (DN level)
  - preview-interaction (DN level)
  - rfc-2976-dtmf (DN level)
  - reinvite-requires-hold (DN level)
- Added new features for post 8.1.0 releases. See "New in This Release" on page 33.
- Added a feature description for ACD Queue functionality on "ACD Queue" on page 99.
- Added memory limit recommendation for deployments on Linux, Solaris, and AIX. See "Guidelines for Deploying SIP Server on Various Operating Systems" on page 54.
- Updated the following configuration options:
  - sip-enable-100rel (Application level)
  - enable-agentlogin-subscribe (DN level)
  - override-to-on-divert (Application level)
  - internal-registrar-persistent (Application level)
- Added the force-p-early-media configuration option.

GENESYS

Part

1 SIP Server Deployment

Part One of this *SIP Server Deployment Guide* contains deployment information specific to your SIP Server. The information in Part One is divided into the following chapters:

- Chapter 1, "SIP Server Fundamentals," on page 43, provides information about SIP Server architectures and deployment considerations.
- Chapter 2, "SIP Server General Deployment," on page 53, presents configuration and installation procedures for SIP Server.
- Chapter 3, "Starting and Stopping SIP Server," on page 69, describes how, and in what order, to start up T-Server among other Framework components. It also provides possible stopping commands.
- Chapter 4, "SIP Devices Support," on page 77, describes compatibility and configuration information specific to SIP Server, including instructions for setting the DN properties and recommendations for the device configuration.
- Chapter 5, "SIP Server Feature Support," on page 97, describes which features SIP Server supports and how to configure them.
- Chapter 6, "T-Library Functionality Support," on page 405, describes the T-Library functionality that SIP Server supports, known limitations, and error messages.
- Chapter 7, "SIP Server Configuration Options," on page 437, describes configuration options specific to SIP Server.

New in This Release

This section describes new or changed functionality that was introduced in SIP Server 8.1.x releases:

New in Release 8.1.1

The following features are introduced in release 8.1.1 of SIP Server:

- **Asynchronous DNS resolution.** SIP Server can resolve a Fully Qualified Domain Name (FQDN) specified in the `contact` option of a DN using the asynchronous DNS resolution method and place the DN out of service if the FQDN is unresolvable. See "Asynchronous DNS Resolution" on page 219.
- **Enhanced handling of XS requests.** SIP Server can handle different HTTP error responses from SIP Feature Server for Dial Plan extended service (XS) requests in an enhanced way to address connection instabilities and provide a quality response to the origination side. See "Enhanced Handling of XS Requests" on page 214.
- **Strict SIP endpoint registration.** SIP Server in standalone mode can restrict SIP endpoint registration if its IP address is not included in a list of trusted IP addresses. See "Strict SIP Endpoint Registration" on page 366.
- **Treating incoming calls as inbound calls.** SIP Server can treat incoming calls from external callers (agents behind SIP trunks) as inbound calls. See "Treating Incoming Calls As Inbound Calls" on page 369.
- **Passing extended recording metadata to GVP.** SIP Server in standalone mode supports passing the additional GVP parameters (which have Agent Assist supporting key-value pairs (KVPs) and Streaming KVPs) from `AttributeExtensions` of `TRouteCall` to MCP in the recording `INFO` messages, under existing recording metadata. See "Passing Extended Recording Metadata to GVP" on page 403.
- **Secure SIP Signaling.** SIP Server supports the secure SIP signaling schema, or `sips`, in accordance with RFC 5630. See "Secure SIP Signaling" on page 348.
- **Remote Agents with Non-provisioned DNs.** Remote agents and agents with nailed-up connections can use external numbers that are not provisioned in the Configuration Database. See "Configuring Remote Agents with Non-provisioned Phone Numbers" on page 345.
- **SRV address support in `Contact` and `Record-Route` headers.** SIP Server supports the SRV FQDN—FQDN resolving to SRV records—received in the `Contact` or `Record-Route` headers of a SIP message. See "SRV Address Support in Contact and Record-Route Headers" on page 365.
- **Masking sensitive data in SIP messages.** SIP Server can mask sensitive data in SIP messages contained in SIP Server logs. See "Masking Sensitive Data in SIP Messages" on page 276.
- **Recording an Agent Greeting.** SIP Server supports recording of the agent call leg during the personal greeting. See "Recording an Agent Greeting" on page 324.
- **HTTP Live Streaming.** SIP Server must be integrated with MCP version 8.5.161.34 or later. See "HTTP Live Streaming" on page 244.

- **SIP INVITE timeout for individual DNs.** See "Setting SIP INVITE Timeout for Individual DNs" on page 108.
- **Music-on-hold enhancement.** SIP Server lets you customize music for music-on-hold treatments. See "Customizing Music on Hold and in Queue" on page 179.
- **CPU usage overload control.** This feature provides the ability to control SIP Server's CPU usage overload by automatically decrementing the server's log level when the CPU usage overload threshold is reached. See "CPU Usage Overload Control" on page 316.
- **New Standard-level log events to monitor SIP Feature Server availability.** See "SIP Feature Server Log Messages" on page 213.
- **Dial Plan enhancement.** SIP Server supports the SIP Feature Server dial plan as an alternative to the internal SIP Server dial plan. See "Dial Plan" on page 195.
- **Instant Messaging enhancement.** SIP Server supports Instant Messaging for multi-site calls. See "Instant Messaging For Multi-Site Calls" on page 255.
- **No-Answer Supervision enhancement**. You can define SIP Server's default action for setting the state of an agent who was not able to answer the routed call before the `after-routing-timeout` expired. See "Defining After Routing Timeout Action" on page 303.
- **Caller Information Delivery Content for AT&T Trunks enhancement.** SIP Server supports passing the multipart body content received in `INVITE` messages (as described in RFC 5621) to GVP. See "Passing CID Content to SIP Destinations (GVP)" on page 119.
- **From Header in SIP INVITE.** SIP Server provides the ability to modify the `From` header in outgoing SIP `INVITE` messages. See "Modifying the From Header in SIP INVITE" on page 279.
- **Muting/Unmuting a Party in a Conference.** SIP Server allows any conference party on the call to mute or unmute any internal party in a conference. See "Muting/Unmuting a Party in a Conference" on page 167.
- **HTTP Monitoring Interface.** SIP Server provides the ability to monitor various operational statistics for its internal modules and statistics relating to trunks. See "HTTP Monitoring Interface" on page 245.
- **Caller Information Delivery Content for AT&T Trunks.** SIP Server supports passing the multipart body content received in `INVITE` messages (as described in RFC 5621) to URS/ORS. See "Caller Information Delivery Content for AT&T Trunks" on page 117.
- **Alternate Routing enhancement.** SIP Server supports delivering calls to an alternative location in situations in which the Universal Routing Server (URS) or Orchestration Server (ORS) becomes non-operational or unresponsive. See "Alternate Routing for Unresponsive URS/ORS" on page 109.

- **Call Supervision enhancement.** SIP Server supports muting a customer who is on hold to the supervisor and agent(s) who are sharing the call. See "Customer-on-Hold Privacy" on page 143.
- **DTMF Clamping in a Conference.** This feature guards a customer's sensitive credit card information from an agent's ears and from call recording. See "DTMF Clamping in a Conference" on page 220.
- **Find Me Follow Me.** SIP Server supports the SIP Feature Server Find Me Follow Me functionality for any 1pcc and 3pcc calls where Feature Server dial plans are applied to destinations. See "Find Me Follow Me" on page 243.
- **Private Conversations During Conference.** SIP Server supports T-Library requests `TListenDisconnect` and `TListenReconnect`. These requests can be used in a conference with three or more participants. Any agent who is using a T-Library desktop can submit a `TListenDisconnect` request to disconnect any other party from the conference temporarily. See "Private Conversations During Conference" on page 166.
- **Sending Outgoing INVITEs with Multipart Body.** SIP Server supports passing geo-location information formed by the routing strategy in the multi-part body of the outgoing `INVITE` message. See "Sending Outgoing INVITEs with Multipart Body" on page 350.
- **Providing Origination DN Name and Location in EventRinging.** SIP Server provides the origination DN name and location in `EventRinging`. The agent desktop can use this information to collect extended data about the originating party, such as the agent name, and present it to the destination party while the phone is ringing. See "Providing Origination DN Name and Location in EventRinging" on page 338.
- **Providing Call Participant Info enhancement.** SIP Server can distribute information about all call participants, including a supervisor's in-call presence, to logged-in agents by using the SIP `NOTIFY` method and `EventUserEvent` messages in multi-site and complex single-site scenarios. See "Providing Call Participant Info" on page 336.
- **Shared Call Appearance.** SIP Server supports Shared Call Appearance (SCA) that enables a group of SIP phones to receive inbound calls directed to a single destination (shared line); that way, any phone from this group can answer the call, barge-in to the active call, or retrieve the call placed on hold. See "Shared Call Appearance" on page 357.
- **Disabling Media Before Greeting.** SIP Server provides the ability to prevent establishing a preliminary audio/video connection between a caller and an agent before greetings are applied. See "Disabling Media Before Greeting" on page 324.
- **TDeleteFromConference requests.** SIP Server supports `TDeleteFromConference` requests in multi-site deployments. See "Deleting Party From Conference in Multi-site Deployments" on page 165.

- **Geo-location support by GVP enhancement.** See "Geo-Location Support by GVP" on page 392.
- **Nailed-up connection enhancement.** Nailed-up connections can be established on agent login or when an agent is in the Ready state. See "Nailed-Up Connections for Agents" on page 287.
- **Agent Login and State Update on SIP Phones enhancement.** See "Agent Login and State Update on SIP Phones" on page 329.
- **Single-step conference enhancement.** SIP Server supports a `TSingleStepConference` request to an external destination.
- **Call Supervision enhancement.** SIP Server supports switching between supervision modes. See "Switching Between Supervision Modes" on page 142.
- **Keep Alive for TCP connections.** SIP Server provides the ability to detect stale TCP connections between SIP Server and a SIP device using the TCP keep-alive mechanism. See "Keep Alive for TCP Connections" on page 259.
- **Geo-location for MSML-based services: strict matching.** SIP Server supports strict geo-location matching for MSML-based services by ensuring that a call with a particular geo-location is served only by an MSML service within the same geo-location or by an MSML service within the alternate location (if configured). See "Geo-Location for MSML-Based Services: Strict Matching" on page 393.
- **ISCC Path Optimization.** This improves the TEvent Propagation path in trunk optimization scenarios. See "ISCC Path Optimization" on page 378.
- **Hunt Groups enhancement.** Hunt Groups enhancement to support sequential ringing. See"Hunt Groups" on page 245.
- **VXML support for Agent Greeting enhancement.** This enhancement allows an agent to accept, reject, transfer the call, or redirect the call to a new destination. See "VXML Support for Agent Greetings" on page 321.
- **IMS integration enhancement.** See "Routed Calls as Originating or Terminating" on page 248.
- **Nailed-up connections enhancement.** SIP Server supports agents with nailed-up connections in Business Continuity deployments. See the *SIP Server 8.1 High-Availability Deployment Guide*.
- **Video Blocking.** SIP Server provides the ability to block video streams from SDP offers during the call negotiation/establishment process, so video will not be played when a call is established. See "Video Blocking" on page 381.
- **Trunk Capacity Control enhancement.** SIP Server enables control of the number of outgoing and incoming calls to be handled by a specific trunk or a group of trunks in single-site deployments. See "Trunk Capacity Control" on page 372.

- **Dial Plan enhancement.** SIP Server provides the ability for internal and inbound calls coming to a Routing Point to remove overdialed digits from DNIS when the `dnis-max-length` dial-plan rule parameter is specified. See “Removal Overdialed Digits From DNIS” on page 198.
- **Call Recording enhancement.** Call recording can be selectively disabled through a routing strategy by overriding the `record` option configured on a DN. See “DN Recording Override” on page 132.
- **Hardware Sizing Tool.** SIP Server offers the Sizing Tool to evaluate the CPU load and network traffic of SIP Server and SIP Proxy applications in your environment. Download the tool from the Genesys SIP Server documentation.
- **Call Park/Retrieve support.** See “Call Park/Retrieve” on page 119.
- **Call Pickup support.** See “Call Pickup” on page 120.
- **Hunt Group support.** See “Hunt Groups” on page 245.
- **IPv6 support.** See “IPv6 Support” on page 257.
- **Support for Resource Manager in Active-Active HA mode.** See “Active-Active Resource Managers” on page 396.
- **Support for Genesys SIP Proxy.** Starting with 8.1.1 release, SIP Server supports Genesys SIP Proxy, which provides an alternative high-availability option without requiring a virtual IP address. In addition, it provides an interface for SIP communication between SIP devices and SIP Server components. See“SIP Proxy Support” on page 354.
- **High-Availability enhancements.** High-availability functionality has been improved in the following areas:
  - Network-interface card (NIC) status monitoring
  - Recovery after network failure
  - Primary/backup SIP Server synchronization

  Note: HA improvements depend on 8.1.2 Management Framework components.
- **Overload Control enhancement.** This allows SIP Server to control incoming T-Requests. See “Overload Control” on page 313.
- **Network Asserted Identity enhancement.** See “Network Asserted Identity” on page 292.
- **Personal Greeting enhancement.** SIP Server provides the ability to suppress agent greetings for different call types. You can block greetings for internal, consultation, and outbound calls, either globally at the Application-level, or individually per Agent Login. See “Personal Greetings” on page 319.

- **Advice of Charge enhancement.** SIP Server provides the ability to send AoC (Advice of Charge) notifications only when a call is answered (that is, the destination party is in the established state). It is a regulatory requirement in many countries. See "Providing AoC Notifications for Established Calls" on page 102.
- **Dial Plan enhancement**. Support for the `original-dialplan-digits` extension key to provide the original destination number before the dial plan is applied. See "Dial Plan" on page 195.
- **Blind transfer support.** SIP Server supports 3pcc and 1pcc blind transfer operations when a complete transfer operation is performed while the transfer destination party is in the alerting state. To enable this feature, set the `blind-transfer-enabled` option to `true` at either the Application level, or at the DN level of the transfer destination.
- **Call recording enhancement.** SIP Server provides the ability to record a call without recording a music-on-hold treatment when a call is placed on hold. See "Recording Calls Without Music-on-Hold Treatment" on page 128.
- **Call recording alarms**. SIP Server can monitor the health and status of Active Call Recording, and generate an alarm if required. See "Call Recording Alarms" on page 130.
- **TLS encryption** is supported between SIP Server and Active-Active Resource Managers in a deployment where SIP Server high-availability is configured using the F5 Networks BIG-IP Local Traffic Manager. See the *SIP Server 8.1 Integration Reference Manual*.
- **SIP-to-T-Library Mapping enhancement.** SIP Server supports the `EXTRACT_SIP_HEADERS` extension key in the `TMakeCall`, `TInitiateTransfer`, and `TInitiateConference` requests. See "Using EXTRACT_SIP_HEADERS" on page 270.
- **Support for CPD performed by Genesys Media Server.** SIP Server enables you to improve the reliability of silence detection in deployments where CPD is performed by the Genesys Media Server. See "Outbound IP Solution Integration" on page 306.
- **Referred-By Header support**. SIP Server provides the ability to pass the identity of the party, which has originated the transfer, in the SIP URI of the outgoing `REFER` request's `Referred-By` header. See "Referred-By Header Support" on page 172.

New in Release 8.1.0

The following features are introduced in release 8.1.0 of SIP Server:

- **Support for Geo-location in Active Call Recording.** See "Call Recording—Geo-location" on page 136.

- **Call Completion Feature support.** SIP Server supports the features Call Completion on Busy Subscriber (CCBS) and Call Completion on No Reply (CCNR). See “Call Completion Features” on page 114.
- **Additional parameters in Request-URI**. SIP Server can be configured to include additional parameters in the `Request-URI`, where the deployment requires it. For example, `user=phone` in `INVITE` requests to a particular DN. See “Enabling Additional Parameters in Request-URI” on page 183.
- **Support for Server and User-Agent headers**. See “Enabling Server and User-Agent Headers” on page 185.
- **Enhanced MWI support**. SIP Server can send `NOTIFY` requests to endpoint subscriptions regardless of the SIP registration for that endpoint. See “mwi-notify-unregistered-dn” on page 491.
- **Network Attended Transfer support**. SIP Server supports network attended transfer, alternate and reconnect operations in multi-site environments. See “Network Attended Transfer” on page 299.
- **Enhanced Error code handling.** SIP Server supports the following improvements to its error code handling:
  - **Simplified error response handling.** SIP Server supports mapping different error messages from multiple GVP instances to a single consistent error message (typically, 503 Service Unavailable) that it sends to the network. See “Consolidated Error Response” on page 175.
  - **Enhanced error code mapping.** SIP Server supports mapping standard SIP and MSML errors sent by GVP to resulting T-Library messages. See “Mapping Treatment Errors” on page 260.
- **Disconnect on remote agent logout**. SIP Server releases the nailed-up connection when the remote agent logs out. See “Disconnecting the Nailed-Up Connection” on page 290.
- **Call release tracking.** SIP Server supports reporting the identity of which party (agent or customer) is responsible for releasing a particular call. See “Call Release Tracking” on page 138.
- **Alternate ringtones.** SIP Server supports insertion of the SIP `Alert-Info` header into `INVITE` requests, in order to specify a distinctive ring-tone depending on the type of call. See “Alternate Ringtones” on page 102.
- **Diversion header support.** SIP Server supports the `Diversion` header for redirected calls. See “DNS Name Resolution” on page 217.
- **Enhanced Private and Custom header support**. SIP Server supports the following new private and custom header functionality:
  - Support for `P-Early-Media` header, used to control the flow of media in the early dialog state. See “Early Media Private Header” on page 193.
  - Support for `P-Access-Network-Info` header, used to provide access to network information about the user. See “P-Access-Network-Info Private Header” on page 319.

  - Forwarding custom headers—SIP Server can pass custom headers from a `REFER` to an outgoing `INVITE` or `REFER`. See "Forwarding Custom Headers" on page 194.
  - Filtering custom headers—SIP Server can filter custom headers from a T-Library request to an outgoing `INVITE` or `REFER`. See "Filtering Custom Headers" on page 195.
- **MSML-based Call Recording**. SIP Server supports call recording through Media Server Markup Language (MSML), based on integration with the Genesys Media Server. See "Call Recording—MSML-Based" on page 125.
- **Media Server Reliability—MSML.** SIP Server supports the `SUBSCRIBE/NOTIFY` method for providing reliable MSML-based media services. See "Media Server Reliability—NETANN/MSML" on page 276.
- **User to User Information (UUI) support**. SIP Server supports the SIP User-to-User header, as specified in the RFC draft "A Mechanism for Transporting User to User Call Control Information in SIP". It also supports a configurable limit for the length of data included in the UUI header, up to a maximum of 8192 characters. See "User to User Information (UUI)" on page 379 and `sip-max-uui-length` on page 527.
- **Network requests for media services.** SIP Server supports network requests for media services as described in RFC 4240 "Basic Network Media Services with SIP". For general support, see `sip-proxy-uri-parameters` on page 622. For information about supporting network requests through Genesys Media Server, see"Requests from the Network" on page 92.
- **Advice of Charge.** SIP Server supports the transfer of Advice of Charge (AoC) information between the T-Library client that determines the charge and the third-party component that generates the charge. See "Advice of Charge" on page 101.
- **Smart OtherDN handling.** SIP Server supports converting the Agent ID to the corresponding DN in certain T-Library messages where the Agent ID is included as the value of the `OtherDN` field. See "Smart OtherDN Handling" on page 363.
- **Monitoring of consultation calls.** SIP Server supports supervisor monitoring of DNs involved in a consultation call. See "Monitoring Consultation Calls" on page 141.
- **Enhanced logging for multi-threaded mode.** SIP Server can write log files for each module in a multi-threaded mode architecture to a separate log file. See "Multi-Threaded Logging" on page 280.
- **DNS Name Resolution.** SIP Server supports DNS name resolution in accordance with RFC 3263. For example, it includes priority and weight information from returned DNS records when resolving hostnames to multiple corresponding IP addresses. See "DNS Name Resolution" on page 217.

- **Set trunk capacity from routing strategy.** SIP Server supports the use of the `Dest-Capacity` key-value pair in the `Extensions` attribute, as applied by the URS routing strategy, to set the capacity for a targeted Trunk. See "Dest-Capacity" on page 430.
- **SIP Authentication for outbound trunks.** SIP Server can respond to HTTP Digest authentication challenges with authentication parameters configured on the outbound Trunk DN. See "SIP Authentication" on page 352.
- **Configurable domain in the Refer-To header of a REFER message**. See the option description "override-domain-oosp" on page 595.
- **Send CPD results from gateway to GVP using MSML.** SIP Server sends CPD results to GVP in the Outbound IP Solution using the existing MSML dialog, instead of using particular SIP messages depending on the gateway type. See "CPD Performed by Media Gateway" on page 308.
- **Support for graceful shutdown.** Users can shut down applications and solutions gracefully. During this process, applications may be in the new `SUSPENDING` or `SUSPENDED` state before they are finally stopped. For more information, refer to the *Framework 8.1 Deployment Guide*.
- **Genesys Voicemail Server integration**. SIP Server supports integration with Genesys SIP Voicemail Server. For details, see the *Genesys SIP Voicemail 8.1 Deployment Guide*.
- **Additional dial-plan parameters.** SIP Server supports new parameters in the dial-plan rule: "onunreach", "unreach-timeout", and "onnotreg". For details, see "Dial-Plan Rule Parameters" on page 202.

Notes:
- For a list of configuration-option changes that apply to SIP Server, see "Changes from Release 8.0 to Release 8.1" on page 638.
- For a list of new features that are common to all T-Servers, see "New for All T-Servers in 8.1" on page 647 of this document.

GENESYS

Chapter

1 SIP Server Fundamentals

This chapter provides more in-depth information about SIP Server and contains the following sections:

Overview

SIP Server has the same position in the Genesys Media Layer as all Genesys T-Servers. SIP Server is a combined T-Server and a call-switching component, in which the call-switching element functions as a SIP (Session Initiation Protocol) Back-to-Back User Agent (B2BUA). In concrete terms, this means that call switching and control is performed by Genesys—no third-party PBX or ACD system is required. A call's audio signal and its associated data travel on a single network, which eliminates the problems associated with synchronizing separate voice and data networks. Because SIP Server supports the Internet Engineering Task Force (IETF) SIP RFC 3261 suite, it is compatible with the most popular SIP-compatible, off-the-shelf hardware or software.

SIP Server can operate with or without a third-party softswitch. Genesys SIP Server gives the entire Genesys line of products access to SIP networks, offering a standards-based, platform-independent means of taking full advantage of the benefits of voice/data convergence.

SIP Server Architecture

Figure 1 presents a generalized architecture of the SIP Server network.

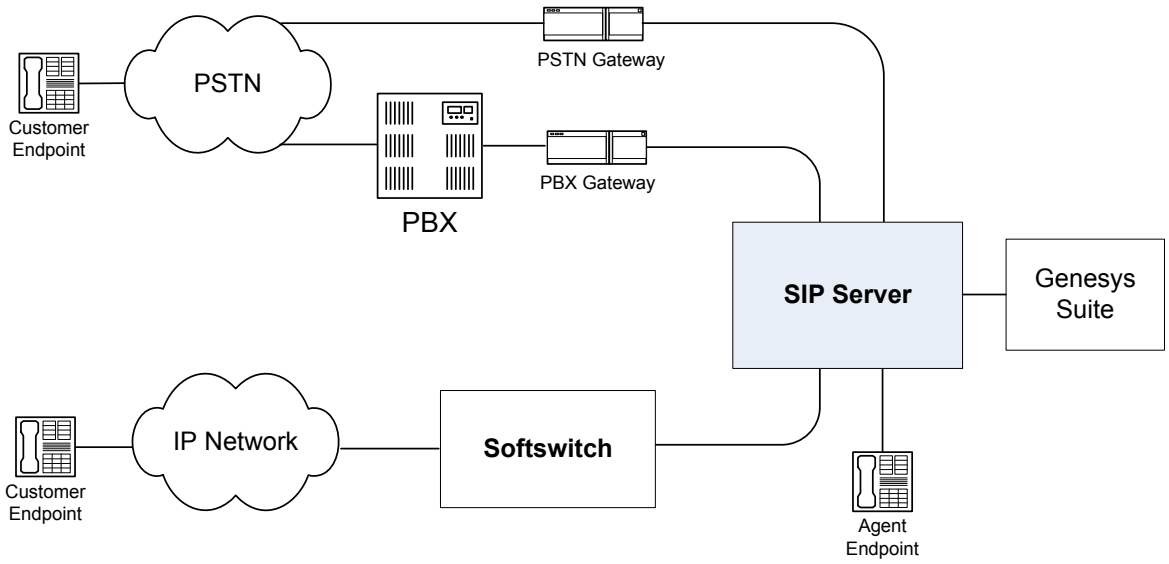

Figure 1: SIP Network Architecture

SIP Server provides all SIP signaling and T-Server functions. Media Server is an optional component that is used to play music-on-hold, music-in-queue and announcements, and to collect DTMF digits. A third-party music server is an optional component that is used as an external music source for music-on-hold. A Multipoint Conference Unit (MCU) is an optional component that is used for third-party call control (3pcc) conference calls. It is also used for silent voice monitoring, whisper coaching, intrusion monitoring, and agent-initiated call recording.

The SIP messages that SIP Server sends or receives are very similar in all configurations, but the destination to which SIP Server sends the SIP requests differs according to the deployment configuration. This mostly applies to the routing of `INVITE` messages. Other messages follow the path established by `INVITE`.

See Chapter 4, “SIP Devices Support,” on page 77 for full details on how to configure the elements of such a network.

SIP Server Deployment Modes

The following SIP Server deployment modes are currently supported:

- Standalone mode
- Application Server mode
- Customer-side proxy mode

Media Server is used as an MCU in these scenarios.

Standalone Mode

Figure 2 illustrates SIP Server in the Standalone mode.

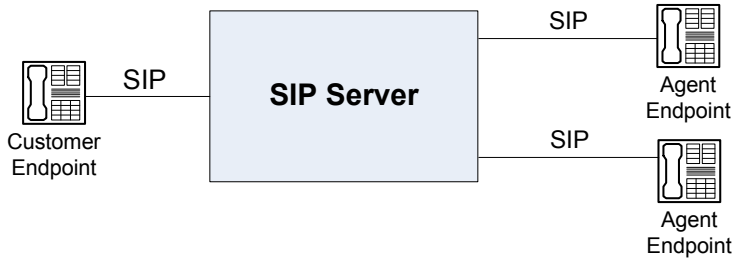

Figure 2: Standalone Mode

In this configuration, SIP Server sends all messages to the addresses of the customer and agent endpoints. The IP addresses in this scenario are determined by SIP Server from either of the following sources:

- Configuration Layer.

  When, for each agent DN, an IP address is configured on the DN `Options/Annex` tab. For example, for DN `1077` on the `Options/Annex` tab in the `TServer` section, you set the `contact` option to the `1077@192.168.2.55` value. This is useful for agent endpoints.

- Lookup in the local registrar.

  When the agent DN is defined with the registrar as `agent1@company.com`, and its SIP endpoint has registered this SIP URI (Uniform Resource Indicator) with the registrar as `1077@192.168.2.55`, the `INVITE` message is sent to the IP address `192.168.2.55`.

- SIP Server resolves the name as it was dialed.

  For example, if the dialed name is `customer@somedomain.com`, the request is sent to the address `somedomain.com`.

Application Server Mode

In the configuration shown in Figure 3, SIP Server is deployed as an Application Server behind a softswitch. This is the most common deployment of SIP Server.

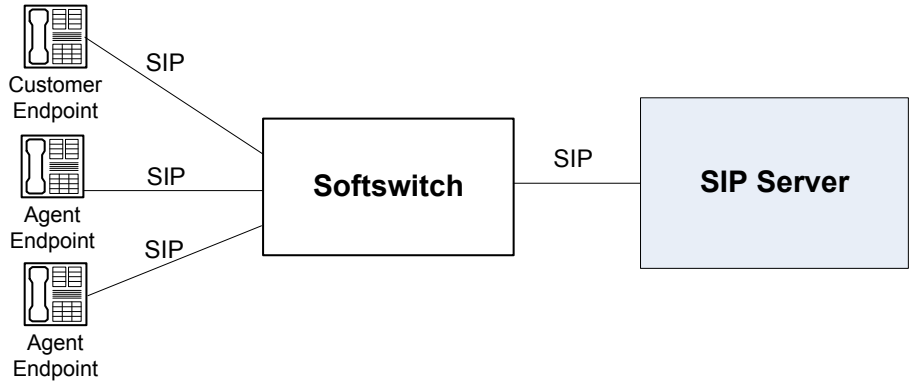

Figure 3: Application Server Mode

In the Application Server mode, SIP Server communicates with a single softswitch. SIP Server is configured to send all `INVITE` requests to the IP address of the softswitch.

In this configuration, the softswitch bypasses SIP Server for direct agent-to-agent calls. As a consequence, agent-to-agent calls are not visible to SIP Server, and it cannot provide any control over these calls.

Customer-Side Proxy Mode

In the configuration shown in Figure 4, a softswitch is deployed between SIP Server and agent endpoints, but customer endpoints communicate directly with SIP Server.

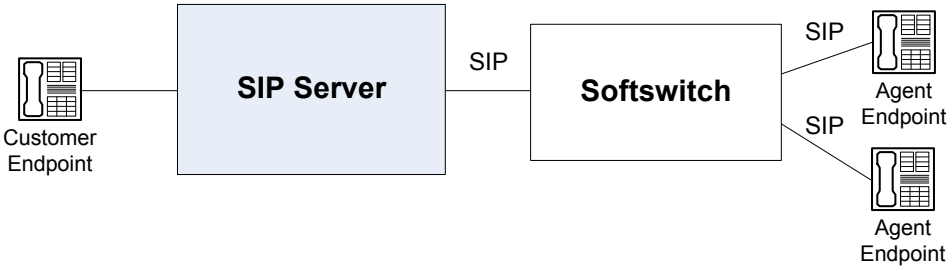

Figure 4: Customer-Side Proxy Mode

All inbound calls (from customers to agents) are routed by SIP Server to a softswitch, the IP address of which is configured in the Configuration Layer.

For outbound calls (from agents to customers), the IP addresses are determined from the `Request URI` message, or they are configured in the Configuration Layer as gateways (DNs of type `Trunk`). Alternatively, SIP Server can resolve the name as it was dialed. For example, if the dialed name is `customer@somedomain.com`, the request is sent to the address `somedomain.com`.

In this configuration, the softswitch bypasses SIP Server for direct agent-to-agent calls. As a consequence, agent-to-agent calls are not visible to SIP Server, and it cannot provide any control over these calls.

Media Server Deployment Architecture

Figure 5 illustrates one possible deployment architecture for a third-party media server (such as a music server or MCU), or for Genesys Media Server used in conjunction with SIP Server and Genesys business applications.

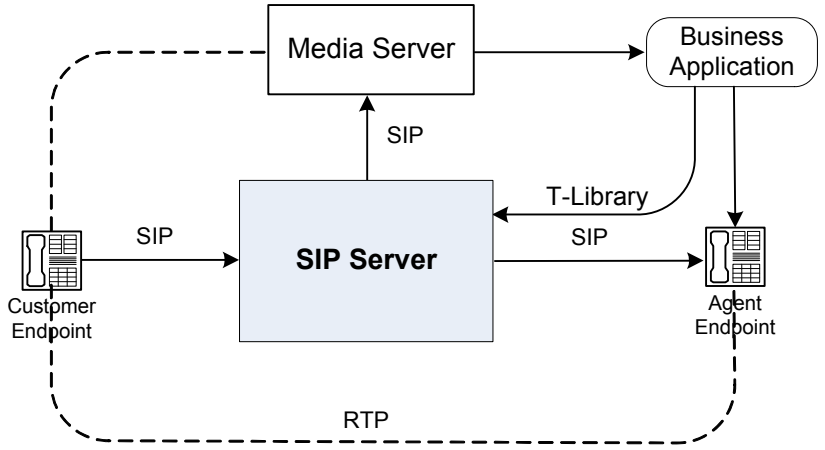

Figure 5: Media Server Deployment Architecture

Call Scenario

1. A call arrives and is established with a contact center agent. SIP Server operates as a SIP B2BUA and maintains two separate SIP dialogs, one for the customer and one for the agent. The RTP stream is negotiated between the customer and agent endpoints directly, using the codec. SIP Server provides flexibility with manipulation of SDP information.
2. The agent invokes a call-hold operation either by using a `THoldCall` request to SIP Server, or by pressing the `Hold` button on the endpoint.
3. SIP Server selects a media server in sequence from among all configured servers with the same priority, and then establishes a new SIP dialog to the music server. SIP Server then sends the `INVITE` message to the customer session to connect the RTP stream to the music server, and then re-`INVITE`s the agent session to stop RTP traffic from it.
4. When the agent invokes a call-retrieve operation either by using the `TRetrieveCall` request, or by pressing the `Retrieve` button on the endpoint, SIP Server terminates the music dialog by sending a `BYE` message, and then re-`INVITE`s the customer session to connect the RTP stream with the agent endpoint.

Redundant SIP Servers (High Availability)

SIP Servers can operate in a high-availability (HA) environment, providing you with redundant systems. One basic principle of redundant SIP Servers is the standby redundancy type, which dictates how quickly a backup SIP Server steps in when the primary SIP Server goes down. The Framework Management Layer currently supports two types of redundant configurations: `warm standby` and `hot standby`.

SIP Server in an HA configuration differs from most Genesys T-Servers in the role it performs in the SIP network. It is not a switch, but it does have traditional switching capabilities.

Supported HA Configuration Models

There are several options available for setting up a high-availability SIP Server deployment. Each approach has benefits and drawbacks. For more information about the different HA models, as well as detailed configuration information, see the *Framework 8.1 SIP Server High-Availability Deployment Guide*.

Note: If you have to stop SIP Server running in HA mode, you must first promote it to a backup role. Likewise, you must do this if you have to reboot or stop the host on which the primary SIP Server is running.

Load Balancing

Figure 6 illustrates a load-balancing architecture for situations in which the call rate exceeds the capacity of a single SIP Server.

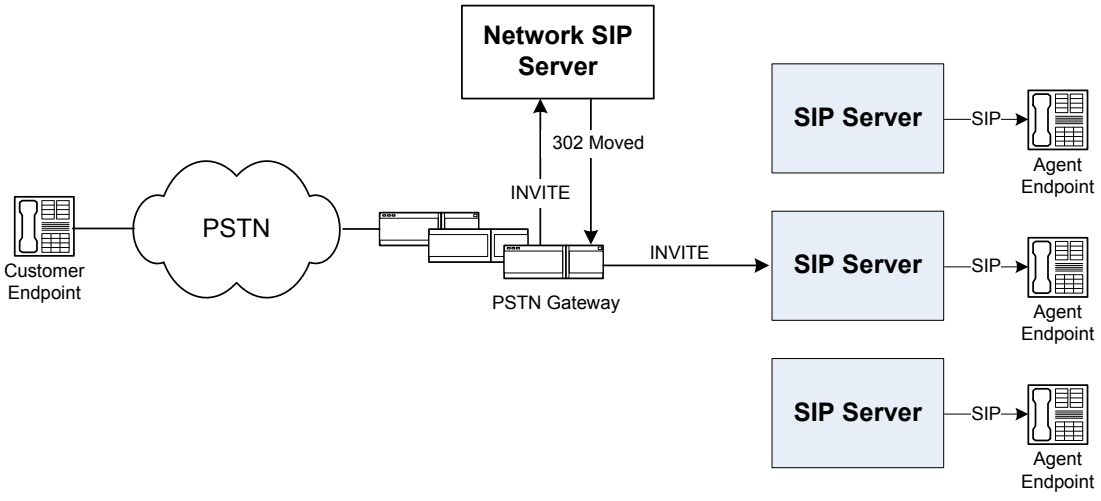

Figure 6: SIP Server Sample Load-Balancing Configuration

In this configuration, all inbound calls first arrive at the Network SIP Server, which performs all initial routing. The routing on the Network SIP Server is either direct to agents, or to the second tier of Routing Points on the SIP Server.

Routing inbound calls directly to agents assumes the following:

- Multiple TDM-to-VoIP gateways (or incoming SIP firewalls for pure VoIP calls) are deployed to provide sufficient call capacity.
- Multiple SIP Servers are deployed.
- One or more Network SIP Servers are deployed. Given the high throughput of a Network SIP Server, it is very likely that a single SIP Server will be sufficient for most deployments.
- Gateways are configured to send all calls to the Network SIP Servers. If multiple Network SIP Servers are required, you can partition gateways, so that they send calls to different SIP Servers; or you can configure each gateway to send calls to one of the Network SIP Servers, based, for example, on call origination or destination.

As a result, the following provides a generalized call flow:

1. Network SIP Servers communicate with Universal Routing Server (URS) (not shown in Figure 6 on page 48). URS selects an agent on one of the SIP Servers by using real-time state information available from the SIP Servers via Stat Servers (not shown). URS responds to the Network SIP Server with the agent's DN and the location name of the SIP Server.
2. Network SIP Server communicates the `ConnID` and attached data to the selected SIP Server. It then responds to the PSTN gateway with a `302` message containing the IP address and External Routing Point number on the destination SIP Server.
3. The gateway processes the `302` message, and then sends a new `INVITE` message to the selected SIP Server.
4. SIP Server receives the `INVITE` message from the External Routing Point, matches the call, and reroutes it to the selected agent by means of Inter Server Call Control (ISCC).

Multi-Threaded Architecture

The SIP Server application is made up of several internal modules that are able to run separately from one another, as individual threads in their own apartments, using internal interfaces to communicate. This multi-threaded organization allows SIP Server to take advantage of computers with multiple CPUs. By running the modules in separate threads, SIP Server can execute different tasks in parallel, simultaneously across several CPUs. This parallel processing enables SIP Server to handle more calls over a given length of time, as the number of processors available in the system is increased.

Multi-Threaded Versus Single-Threaded Mode

Starting in release 8.0.3, using the `sip-link-type` option, SIP Server can be configured to run in either multi-threaded mode, or as a single thread for backward compatibility.

If running in single-threaded mode, the SIP Server Main thread is responsible for processing T-Library requests, distributing Events, managing SIP calls, and processing SIP signaling. If running in multi-threaded mode, the Main thread functionality is split into three threads, each performing the following functions:

- The T-Server thread processes T-Library requests and distributes Events
- The Call Manager thread manages SIP calls and processes SIP signaling (except `OPTIONS` messages)
- The Service Checker thread performs Active Out-of-Service Detection (`OPTIONS` messages)

In both single-threaded and multi-thread modes, SIP Server runs the following threads:

- The SIP transport layer thread dispatches SIP messages
- The Operational Information thread collects and reports statistics; performs NIC monitoring
- A number of auxiliary threads

Logging

By default, in multi-threaded mode SIP Server only logs messages from the T-Library thread into a single log file. However, using the `x-sip-log` option, you can configure SIP Server to create separate log files to handle messages from the other threads. Log messages from each separate thread do not mix into a single file. For details, see “Multi-Threaded Logging” on page 280.

Multi-Site Support

SIP Server, like any conventional T-Server, is built with the T-Server Common Part that contains the ISCC component responsible for call data transfer between multiple sites. Currently, SIP Server supports the following ISCC transaction types: `route`, `direct-notoken`, `direct-uui`, `pullback`, and `reroute`. However, `direct-uui` is supported only in a pure SIP environment.

For instructions on installing and configuring a multi-site environment, including information on the ISCC features, please see Chapter 9, “Multi-Site Support,” on page 659.

Next Steps

Now that you have gained a general understanding of the roles and features available in SIP Server, you are ready to learn how SIP Server is installed and configured. That information is presented in the next few chapters of this *Deployment Guide.* So unless you are already familiar with SIP Server deployment and operation procedures, continue with Chapter 2, "SIP Server General Deployment," on page 53. Otherwise, you may want to proceed to Chapter 5, "SIP Server Feature Support," on page 97, where you will find information about feature configurations that SIP Server supports.

GENESYS

Chapter

2 SIP Server General Deployment

This chapter contains general information for the deployment, configuration, and installation of your SIP Server. You may have to complete additional configuration and installation steps specific to your SIP Server and devices. You will find these steps in Part One of this document.

This chapter contains the following sections:

Note: You *must* read the *Framework Deployment Guide* before proceeding with this SIP Server guide. That document contains information about the Genesys software you must deploy before deploying SIP Server.

Prerequisites

SIP Server has a number of prerequisites for deployment. Read through this section before deploying your SIP Server.

Software Requirements

Framework Components

You can only configure SIP Server after you have deployed the Configuration Layer of Genesys Framework. This layer contains DB Server, Configuration Server, and Genesys Administrator Extension (GAX). If you intend to monitor or control SIP Server through the Management Layer, you must also install and configure components of this layer, such as Local Control Agent (LCA), Message Server, and Solution Control Server (SCS), before deploying SIP Server.

Refer to the *Framework Deployment Guide* for information about, and deployment instructions for, these Framework components.

Refer to the *Genesys Administrator Extension Deployment Guide* for information about deploying GAX.

Media Layer and LCA

To monitor the status of components in the Media Layer through the Management Layer, you must load an instance of LCA on every host running Media Layer components. Without LCA, Management Layer cannot monitor the status of any of these components. If you do not use the Management Layer, LCA is not required.

Supported Platforms

Refer to the *Genesys Supported Operating Environment Reference Guide* for the list of operating systems and database systems supported in Genesys releases 8.x.

For UNIX-based (UNIX) operating systems, also review the list of patches Genesys uses for software product builds, and upgrade your patch configuration if necessary. A description of patch configuration is linked to installation `read_me.html` files for the Genesys applications that operate on UNIX, and is available within the installation packages.

Guidelines for Deploying SIP Server on Various Operating Systems

Genesys recommends running SIP Server on Windows or Linux operating systems. AIX and Solaris are also supported.

SIP Server must be run as a service or daemon in the background under an account with appropriate permissions. SIP Server requires permissions to access the network, network configuration, and file system.

SIP Server is started by LCA. On Windows, LCA runs as a service under the "Local System" account. On all platforms, LCA is started as a service or daemon when the host starts. LCA must be configured to be restarted by the operating system if it is stopped or terminated. In some cases, when system settings are changed, LCA and then SIP Server must be restarted for the new settings to take effect. SIP Server inherits its permission from LCA.

The number of allowed network connections for the SIP Server process must be adequate or "unlimited." This should take into account the requirements of other Genesys components, T-Library clients, and SIP Endpoints that communicate with SIP Server over TCP/IP.

The Windows Server default configuration does not limit the number of open connections. Refer to the Microsoft documentation for details.

On Linux, AIX, and Solaris, the maximum number of open files/sockets for the SIP Server process must be set to an adequate value or "unlimited." The `ulimit` settings have effect only for the SIP Server process. A new shell must be forked and the desired limit set, and then SIP Server is started. Otherwise, SIP Server inherits limits from the parent process (LCA).

On Linux, AIX, and Solaris, the maximum core file size and the maximum memory size of the process must be set to an adequate number. Genesys recommends at least 4 GB. See the respective documentation about the `ulimit` parameter.

Antivirus Guidelines

Antivirus software can affect system performance and the call response time. If a customer's security policy requires antivirus software enabled, Genesys recommends enabling it on hosts where SIP Server runs. Ensure that you analyze the performance of all applications on a particular host.

Some applications might be more vulnerable than SIP Server. Consider moving vulnerable applications to a different host.

Genesys does not recommend excluding SIP Server from the antivirus scanning, but in case of a significant overload, consider excluding the following items from the scan:

- The SIP Server installation folder
- Any logs folders
- The `sip_server.exe` process on Windows
- The `sip_server_32` or `sip_server_64` processes on Linux

The antivirus software must not restrict any ports that are used by the Genesys applications.

Hardware and Network Environment Requirements

Hosting

Genesys recommends that you or your IT specialist assign host computers to Genesys software before you start Genesys installation. Remember the following restrictions:

- Do not install all the Genesys server applications on the same host computer.
- When installing a few server applications on the same host computer, prevent them (except for Configuration Server) from using the swap area.

Installation Privileges

During deployment, be sure to log in with an account that will permit you to perform administrative functions—that is, one that has root privileges.

Server Locations

Refer to the "Network Locations for Framework Components" chapter of the *Framework Deployment Guide* for recommendations on server locations.

Supported Platforms

Refer to the *Genesys Supported Media Interfaces Reference Manual* for the list of supported switch and PBX versions.

Licensing Requirements

All Genesys software is licensed—that is, it is not shareware. Genesys products are protected through legal license conditions as part of your purchase contract. However, the level of technical license-control enforcement varies across different solutions and components.

Before you begin to install SIP Server, remember that, although you may not have had to use technical licenses for your software when you deployed the Configuration and Management Layers in their basic configurations, this is not the case with the Media Layer.

SIP Server requires seat-related DN technical licenses to operate even in its most basic configuration. Without appropriate licenses, you cannot install and start SIP Server. If you have not already done so, Genesys recommends that you install License Manager and configure a license file at this point. For complete information on which products require what types of licenses, and on the installation procedure for License Manager, refer to the *Genesys Licensing Guide*.

The sections that follow briefly describe the T-Server/SIP Server license types.

Note: Starting with release 7.2, the licensing requirements for T-Server (including SIP Server) have changed from previous releases. Please read this section carefully and refer to the *Genesys Licensing Guide* for complete licensing information.

Licensing Basic Implementations

A standalone SIP Server serving a single site requires licenses to register all DNs it monitors. DNs that agents use in day-to-day contact center operations, such as Extensions and ACD Positions, have to be registered using licenses that control agent seats.

Note: Configure all seat DNs that agents use (Extensions and ACD Positions) in the Configuration Layer. This enables detailed call monitoring through Genesys reporting, and generally allows you to control access to individual DNs.

Licensing HA Implementations

SIP Servers operating with the `hot standby` redundancy type require a special CTI HA technical license, which allows for high-availability implementations, in addition to regular SIP Server licenses. Neither SIP Server in a redundant pair configured for `hot standby` starts if this license is unavailable. Moreover, the primary and backup SIP Servers must use the same licenses to control the same pool of DNs. If your SIP Servers are configured with the `hot standby` redundancy type, order licenses for CTI HA support.

Licensing Multi-Site Implementations

SIP Servers performing multi-site operations require licenses that allow for such operations, in addition to regular SIP Server licenses. If some of your SIP Servers are configured for multi-site routing while others are not, either order licenses for multi-site support for all SIP Servers or install an additional License Manager to handle the SIP Servers involved in multi-site routing.

Note: You do not need licenses for multi-site support if some SIP Server clients include the local location as the `location` attribute value in their requests for routing within the same site.

Configuring License Files

You need a license to configure and install Media Layer components. Genesys recommends that, if you have not already done so, at this point you:

1. Install License Manager.
2. Configure license files.

Note: If you use the `<port>@<server>` format when entering the name of the license server during installation, remember that some operating systems use `@` as a special character. In this case, the installation routine is unable to write license information for SIP Server to the Configuration Layer or the `run.sh` file. Therefore, when you use the `<port>@<server>` format, you must manually modify the command-line license parameter after installing SIP Server.

For information about which products require what types of licenses and for the installation procedure for License Manager, refer to the *Genesys Licensing Guide*.

About Configuration Options

Configuring SIP Server is not a onetime operation. It is something you do at the time of installation and then in an ongoing way to ensure the continued optimal performance of your software. You must enter values for SIP Server configuration options on the `Application Options` tab of your SIP Server `Application` object in the Configuration Layer or Genesys Administrator Extension (GAX). The instructions for configuring and installing SIP Server that you see here are only the most rudimentary parts of the process. You must refer extensively to the configuration options chapters located in Part One and Part Two of this book. Pay particular attention to the configuration options specific to SIP Server.

Configuration options common to all T-Servers, independent of switch type, are described in Chapter 11, "T-Server Common Configuration Options," on page 737. *SIP Server-specific* configuration options are described in Chapter 7, "SIP Server Configuration Options," on page 437. SIP Server also supports unified Genesys log options, as described in the Chapter 10, "Common Configuration Options," on page 715.

Options that configure values for the TSCP (T-Server Common Part) software in your SIP Server are common to all T-Servers. Options based on the SIP custom features apply to your SIP Server only. Familiarize yourself with both types of options. You will want to adjust them to accommodate your production environment and the business rules that you want implemented there.

Network Considerations

This section is for system administrators, contact center operations heads, and developers who are planning to deploy Genesys SIP Server.

Deploying SIP Server is similar in many ways to deploying other components of the Genesys Framework, with the significant exception that the voice signal is carried over the data network. This has serious implications for network planning and server sizing. The primary purpose of this section is to highlight the major planning and resource concerns you face in rolling out SIP Server, and to explain how it overlaps with the underlying data network. However, this section is not intended to be an exhaustive guide to network planning. Refer to the *Framework Deployment Guide* for further help with Framework rollout.

The performance of SIP Server is directly linked to that of the underlying data network. It is essential that you perform a proper network audit to ensure that the data network has been properly sized and tuned for real-time (voice) packet transport. This section discusses the factors that affect overall performance of an IP-based configuration, and provides some general rules to follow when deploying SIP Server.

Voice Quality

The following factors have a direct impact on voice quality:

- Network latency—Overall network delay.

  To minimize network latency and ensure acceptable voice quality, you need to tune the network to prioritize real-time voice packets. There are various available schemes for prioritizing voice packets, depending on the IP router vendor.
- Packet loss—Voice packets that are dropped for various reasons (physical media error, timeout due to network congestion, and so on).

  Packet loss is a function result of several factors, including network bandwidth.
- Packet jitter—Variation in voice packet arrival times.

  You can minimize packet jitter by using a jitter buffer at the endpoint device. As a general rule, you must set the buffer size to the maximum anticipated deviation from the typical interpacket emission time.

Other factors that influence voice quality include:

- Packet misordering—Packets arrive in the wrong order (similar to packet loss).
- Type of codec used—Codecs that do not compress the audio signal produce better voice quality but use greater bandwidth.
- Silence suppression—Silence suppression can save bandwidth, but it can also impact voice quality.

Bandwidth Requirements

Determining the bandwidth requirements for the underlying data network is another critical step in achieving proper performance and voice quality. Bandwidth requirements for a video connection are, of course, much higher. Genesys recommends that you verify network performance and voice quality by conducting performance tests and measurements in a lab environment prior to production rollout.

For an IP/Ethernet network, two factors that affect bandwidth requirements are:

- Codec used.
- Protocol headers.

When estimating actual network bandwidth needs, you must also consider such factors as network efficiency and utilization.

Note: Genesys recommends careful network planning to avoid conditions which result in excess latency or packet loss.

Remote Agent Configuration

SIP Server's remote agent capabilities range from a single remote agent, to a group of remote agents in a branch office environment. The distributed nature of branch office and remote agent architectures adds to the complexity of network sizing and tuning.

Bandwidth and Network Tuning

Just as for local network deployment of a VOIP-based system, you must, if at all possible, allot proper bandwidth for voice communication and tune the underlying network for real-time media. Remote agents using a dial-up connection require greater bandwidth (at least 33 Kbps, with 56 Kbps recommended) because of the extra network overhead. This assumes use of the G.723.1 codec, although some dial-up connections may accommodate G.729. A Digital Subscriber Line (DSL) connection is a better alternative than a dial-up connection. The choice of remote access method is important—avoid sending voice communication over an unmanaged data network, such as the public Internet, where voice quality cannot be guaranteed.

For a branch office environment, network bandwidth requirements depend on the number of agents. Again, wide-area network (WAN) connectivity to the corporate LAN must be tuned for real-time voice communications. You need to ensure that service-level agreements from your virtual private network (VPN) provider give details of such requirements. End-to-end network latency must not exceed 250 msec.

Firewalls

This release of SIP Server provides no explicit support for Network Address Translation (NAT). Genesys recommends using virtual private networks (such as PPTP) and ensuring that all agents are on the same network, without NAT translators between the agents and SIP Server.

Deployment Sequence

This is the recommended sequence to follow when deploying SIP Server.

Task Summary: SIP Server Deployment Sequence

Objective	Related Procedures and Actions
1. Deploy Configuration Layer objects and ensure Genesys Administrator Extension is running.	See the *Framework Deployment Guide* for details at: https://docs.genesys.com/Documentation/FR/latest/Dep/Welcome and the *Genesys Administrator Extension Deployment Guide* at: https://docs.genesys.com/Documentation/GA/latest/Dep/Welcome
2. Deploy Network objects (such as Host objects).	See the *Framework Deployment Guide* for details at: https://docs.genesys.com/Documentation/FR/latest/Dep/DepHosts
3. Deploy the Management Layer.	See the *Framework Deployment Guide* for details at: https://docs.genesys.com/Documentation/FR/latest/Dep/DepMgmtLayer
4. Deploy SIP Server.	See "Deployment of SIP Server" on page 62.
5. Test your configuration and installation.	See Chapter 3, "Starting and Stopping SIP Server," on page 69.

Note: If, during the installation procedure for any of the Genesys applications, the script warns you that Configuration Server is unavailable and that the configuration cannot be updated, continue with the installation. Following the installation, you must complete the information on the `Start Info` tab to ensure that SIP Server will run.

Deployment of SIP Server

Deploying SIP Server manually requires that you configure a number of different objects in the Configuration Layer prior to setting up your SIP Server objects and then installing SIP Server. This section describes the manual deployment process.

Configuration of Telephony Objects

This section describes how to manually configure SIP Server telephony objects. For information about configuring SIP Server telephony objects using Genesys Administrator Extension (GAX), refer to the *Genesys Administrator Extension Help*.

Recommendations

Genesys recommends registering (configuring) only those entities you plan to use in the current configuration. The more data there is in the Configuration Database, the longer it takes for the CTI setup to start, and the longer it will take to process configuration data. Remember that adding configuration objects to the Genesys Configuration Database does not cause any interruption in contact center operation.

Depending on how much work is required to manually configure all applications and objects, consider registering more `Person` objects first, with a set of privileges that lets them perform configuration tasks.

Switching Offices

Your telephony network may contain many switching offices, but you should only configure those that are involved with customer interactions.

Using GAX, be sure to register a `Switching Office` object of type `SIP Switch` that accommodates your `Switch` object under `Environment`.

Note: The value for the switching office name must not have spaces in it.

Switches

1. Configure a `Switch` object for each switch on your telephony network. Assign each `Switch` object to the appropriate SIP Server `Application` object.
2. If implementing the multi-site configuration, specify access codes for all switches on the network so that the call-processing applications can route and transfer calls between switches.

Two types of access codes exist in a Genesys configuration:

- Default access codes that specify how to reach this switch from any other switch in the Genesys environment.
- Switch-to-switch access codes that specify how to reach a particular switch from any other switch. Use this type when either a nondefault dial number or routing type is required between any two locations. When a switch-to-switch access code is configured, its value has a higher priority than that of a default access code.

See Chapter 9, "Multi-Site Support," on page 659, for step-by-step instructions.

Note: When the numbering plan uses unique directory number (DN) assignment across sites and multi-site routing is not used, you do not have to configure access codes.

DNs and Agent Logins

For each SIP Server for which you are configuring DNs, you must configure all DNs that agents and their supervisors use in day-to-day contact center operation—so-called *seat-related DNs*—such as Extensions and ACD Positions. Otherwise, SIP Server does not register such DNs.

1. To configure telephony objects within each switch, consult the switch documentation. For configuration information specific to your SIP devices, see Chapter 4, "SIP Devices Support," on page 77.
2. Check the numbering plan for different types of DNs, to see if you can save time by registering Ranges of DNs. Usually, DNs of the same type have consecutive numbers, which will make an otherwise tedious configuration task easy. `Agent Login` objects almost always have consecutive numbers, which means you can register them through the Range of Agent Logins feature as well.
3. If you plan to use Virtual Queues and Virtual Routing Points in the contact center operation, Genesys recommends registering them after you have outlined the call-processing algorithms and identified your reporting needs.

Note: Remember that CTI applications, not the switch, generate telephony events for DNs of these types.

Multi-Site Operations

See the section, "Configuring Multi-Site Support" on page 700, for information on setting up DNs for multi-site operations.

Configuration of SIP Server

Use the *Framework Deployment Guide* to prepare accurate configuration information. You may also want to consult *Genesys Administrator Extension Help,* which contains detailed information about configuring objects.

Recommendations

Genesys recommends using an Application Template when you are configuring your SIP Server application. The Application Template for SIP Server contains the most important configuration options set to the values recommended for the majority of environments. When modifying configuration options for your SIP Server application later in the process, you can change the values inherited from the template rather than create all the options by yourself.

Procedure: Configuring SIP Server

Start of procedure

1. Follow the standard procedure for configuring all `Application` objects to begin configuring your SIP Server `Application` object. Refer to the *Framework Deployment Guide* for instructions.
2. In a multi-tenant environment, specify the `Tenant` to which this SIP Server belongs on the `General` tab of the `Properties` dialog box.
3. On the `Connections` tab, add all Genesys applications to which SIP Server must connect.

   Note: For multi-site deployments, you should also specify SIP Server connections on the `Connections` tab for any SIP Servers that may transfer calls directly to each other.

4. On the `Application Options` tab, specify values for configuration options as appropriate for your environment.

   Note: For SIP Server option descriptions, see Chapter 7, "SIP Server Configuration Options," on page 437. The configuration options common to all T-Servers are described in the Chapter 11, "T-Server Common Configuration Options," on page 737 chapter. SIP Server also uses common Genesys log options, described in the Chapter 10, "Common Configuration Options," on page 715.

5. In a multi-site environment, you must complete additional SIP Server configuration steps to support multi-site operations; see Chapter 9, "Multi-Site Support," on page 659.

End of procedure

Next Steps

- See "Installation of SIP Server" on page 65.

Installation of SIP Server

The following directories on the Genesys 8.1 SIP Server product CD contain SIP Server installation packages:

- `SIP_Server/<component>/<platform>` for UNIX installations, where `<component>` is SIP Server, and `<platform>` is your operating system.
- `SIP_Server\<component>\windows` for Windows installations, where `<component>` is SIP Server.

Procedure: Installing SIP Server on UNIX

Note: During installation on UNIX, all files are copied into the directory you specify. No additional directories are created within this directory. Therefore, do not install different products into the same directory.

Start of procedure

1. In the directory to which the SIP Server installation package was copied, locate a shell script called `install.sh`.
2. Run this script from the command prompt by typing `sh` and the file name. For example: `sh install.sh`.
3. When prompted, confirm the host name of the computer on which SIP Server is to be installed.
4. When prompted, specify the host and port of Configuration Server.
5. When prompted, enter the user name and password to access Configuration Server.
6. When prompted, select the SIP Server application you configured in "Configuring SIP Server" on page 64 from the list of applications.
7. Specify the destination directory into which SIP Server is to be installed, with the full path to it.

8. If asked which version of the product to install, the 32-bit or the 64-bit, choose the one appropriate to your environment.
9. Specify the license information that SIP Server is to use: either the full path to, and the name of, the license file, or the license server parameters.
10. As soon as the installation process is finished, a message appears announcing that installation was successful. The process places SIP Server in the directory with the name specified during the installation.

End of procedure

Next Steps

- To verify manual installation, go to "Verifying the installation of SIP Server" on page 67.
- To test your configuration and installation, go to Chapter 3, "Starting and Stopping SIP Server," on page 69, and try it out.
- To configure and install redundant SIP Servers, see "Redundant SIP Servers (High Availability)" on page 48.
- To install SIP Servers for a multi-site environment, proceed to Chapter 9, "Multi-Site Support," on page 659.

Procedure: Installing SIP Server on Windows

Start of procedure

1. In the directory to which the SIP Server installation package was copied, locate and double-click `Setup.exe` to start the installation.
2. When prompted, specify the connection parameters to the Configuration Server associated with this SIP Server.
3. When prompted, select the SIP Server `Application` object you configured in "Configuring SIP Server" on page 64 from the list of applications.
4. Specify the license information that SIP Server is to use: either the full path to, and the name of, the license file, or the license server parameters.
5. Specify the destination directory into which SIP Server is to be installed.
6. Click `Install` to begin the installation.
7. Click `Finish` to complete the installation.

By default, SIP Server is installed as a Genesys service (Windows `Services`) with `Automatic` startup type.

End of procedure

Next Steps

- To verify manual installation, go to "Verifying the installation of SIP Server" on page 67.
- To test your configuration and installation, go to Chapter 3, "Starting and Stopping SIP Server," on page 69, and try it out.
- To configure and install redundant T-Servers, see "Redundant SIP Servers (High Availability)" on page 48.
- To install SIP Servers for a multi-site environment, proceed to Chapter 9, "Multi-Site Support," on page 659.

Procedure: Verifying the installation of SIP Server

Purpose: To verify the completeness of the installation of SIP Server to ensure that SIP Server will run.

Prerequisites

- Procedure: Installing SIP Server on UNIX, on page 65
- Procedure: Installing SIP Server on Windows, on page 66

Start of procedure

1. In GAX, click a corresponding `Application` object to open its properties.
2. Verify that the `State Enabled` check box on the `General` tab is selected.
3. Verify that the `Working Directory`, `command-line`, and `Command-Line Arguments` are specified correctly.

End of procedure

Next Steps

At this point, you have configured and installed SIP Server. If you want to test your configuration and installation, go to Chapter 3, "Starting and Stopping SIP Server," on page 69, and try it out. Otherwise, if you want to configure and install redundant SIP Servers, see "Redundant SIP Servers (High Availability)" on page 48. If you want to install SIP Server for a multi-site environment, proceed to Chapter 9, "Multi-Site Support," on page 659.

GENESYS

Chapter

3 Starting and Stopping SIP Server

This chapter describes methods for stopping and starting SIP Server, focusing on manual startup for SIP Server. It contains the following sections:

Command-Line Parameters

You can start and stop Framework components using the Management Layer, Genesys Administrator Extension (GAX), a startup file, a manual procedure, or the Windows Services Manager.

With all these methods, command-line parameters are usually required for a server application in addition to an executable file name.

Common command-line parameters are as follows:

`-host`	The name of the host on which Configuration Server is running.
`-port`	The communication port that client applications must use to connect to Configuration Server.
`-app`	The exact name of an Application object as configured in the Configuration Database.

`-l`	The license address. Use for the server applications that check out technical licenses. Can be either of the following: • The full path to, and the exact name of, the license file used by an application. For example, `-l /opt/mlink/license/license.dat`. • The host name and port of the license server, as specified in the `SERVER` line of the license file, in the `port@host` format. For example, `-l 7260@ctiserver`. **Note:** Specifying the License Manager's host and port parameter eliminates the need to store a copy of a license file on all computers running licensed applications.
`-V`	The version of a Framework component. Note that specifying this parameter does not start an application, but returns its version number instead. You can use either uppercase or lowercase.
`-nco X/Y`	The Nonstop Operation feature is activated; `X` exceptions occurring within `Y` seconds do not cause an application to exit. If the specified number of exceptions is exceeded within the specified number of seconds, the application exits or, if so configured, the Management Layer restarts the application. If the `-nco` parameter is not specified, the default value of `6` exceptions handled in `10` seconds applies. To disable the Nonstop Operation feature, specify `-nco 0` when starting the application.
`-lmspath`	The full path to log messages files (the common file named `common.lms` and the application-specific file with the extension `*.lms`) that an application uses to generate log events. This parameter is used when the common and application-specific log message files are located in a directory other than the application's working directory, such as when the application's working directory differs from the directory to which the application is originally installed. Note that if the full path to the executable file is specified in the startup command-line (for instance, `c:\gcti\multiserver.exe`), the path specified for the executable file is used for locating the `*.lms` files, and the value of the `lmspath` parameter is ignored.

Note: In the command-line examples in this document, angle brackets indicate variables that must be replaced with appropriate values.

Starting and Stopping with the Management Layer or GAX

Procedure: Configuring SIP Server to start with the Management Layer or GAX

Start of procedure

- In the SIP Server Application's Properties:
  - Specify the directory where the application is installed and/or is to run as the Working Directory.
  - Specify the name of the executable file as the command-line.
  - Specify command-line parameters as the Command–Line Arguments.
  - The command-line parameters common to Framework server components are described on page 69.

End of procedure

Note: Before starting an application with the Management Layer or GAX, make sure the startup parameters of the application are correctly specified in the application's Properties.

For instructions on starting and stopping applications using the Management Layer, refer to the *Framework Management Layer User's Guide*.

For instructions on starting and stopping applications using GAX, refer to the *Genesys Administrator Extension Help*.

You can also use the Management Layer or GAX to start a SIP Server that has failed.

To enable SIP Server's auto-restart functionality, select the corresponding check box in the Application's Properties dialog box.

Note that when you start (or restart) an application via the Management Layer, the application inherits environment variables from Local Control Agent (LCA), which executes the startup command. Therefore, you must also set the

environment variables required by the application for the account that runs LCA.

Warning! *Stopping* an application via the Management Layer or GAX is not considered an application failure. Therefore, the Management Layer or GAX does not restart applications that it has stopped unless an appropriate alarm condition and alarm reaction are configured for these applications.

Note: If you have to stop SIP Server running in HA mode, you must first promote it to a backup role. Likewise, you must do this if you have to reboot or stop the host computer on which the primary SIP Server is running.

Starting with Startup Files

Startup files are files with the extension `run.sh` (on UNIX) or `startServer.bat` (on Windows), which installation scripts create and place into the applications' directories during the installations. These files are created for all Framework server applications except:

- Configuration Server (primary or backup) running on Windows.
- Backup Configuration Server running on UNIX.
- DB Server running on Windows.
- LCA running on either Windows or UNIX.

When using a startup file, verify that the startup parameters the installation script inserted in the startup file are correct. Use the following instructions for UNIX and Windows to start those application for which startup files are created. See the appropriate sections in "Starting Manually" on page 73 to identify which applications should be running for a particular application to start.

Procedure: Starting SIP Server on UNIX with a startup file

Start of procedure

1. Go to the directory where an application is installed.
2. Type the following command line:

   ```
   sh run.sh
   ```

End of procedure

Procedure:
Starting SIP Server on Windows with a startup file

Start of procedure

To start SIP Server on Windows with a startup file, use either of these methods:

- Go to the directory where an application is installed and double-click the `startServer.bat` icon.

Or

- From the `MS-DOS` window, go to the directory where the application is installed and type the following command-line:

  ```
  startServer.bat
  ```

End of procedure

Starting Manually

When starting an application manually, you must specify the startup parameters at the command prompt, whether you are starting on UNIX or Windows. At the command prompt, command-line parameters must follow the name of the executable file. On the `Shortcut` tab of the Program `Properties` dialog box, command-line parameters must also follow the name of the executable file.

The command-line parameters common to Framework server components are described on page 69.

If an `Application` object name, as configured in the Configuration Database, contains spaces (for example, `SIP Server`), the `Application` name must be surrounded by quotation marks in the command-line:

```
-app "SIP Server"
```

Before starting SIP Server, be sure that the following components are running:

- DB Server that provides access to the Configuration Database
- Configuration Server
- License Manager

Procedure:
Starting SIP Server on UNIX manually

Start of procedure

- Go to the directory where SIP Server is installed, and type the following command-line:

```
sip_server -host <Configuration Server host>
-port <Configuration Server port> -app <SIP Server Application>
-l <license address> -nco [X]/[Y]
```

End of procedure

Procedure:
Starting SIP Server on Windows manually

Start of procedure

- Start SIP Server from either the `Start` menu or the `MS-DOS` window. If you use the `MS-DOS` window, go to the directory where SIP Server is installed, and type the following command-line parameters:

```
sip_server.exe -host <Configuration Server host>
-port <Configuration Server port> -app <T-Server Application>
-l <license address> -nco [X]/[Y]
```

End of procedure

Verifying Successful Startup

After executing the startup command, you might want to check whether it was successful.

If you used GAX to start SIP Server, check whether GAX displays `Started` or `Service Unavailable` status for the corresponding application. Refer to the "Troubleshooting" section of the *Framework Management Layer User's Guide* if the startup command does not result in either `Started` or `Service Unavailable` status for some period of time.

If you start your SIP Server with startup files or manually, and if you have configured logging to console or a log file, check the log for messages similar to the following:

- SIP Server log file: `Link connected`

Stopping Manually

The following stopping procedures apply to Genesys server applications, such as DB Server, Configuration Server, Message Server, Local Control Agent, Solution Control Server, SIP Server, and Stat Server.

Procedure: Stopping SIP Server on UNIX manually

Start of procedure

To stop a server application from its console window on UNIX, use either of these commands:

- `Ctrl+C`
- `kill <process number>`

End of procedure

Procedure: Stopping SIP Server on Windows manually

Start of procedure

To stop a server application on Windows, use either of these commands:

- To stop a server application from its console window on Windows, use the `Ctrl+C` command.
- To stop a server application on Windows, use the `End Task` button on the Windows `Task Manager`.

End of procedure

Note: If you have to stop SIP Server running in HA mode, you must first promote it to a backup role. Likewise, you must do this if you have to reboot or stop the host on which the primary SIP Server is running.

Starting and Stopping with Windows Services Manager

When starting an application installed as a Windows Service, make sure the startup parameters of the application are correctly specified in the ImagePath in the Application folder in the Registry Editor. The ImagePath must have the following value data:

```
<full path>\<executable file name> -service <Application Name as
Service> -host <Configuration Server host>
-port <Configuration Server port> -app <Application Name>
-l <license address>
```

where the command-line parameters common to Framework server components are described on page 69 and

`-service` The name of the Application running as a Windows Service; typically, it matches the Application name specified in the `-app` command-line parameter.

Framework components installed as Windows Services with the autostart capability are automatically started each time a computer on which they are installed is rebooted.

You can start Framework components installed as Windows Services with the manual start capability with the Start button in Services Manager.

Note: Use the Windows Services window to change the startup mode from Automatic to Manual and vice versa.

Regardless of a component's start capability, you can stop Framework components installed as Windows Services with the Stop button in Services Manager.

Next Steps

This chapter concludes SIP Server general deployment. Refer to subsequent chapters in this guide for detailed reference information and any special procedural instructions that pertain to SIP Server.

GENESYS

Chapter

4 SIP Devices Support

This chapter presents reference information for configuring devices and the switch elements of SIP Server. It contains the following sections:

Overview

SIP devices that represent SIP endpoints are configured using Genesys Administrator Extension (GAX) as the following DN types:

- `Extension` (or `ACD Position`)—An agent's endpoint (SIP phone)
- `Trunk`—Any external number (for example, a gateway access number)
- `Trunk Group`—An internal DN (for example, used to represent GVP in Outbound IP integrations)

> **Note:** For more information about the difference between `Trunk` and `Trunk Group` DNs, see "About Trunk and Trunk Group DNs" on page 79.

- `Voice over IP Service`—SIP services (Music-On-Hold server, Genesys Media Server)
- `Routing Point`—Used internally by SIP Server
- `ACD Queue`—Used internally by SIP Server

Note: DNs of type `External Routing Point` are also supported by SIP Server. They are not specific to SIP Server and are used by the T-Server Common Part (TSCP) component of SIP Server in a multi-site environment.

Table 1 contains cross-reference information on SIP devices and Genesys DN types. Use this information to configure SIP devices properly in the Configuration Layer.

Table 1: Device Type Cross Reference

SIP Device Type	Genesys DN Type
Endpoints (SIP phones)	`Extension` (or `ACD Position`)

Routing Points	`Routing Point`
	`ACD Queue`
	Note: SIP Server does not support `Routing Queue` DNs

Gateway SIP Proxy SIP Server in a multi-site deployment	`Trunk`
MCU	`Voice over IP Service`, with `service type` set to `mcu`
Softswitch	`Voice over IP Service`, with `service-type` set to `softswitch`
Music servers	`Voice over IP Service`, with `service-type` set to `music`
Treatment service	`Voice over IP Service`, with `service-type` set to `treatment`
Recording service	`Voice over IP Service`, with `service-type` set to `recorder`
Application service	`Voice over IP Service`, with `service-type` set to `application`
MSML service	`Voice over IP Service`, with `service-type` set to `msml`.

About Trunk and Trunk Group DNs

For SIP Server, `Trunk Group` DNs are a special class of internal DNs used to handle multiple calls—similar to regular `Trunk` DNs, but with the full range of T-Library messaging needed to track internal call processing and generate reports.

`Trunk` and `Trunk Group` DNs are not interchangeable. The way SIP Server selects the DN, the kind of reporting available to the DN, and the available features are all different.

When to Use Trunk DNs

Use `Trunk` DNs for external devices where:

- The DN needs prefix-based dialing.
- Active Out-of-Service Detection must be enabled.
- The external device needs to be configured in a primary/backup model. In this case you can configure multiple `Trunk` DNs with the same prefix but set to different priority. For details, see "Working with Multiple Devices" on page 386.

When to Use Trunk Group DNs

Use `Trunk Group` DNs to represent an internal module or process that requires full T-Library messaging and reporting.

Typically, `Trunk Group` DNs are used in integrations with other Genesys products or solutions, and the documentation for that product would make it clear when this configuration is required. For example, `Trunk Group` DNs are used to represent GVP in Outbound IP integrations, where T-Library requests must be made on behalf of the DN. This functionality is available only with internal DNs.

Note: Regular GVP integrations for inbound calls use `Trunk` DNs.

Usage Guidelines

Table 2 highlights the main differences between these two types of DNs.

Table 2: Guidelines for Trunk and Trunk Group DNs

Trunk DNs	Trunk Group DNs
• External: `Trunk` DNs are used to represent external SIP devices, like gateways.	• Internal: `Trunk Group` DNs are used to represent internal Genesys modules, like GVP in an Outbound IP integration.
• Limited T-Library messaging: For example, as Trunks represent external devices, `EventRinging` and `EventEstablished` messages are not required.	• Full T-Library messaging: This supports detailed reporting on internal call processing.
• Prefix-based DN selection: SIP Server selects `Trunk` DNs based on the `prefix` option.	• Name-based DN selection: SIP Server selects `Trunk Group` DNs based on the name of the DN only.
• Full-featured: `Trunk` DNs can be configured for a range of features unavailable to `Trunk Group` DNs. For example, ◆ Device selection algorithm.	• Limited features: The features available to `Trunk Group` DNs are limited by design.

Configuring Devices and Services

This section describes how to configure the SIP device types for SIP Server environments. It contains the following sections:

Configuring ACD Queues

Follow a common procedure to configure new DNs in GAX at:
https://docs.genesys.com/Documentation/GA/9.0.0/user/CfgDN

To configure ACD Queues, refer to Table 3.

Table 3: Configuring an ACD Queue

Objective	Key Procedures and Actions
Configure an ACD Queue.	Create a DN with the following properties: • `Number`—Enter the number of the configured DN. This value must be a dialable number on the switch. You must not use the @ symbol or a computer name when configuring this property. • `Type`—Select `ACD Queue` from the drop-down menu.

Configuring MCUs

Follow a common procedure to configure new DNs in GAX at:
https://docs.genesys.com/Documentation/GA/9.0.0/user/CfgDN

To configure a Multipoint Conference Unit (MCU), refer to Table 4.

You can configure multiple MCUs. In this case, SIP Server distributes the load for all MCUs in a round-robin fashion.

Table 4: Configuring an MCU

Objective	Key Procedures and Actions
Configure an MCU.	1. Create a DN with the following properties: • `Number`—Enter the MCU name. This name is used during SIP registration only if the MCU registers with the SIP registrar. If the MCU does not register with the registrar, enter a short description of the MCU for this property. • `Type`—Select `Voice over IP Service` from the drop-down menu. 2. In the `Options` tab, create a section named `TServer`. In the `TServer` section, configure the following options: • `contact`—Specify the contact URI that SIP Server uses for communication with the MCU. See the URI format and option description on page 566. • `oos-check`—(Optional) Specify how often (in seconds) SIP Server checks a device for out-of-service status. • `oos-force`—(Optional) Specify the time interval (in seconds) that SIP Server waits before placing a device that does not respond in out-of-service state when the `oos-check` option is enabled. • `prefix`—(Optional) Specify the starting digits of the number that are used when sending calls to MCU. • `recovery-timeout`—(Optional) Specify whether an MCU is taken out of service when an error is encountered, and for how long it is out of service. • `service-type`—Set this option to `mcu`.

Configuring Endpoints

Follow a common procedure to configure new DNs in GAX at:
https://docs.genesys.com/Documentation/GA/9.0.0/user/CfgDN

To configure SIP endpoints, refer to Table 5.

Note: In order to update the `DN` object, SIP Server must have `Full Control` permission for it. By default, it does not have this permission. You must grant the `System` account `Full Control` permission by changing the `Permissions` on the `DNs` folder object in the Configuration Layer.

Table 5: Configuring Endpoints

Objective	Key Procedures and Actions

Configure endpoints.	1. Create a DN with the following properties:
	• `Number`—Enter the `username` part of the endpoint's Address of Record (AOR) as an alphanumeric string. You must not use the @ symbol or a computer name when configuring this property.
	• `Type`—Select `Extension` (or `ACD Position`) from the drop-down menu.
	2. In the `Options` tab, create a section named `TServer`. In the `TServer` section, configure the following options:
	• `authenticate-requests`—Specify whether incoming SIP requests are treated with an authentication procedure under the following conditions:
	  ◦ The name of the incoming SIP message exists in the list of the `authenticate-requests` parameter.
	  ◦ The option password is configured on the same DN object.
	• `contact`—(Optional, depends on the phone registration) Specify the contact URI that SIP Server uses for communication with the endpoint. See the URI format and option description on page 566.
	• `dual-dialog-enabled`—Set the value to `false` for endpoints that accept only one active SIP dialog and for endpoints that can accept more than one SIP dialog but are not able to place the call on hold or retrieve it through the SIP `NOTIFY` request. Set the value to `false` for Siemens optiPoint phones that are used in re-`INVITE` mode for third-party call control (3pcc) operations.
	• `geo-location`— The `geo-location` set on the `Extension` DN (or `ACD Position` DN) is used to select the `Trunk` DN for outbound and consultation calls.
	• `make-call-rfc3725-flow`—Specify which SIP call flow will be used when a call is initiated by the `TMakeCall` request. Only flow 1 and flow 2 from RFC 3725 are currently supported.
	• `password`—Specify the password for SIP endpoint registration with the local registrar. If it is present, registration attempts are challenged, and the password is verified. If it is not present, the registration is not challenged. The realm for password authentication is configured globally; there is one realm per SIP Server.

Table 5: Configuring Endpoints (Continued)

Objective	Key Procedures and Actions
(continued)	• recovery-timeout—Specify for how long a device that is taken of out of service remains out of service. • refer-enabled—Specify whether the REFER method is sent to an endpoint. The recommended setting is true. • reinvite-requires-hold—(Optional, for Genesys SIP Endpoints only) Specify whether the endpoint is placed on hold by re-inviting it with the hold SDP. • request-uri—Specify the value of the Request-URI address to be used in the INVITE message, if that address is different from the address where the message will be sent. • sip-cti-control—Specify the behavior of the DN representing the SIP endpoint that supports the BroadSoft SIP Extension Event Package.

Configuring Gateways

Follow a common procedure to configure new DNs in GAX at:
https://docs.genesys.com/Documentation/GA/9.0.0/user/CfgDN

To configure gateways, refer to Table 6.

Table 6: Configuring a Gateway

Objective	Key Procedures and Actions
Configure a gateway.	1. Create a DN with the following properties: • Number—Enter the gateway name. This name is used only during SIP registration when the gateway registers with the SIP registrar. If the gateway does not register with the registrar, enter a short description of the gateway for this property. • Type—Select Trunk from the drop-down menu. 2. In the Options tab, create a section named TServer. In the TServer section, configure the following options: • contact—Specify the contact URI that SIP Server uses for communication with the gateway. See the URI format and option description on page 566. **Note:** SIP Server selects a Trunk DN to represent the external party of an inbound call by matching the IP address from the Via header of the incoming INVITE message to the host address of the contact option of the Trunk DN. If there are more than one Trunk DNs with the same contact matching the incoming INVITE, SIP Server may select any of these DNs. That means to work reliably, the set of options affecting further call flows must be the same on all of these DNs. This rule does not affect the prefix option, because it is only involved into selection of the Trunk DN for outgoing calls.

Table 6: Configuring a Gateway (Continued)

Objective	Key Procedures and Actions
(continued)	• `geo-location`—(Optional) Specify the `geo-location` for a particular gateway. If the `find-trunk-by-location` option on the DN is enabled, SIP Server includes the `geo-location` attribute in the procedure that it uses to select an available gateway or trunk for an outbound call. • `oos-check`—(Optional) Specify how often (in seconds) SIP Server checks a device for out-of-service status. • `oos-force`—(Optional) Specify the time interval (in seconds) that SIP Server waits before placing a device that does not respond in out-of-service state when the `oos-check` option is enabled. • `password`—(Optional) Specify the password for gateway registration with the local registrar. This is used for incoming `REGISTER` requests, not for outgoing `INVITE` requests. • `prefix`—(Optional) Specify the initial characters of the number that must match a particular gateway for that gateway to be selected. If multiple gateways match the prefix, the gateway with the longest prefix that matches is selected. • `priority`—(Optional) Specify a gateway priority when deciding a route—a smaller number designates higher priority. If more than one gateway with the same prefix is selected, the gateway with highest priority is normally selected. This option is used to control primary-backup gateway switchover during a failure, and to provide lowest-cost routing. • `refer-enabled`—(Optional) Specify whether the `REFER` method is sent to an endpoint. The recommended setting is `true`. • `recovery-timeout`—(Optional) Specify for how long a device that is taken of out of service remains out of service. • `replace-prefix`—(Optional) Specify the characters that are inserted in the DN instead of the prefix for the gateway. If this option is empty or absent, the initial characters that match the `prefix` option will be removed from the DN.

Configuring Music Servers

Follow a common procedure to configure new DNs in GAX at:
https://docs.genesys.com/Documentation/GA/9.0.0/user/CfgDN

To configure external music servers, refer to Table 7.

Table 7: Configuring a Music Server

Objective	Key Procedures and Actions
Configure a music server.	1. Create a DN with the following properties: • `Number`—Enter the music server name. This name is used only during SIP registration if the music server registers with the SIP registrar. If the music server does not register with the registrar, enter a short description of the music server for this property. • `Type`—Select `Voice over IP Service` from the drop-down menu. 2. In the `Options` tab, create a section named `TServer`. In the `TServer` section, configure the following options: • `contact`—Specify the contact URI that SIP Server uses for communication with the music server. See the URI format and option description on page 566. • `geo-location`—(Optional) Specify the `geo-location` attribute that SIP Server uses to select a particular music service in load balancing scenarios. SIP Server will consider those `Voice over IP Service` DNs whose configured `geo-location` matches the preferred geo-location assigned for the call. • `oos-check`—(Optional) Specify how often (in seconds) SIP Server checks a device for out-of-service status. • `oos-force`—(Optional) Specify the time interval (in seconds) that SIP Server waits before placing a device that does not respond in out-of-service state when the `oos-check` option is enabled. • `recovery-timeout`—(Optional) Specify for how long a device that is taken of out of service remains out of service. • `request-uri`—Specify the value of the `Request-URI` address to be used in the `INVITE` message, if that address is different from the address where the message will be sent. • `service-type`—Set this option to `music` or `moh`.

Configuring Routing Points

Follow a common procedure to configure new DNs in GAX at:
https://docs.genesys.com/Documentation/GA/9.0.0/user/CfgDN

To configure routing points, refer to Table 8.

Table 8: Configuring a Routing Point

Objective	Key Procedures and Actions
Configure a Routing Point.	Create a DN with the following properties: • `Number`—Enter the numeric-only DN number that is easily dialed directly from a phone keypad. You must not use the @ symbol or a computer name when configuring this property. • `Type`—Select `Routing Point` from the drop-down menu. **Note:** SIP Server does not support the use of `Routing Queue` DNs when configuring the SIP Routing Point device.

Configuring Softswitches

Follow a common procedure to configure new DNs in GAX at:
https://docs.genesys.com/Documentation/GA/9.0.0/user/CfgDN

If you deploy proxies or softswitches between SIP Server and any internal DNs or agent endpoints, configure the proxies or softswitches as described in Table 9.

You can configure multiple softswitches in either an active load-balancing configuration or in a primary-standby configuration. For load-balancing, define services with the same priority to each service. For the primary-standby configuration, give higher priority to the primary service entry.

Table 9: Configuring a Softswitch

Objective	Key Procedures and Actions
Configure a softswitch.	1. Create a DN with the following properties: • `Number`—Enter the softswitch name. This name is currently not used for any messaging, but it must still be unique. Enter a short description for this property. • `Type`—Select `Voice over IP Service` from the drop-down menu. 2. In the `Options` tab, create a section named `TServer`. In the `TServer` section, configure the following options: • `contact`—Specify the contact URI that SIP Server uses for communication with the softswitch. On some softswitches this is the same as the public IP address used by endpoints to contact the softswitch. However, other softswitches require a separate port. **Note**: To resolve the contact SIP URI through DNS/SRV request, you must configure Active Out-of-Service detection on this DN. See "DNS Name Resolution" on page 217. • `geo-location`—(Optional) Specify the `geo-location` attribute that SIP Server uses to select trunks for outbound and consultation calls, as well as to determine the softswitch object to be selected to send the call to a DN for inbound calls. • `oos-check`—(Optional) Set this option to enable DNS name resolution for this DN using SRV records. See "DNS Name Resolution" on page 217. For the option value, enter how often (in seconds) you want SIP Server to check the availability of this DN. • `prefix`—(Optional) Specify the initial characters of the number that must match a particular softswitch for that softswitch to be selected. If multiple softswitches match the prefix, the softswitch with the longest prefix that matches is selected. This setting ensures that all resources that belong to this softswitch will have numbers starting with the same digits. This allows SIP Server to select the softswitch (contact) so that `INVITE` requests can reach DNs behind the softswitch for inbound calls and 3pcc operations. • `public-contact`—Specify the public `host:port` pair for a softswitch. This is the public IP address of the softswitch. SIP Server uses this address to fill the destination (`Refer-To`) address in `REFER` requests. On some switches, this is the same as the `contact` address; if this is the case, you do not need to specify this parameter. • `service-type`—Set this option to `softswitch`.

Configuring an Application Service

Follow a common procedure to configure new DNs in GAX at:
https://docs.genesys.com/Documentation/GA/9.0.0/user/CfgDN

To configure an application service, refer to Table 10.

Table 10: Configuring an Application Service

Objective	Key Procedures and Actions
Configure an application service.	1. Create a DN with the following properties: • `Number`—Enter the application server name. • `Type`—Select `Voice over IP Service` from the drop-down menu. 2. In the `Options` tab, create a section named `TServer`. In the `TServer` section, configure the following options: • `contact`—Specify the contact URI that SIP Server uses for communication with the recorder server. See the URI format and option description on page 566. • `service-type`—Set this option to `application`.

Configuring a Recording Service

Follow a common procedure to configure new DNs in GAX at:
https://docs.genesys.com/Documentation/GA/9.0.0/user/CfgDN

To configure a recording service, refer to Table 11.

Note: To configure a recording service DN for use with Genesys Media Server, consult the *Genesys Media Server Deployment Guide*.

Table 11: Configuring a Recording Service

Objective	Key Procedures and Actions
Configure a recording service.	1. Create a DN with the following properties: • `Number`—Enter the recorder server name. • `Type`—Select `Voice over IP Service` from the drop-down menu. 2. In the `Options` tab, create a section named `TServer`. In the `TServer` section, configure the following options: • `contact`—Specify the contact URI that SIP Server uses for communication with the recorder server. • `request-uri`—For emergency recording, specify the value of the `Request-URI` address to be used in the `INVITE` message, if that address is different from the address where the message will be sent. **Note**: This value is only used for emergency call recording, used in conjunction with `emergency-recording-filename`. *(Optional)* You can also append the Request-URI with the following sub-string: `record=<directory>` You can use this to specify a particular directory or name for the recording file. For example, `record=recs/` saves the recording file to the directory `recs`, with the filename taken from the configuration option. • `service-type`—Set this option to `recorder`.

Additional Information

SIP Server can also record a file name when emergency recording is initiated by an agent. See the `emergency-recording-filename` configuration option for more information.

Configuring a Treatment Service

Follow a common procedure to configure new DNs in GAX at:
https://docs.genesys.com/Documentation/GA/9.0.0/user/CfgDN

To configure a treatment service, refer to Table 12.

Table 12: Configuring a Treatment Service

Objective	Key Procedures and Actions
Configure a treatment service.	1. Create a DN with the following properties: • `Number`—Enter the treatment server name. • `Type`—Select `Voice over IP Service` from the drop-down menu. 2. In the `Options` tab, create a section named `TServer`. In the `TServer` section, configure the following options: • `contact`—Specify the contact URI that SIP Server uses for communication with the treatment server. • `service-type`—Set this option to `treatment`.

Configuring an MSML Service

Follow a common procedure to configure new DNs in GAX at:
https://docs.genesys.com/Documentation/GA/9.0.0/user/CfgDN

To configure an MSML service, refer to Table 13.

Table 13: Configuring an MSML Service

Objective	Key Procedures and Actions
Configure an MSML service.	1. Create a DN with the following properties: • `Number`—Enter the MSML server name. • `Type`—Select `Voice over IP Service` from the drop-down menu. 2. Create a section named `TServer`. In the `TServer` section, configure the following options: • `contact`—Set this option to the Resource Manager IP address and port. Use the following format: `sip: <RM_IP_address:RM_SIP_port>` • `prefix`—Set this option to `msml=` *(Optional. Required for conferencing, call recording, and call monitoring)* • `service-type`—Set this option to `msml`. • `subscription-id`—Set this option to the name of the Tenant to which this DN belongs (used for reliability). For a single-tenant deployment, set this option to `Environment`.
	For Outbound Solution only (to support `MakePredictiveCall`—Genesys Media Server functionality is not affected), configure the following options: • `refer-enabled`—Set this option to `false`. • `make-call-rfc3725-flow`—Set this option to `1`. • `ring-tone-on-make-call`—Set this option to `false`. • `cpd-capability`—Set this option to `mediaserver`.

Configuring Agent Logins

SIP Server can work either with softswitches or in stand-alone mode, in which the SIP endpoint communicates directly with SIP Server. In both scenarios, you must configure the `Switch` object in the Configuration Layer. The manner in which you configure your SIP Server must reflect the properties of all the objects that your SIP Server monitors. If a client issues a `TRegisterAddress` request for a DN that is not configured in the Configuration Database, SIP Server generates an `EventError` message.

Because only SIP Server uses agent logins, they do not need to match user information on the softswitch. SIP Server manages the status of agents who use these logins, and allows these agents to log in to the SIP addresses.

Configuring Genesys Media Server

When integrated with SIP Server, the Genesys Media Server provides Real-Time Protocol (RTP) streaming for a variety of media services—treatments, conferences, call recording, and so on—using the Media Server Markup Language (MSML).

About Genesys Media Server

The Genesys Media Server is a module that provides MSML-based media services offered by the Genesys Voice Platform, as well as NETANN-based services for requests coming in from the network. When integrated with SIP Server, it supports the same set of features that were previously provided by Genesys Stream Manager (7.x). In addition to these features, Genesys Media Server also supports new codec formats for voice delivery associated with outbound calling, call parking, call recording, conferencing, and IVR prompting.

MSML-Based Media Services

When enabled for MSML, SIP Server responds to a media service request by sending an `INVITE` message first, to establish a connection with the media server, then an `INFO` message to start the particular service, such as treatment or conference.

Requests from the Network

For NETANN-based requests for media services coming in from the network, the incoming `INVITE` contains a URI that specifies the kind of media service required for the call. SIP Server forwards this request in the `INVITE` to Genesys Media Server, which can then provide the service for the call. This

functionality can be used for network requests for announcements, conferences, and other NETANN-based media services.

Load Balancing of Media Servers

SIP Server performs load balancing of media services across multiple instances of Media Server (Resource Manager and an MCP farm). If a service does not start at a particular instance of Media Server, SIP Server tries the next instance of Media Server.

Media Server Reliability

SIP Server uses the SIP `SUBSCRIBE`/`NOTIFY` model for monitoring active MCP instances, and reconnecting ongoing media services in case a particular MCP instance becomes unavailable. For more information, see "Media Server Reliability—NETANN/MSML" on page 276.

Geo-Location

SIP Server is able to send geo-location information so that the GVP Resource Manager can select the closest Media Server instance to the caller.

For More Information

For more information about the Media Server, see the *Genesys Media Server Deployment Guide*.

SIP Server and Media Server Integration

A SIP Server deployment with Genesys Media Server includes the following components:

- SIP Server
- GVP Resource Manager
- GVP Media Control Platform

SIP Server integrates with the Media Server using a `Voice over IP Service` DN with `service-type` set to `msml`. Only one DN is required for all media services. SIP Server does not communicate directly with the Media Server (MCP), but instead sends the MSML service requests to Resource Manager, which then selects and manages the MCP independently from SIP Server. This allows for efficiencies in scalability and redundancy.

Genesys Media Server Integration

Tables 14 and 15 describe the required configuration to integrate SIP Server with Genesys Media Server. Table 16 describes the steps to enable a ringing period for predictive calls of greater than 32 seconds.

Table 14: Integrating Media Server for MSML

Objective	Key Procedures and Actions
1. Configure SIP Server for MSML.	In the `TServer` section of the SIP Server Application object, configure the following options: • `msml-support`—Set this option to `true`. • `msml-record-support`—Set this option to `true`.
2. Configure the MSML DN.	1. Create a `Voice over IP Service` DN. 2. In the `TServer` section, configure the following options: ◆ `contact`—Set this option to the Resource Manager IP address and port. ◆ `prefix`—Set this option to `msml=` *(Required for conferencing, call recording, and call monitoring)* ◆ `service-type`—Set this option to `msml` ◆ `subscription-id`—Set this option to the name of the Tenant to which this DN belongs (used for reliability). See Table 13, "Configuring an MSML Service," on page 91.
3. Configure GVP components.	Configure the following GVP components to their default settings: • Resource Manager • Media Control Platform **Note:** SIP Server and Resource Manager use the same port `5060`. If both are deployed on the same host, you may have to change port numbers to avoid conflicts. Genesys suggests shifting the port numbers in the Resource Manager options up by 100—from `5060-5067` to `5160-5167`. For more information, see the *Genesys Media Server Deployment Guide*.

Table 14: Integrating Media Server for MSML (Continued)

Objective	Key Procedures and Actions
4. Configure an MCP resource group for MSML services.	1. Create a resource group for the MCP instances that will be used to provide MSML service. 2. Configure the resource group with the following minimum mandatory options: ◆ `load-balance-scheme`—Set this option to `round-robin`. ◆ `monitor-method`—Set this option to `option`. ◆ `port-usage-type`—Set this option to `in-and-out`. ◆ `resource-confmaxsize`—Set this option to `-1`. ◆ `service-types`—Ensure that `msml` is included in the list of service-types. For more information, see the *Genesys Media Server Deployment Guide*.
5. Configure Resource Manager application.	• In the `rm` section, configure the following parameters: ◆ `conference-sip-error-respcode`—Set to `503`. ◆ `resource-unavailable-respcode`—Set to `603`. • In the `monitor` section, configure the following parameter: ◆ `sip.proxy.releaseconfonfailure`—Set to `false`.
6. Create a default IVR Profile.	Create a new IVR Profile to be used as the default for your particular tenant. 1. In the `Voice Platform Profiles` folder, create a new `GVP IVRProfile`. 2. In the Options tab of the IVR Profile, create a `gvp.general` section, adding the following option: ◆ `service-type`—Set to `voicexml`. For more information, see the *Genesys Media Server Deployment Guide*.
7. Configure the Tenant object.	Assign the IVR Profile as the default for your tenant. 1. Select your `Tenant` object, then select `Properties`. 2. In the `Options` tab, create a `gvp.general` section. 3. Add the following options: ◆ `default-application`—Set to the name of the default IVR Profile. ◆ `service-type`—Set to `voicexml`. For more information, see the *Genesys Media Server Deployment Guide*.

Table 15 describes the steps for allowing request for media services to come in directly from the network.

Table 15: Enabling Network Requests for Media Services

Objective	Key Procedures and Actions
Configure GVP Trunks.	Configure a separate `Trunk` DN for each type of NETANN media service. For example, for NETANN announcements, configure the `Trunk` DN as follows: • Set the `prefix` option to `annc`. This matches the `userpart` of the `Request-URI` in the network `INVITE`: `INVITE sip:annc@172.24.129.75:5060;play=greetings.wav SIP/2.0` • Set `sip-proxy-uri-parameters` to `true`. SIP Server will match the prefix to this Trunk, copying the URI from the network `INVITE` to the outgoing `INVITE` it sends to this Media Server `Trunk` DN. For other NETANN media services, create a separate `Trunk` DN with `prefix` configured as follows: • `conf`—Enables network requests for NETANN conferences. • `dialog`—Enables network requests for simple VoiceXML prompt/collect applications.

SIP Server supports an increased maximum ringing period for predictive calls through Genesys Media Server. Table 16 describes the steps to enable a ringing period of greater than 32 seconds.

Table 16: Increasing Ringing Period for Predictive Calls

Objective	Key Procedures and Actions
1. Configure a Genesys Media Server DN.	In the `TServer` section of the DN (`Trunk Group` or `Voice over IP Service`) object, configure the following option: • `predictive-timerb-enabled`—Set this option to `false`.
2. Configure an MCP application.	1. Set the `sip.timer_si` option to a value greater than the `AttributeTimeout` in `TMakePredictiveCall` used by SIP Server to control the call. Typically this setting comes into effect for stuck calls only, or in cases where `AttributeTimeout` is set to `0`. 2. Set the `sessmgr.acceptcalltimeout` option to a value greater than the `sip.timer_si`. This prevents the MCP application from interfering with the SIP level timers.

GENESYS

Chapter

5 SIP Server Feature Support

This chapter describes the advanced functionality that SIP Server supports. It contains the following sections:

ACD Queue

SIP Server supports Automatic Call Distribution (ACD) Queue functionality. With this feature enabled, SIP Server places queued incoming calls on hold until an agent or representative in the organization becomes available. The caller is placed in a simple queue, where each call is answered in the order it is received. During the wait, SIP Server plays music or other announcements to the caller. When an agent logged into the queue becomes available, SIP Server automatically connects the caller to the agent's DN (no manual connection is required).

How It Works

1. A call arrives at an `ACD Queue` DN configured in the SIP Server switch.

   Note: If SIP Server is behind a third-party softswitch, a `Routing Point` DN is used instead of an ACD Queue.

2. Agents are logged into this queue using Genesys Agent Desktop (Interaction Workspace).

   An agent DN can log into only one queue at a time.

SIP Server searches the list of associated DNs for an available agent `Extension` DN.

3. If no logged in agent is available (all agents currently in the Genesys `NotReady` state or currently on a call), the caller is queued.

   If no agent is currently logged into the queue, SIP Server applies alternate routing to avoid a stranded call. See "Alternate Routing for Stranded Calls" on page 106.
4. SIP Server plays music or other announcement to the caller while they wait. The music played to the caller in the queue is configurable. For more information, see the `music-in-queue-file` option, as well as "Customizing Music on Hold and in Queue" on page 179 for details.
5. As soon as an agent `Extension` DN becomes available (Genesys `Ready` state and no active call), SIP Server automatically connects that caller to the DN. Calls with the longest wait time on an ACD Queue are distributed to agents with the longest idle time.
6. After the agent is done with the call, they are placed at the end of the line for receiving new calls.

Feature Configuration

Table 17 describes how to configure ACD Queue functionality.

Table 17: Configuring ACD Queue

Objective	Related Procedures and Actions
1. Configure an ACD Queue DN.	See "Configuring ACD Queues" on page 81.
2. Configure music.	You can define the music file to be played using the following options. These options are listed in order of priority (default-music on the DN takes precedence over all other settings): 1. `default-music` configured on the DN 2. `music-in-queue-file` configured in the Application 3. `default-music` configured on the Application If none of these are configured, SIP Server tries to use the file in the `music/on_hold` folder.

Feature Limitation

SIP Server only supports first-in-line queue functionality. It does not support prioritization based on any other criteria.

Advice of Charge

SIP Server supports the transfer of Advice of Charge (AoC) information between the T-Library client that determines the charge and the third-party component that generates the charge. For example, when integrated into an IP Multimedia Subsystem (IMS), SIP Server is able to add AoC information received from the Orchestration Server (ORS) to the `INFO` message that it sends to the IMS, which then generates the charge.

How It Works

SIP Server receives AoC information in a `TPrivateService` message sent from a T-Library client. This client is responsible for determining whether a charge is required for calls involving a particular DN. If the client decides a particular DN requires a charge, it forwards the required AoC information in a `TPrivateService` request. SIP Server maps the AoC information from the `TPrivateService` to an `INFO` message that it sends the external component responsible for generating the charge (IMS or other switch).

Sample Call Flow

The following sample call flow describes the steps for an incoming call from IMS, with Orchestration Routing Server (ORS) acting as the T-Library client that determines the charge:

1. The IMS sends an inbound `INVITE` request to SIP Server through the IMS `Trunk` DN.
2. The destination DN (Extension, Routing Point) is registered with ORS; ORS receives any Events related to this DN.
3. Business rules in ORS determine a charge for the call to this destination is required.

   Note: SIP Server does not itself decide about the charge; all decisions are made in the T-Library application, in this case ORS.

4. ORS sends AoC information, as well as related parameters, in a `TPrivateService` request to SIP Server. The request includes the following:
   - AoC information as the SIP MIME body in the `AttributeExtensions`
   - DN of AoC sender
   - Connection ID
5. SIP Server copies this AoC information into an `INFO` request that it sends back to the IMS server.
6. The IMS server is responsible for generating the charge for the call.

Providing AoC Notifications for Established Calls

SIP Server provides the ability to send AoC notifications only when a call is answered (the destination party is in the established state). It is a regulatory requirement in many countries.

For this feature to work, SIP Server distributes calls through a Routing Point that is configured with the `divert-on-ringing` option set to `false`. A T-Library client that monitors the Routing Point (for example, URS or ORS) receives the notification that the call is delivered to the destination when the outgoing call is answered. (SIP Server sends `EventRouteUsed, EventDiverted` to its clients.) This notification can be used as a trigger for generating an AoC notification using a `TPrivateService(3018)` request.

SIP Server is able to process this request and send a SIP `INFO` AoC message to the destination even though the Routing Point DN, used to route the call and passed as a value of `AttributeThisDN` of the `TPrivateService(3018)` request, is already released from the call.

Feature Configuration

Use the `sip-enable-aoc-after-established` option to configure AoC notifications for established calls.

Alternate Ringtones

Some endpoints can provide a distinctive ringtone that tells the user what kind of call is arriving on their phone. For example, a triple ring can be used to identify the caller as external to the company. To support endpoints that offer this feature, SIP Server is able to include the SIP `Alert-Info` header in the `INVITE` request that it sends to the endpoint. The value of this header gives the endpoint the information that it needs to start the alternate ringtone—a URI to a ringtone file, or a code that triggers a stored ringtone on the phone itself.

How It Works

If alternate ringtones are configured, when SIP Server receives a T-Library request to initiate a call, it adds the `Alert-Info` header to the resulting `INVITE` request. A typical call flow is as follows:

1. SIP Server receives a T-Library request to initiate, transfer, or conference a call. Alternate ringtone functionality is configured on the Application, in the destination DN, or in the `SIP_HEADERS` Extension of the T-Library request itself.
2. SIP Server inserts the `Alert-Info` header in the `INVITE` to the call destination. The value of this header depends on the configuration and the type of call.

3. On receiving the INVITE, the endpoint reads the information in the Alert-Info header, which tells it where to go fetch the ringtone, or to start playing the stored ringtone on the phone itself.

How the Alert-Info Header is Built

The content of the Alert-Info header is configured using the following options, which can be applied on the Application or DN-level:

- sip-alert-info
- sip-alert-info-external
- sip-alert-info-consult

Note: The sip-alert-info option takes precedence over the make-call-alert-info option.

The value of these options determines the content of the Alert-Info header that, if configured, will be included in the INVITE. For example, the following value points the endpoint to the ringtone file that will be used for external calls:

<http://www.provider.com/tones/internal_caller.pcm>

If alternate ringtones are also configured for external or consultation calls (sip-alert-info-external or sip-alert-info-consult), that configuration takes precedence over sip-alert-info for those types of calls.

In all cases, if the SIP_HEADERS extension in the original T-Library request includes the Alert-Info header, the value in this extension will take precedence and be used in the INVITE sent to the endpoint.

Using the SIP_HEADERS Extension

You can also enable alternate ringtones from within the T-Library request that starts the call operation. In this case, the request must include an Alert-Info key-value pair in the SIP_HEADERS extension.

For an example, see the text in **bold** in the following TRouteCall:

```
message RequestRouteCall
AttributeThisDN'5000'
AttributeConnID006e01886c3d7001
AttributeOtherDN'21101'
AttributeExtensions[371] 00 0B 00 00..
'SIP_HEADERS' 'Alert-Info'
'Alert-Info' '<http://www.provider.com/tones/internal_caller.pcm>'
AttributeDNIS '5000'
AttributeRouteType 1 (RouteTypeDefault)
AttributeReferenceID 9
```

This TRouteCall request would result in the following SIP INVITE:

```
INVITE sip:21101@ DestinationHost:21101 SIP/2.0
From: <sip:7102@ SourceHost:7102>;tag=28B10B44
To: <sip:21101@ DestinationHost >
Call-ID: 931E620E-F3F9
CSeq: 1 INVITE
Content-Length: 145
Content-Type: application/sdp
Contact: <sip: SourceHost >
Alert-Info: <http://www.provider.com/tones/internal_caller.pcm>
Max-Forwards: 70
Session-Expires: 1800;refresher=uac
Min-SE: 90
Supported: timer
```

The Alert-Info header can be defined using the SIP_HEADERS extension in any of the following T-Library requests:

- TMakeCall
- TInitiateTransfer
- TInitiateConference
- TSingleStepTransfer
- TSingleStepConference
- TRouteCall
- TPredictiveCall
- TRedirectCall

Special Codes for Built-In Ringtones

For endpoints that use ringtones built into the phone itself, you must configure SIP Server to build the Alert-Info header so that it includes the code required by the endpoint to invoke the alternate ringtone. For example, the following string is used by some endpoints to trigger a distinctive external ringtone:

```
<http://notused.invalid>;info=alert-external
```

The URI portion of the string must be enclosed in angle brackets (the URI itself is empty). The second part of the string contains the code used by the particular endpoint to trigger the ringtone.

Other Uses for the Alert-Info Header

Some endpoints may offer other services that can be triggered using the Alert-Info header. For example, an Auto-Answer feature, where stored messages in the endpoint can be triggered for certain types of calls. In this case, the same configuration rules apply as for enabling alternate ringtones: configure SIP Server to build the Alert-Info header as required for your particular endpoint.

Feature Configuration

Table 18 describes how to enable alternate ringtones, listed in order of highest to lowest priority.

Table 18: Configuring Alternate Ringtones

Objective	Related Procedures and Actions
1. Configure the T-Library request.	In the T-Library client or URS routing strategy, configure the request to include the following: • `SIP_HEADERS`—Add `'alert-info'` to the list of custom SIP headers to be added to the `INVITE`. • Define the header as follows: `'alert-info' '<URI>; parameters'` If present, this configuration takes precedence over all other settings.
2. Configure the DN.	You can configure alert-info related options in any of the following DNs: • `Extension` • `ACD Position` In the `Options` tab of the DN, configure any of the following options: • `sip-alert-info` • `sip-alert-info-external` • `sip-alert-info-consult`
3. Configure the SIP Server Application.	In the SIP Server `Application` object, you can apply any of the same options as at the DN-level: • `sip-alert-info` • `sip-alert-info-external` • `sip-alert-info-consult` DN-level and `SIP_HEADERS` extension take precedence.

Alternate Routing

SIP Server supports call delivery to a variety of alternate default-DN locations to handle complications that can arise during the regular processing of inbound calls. SIP Server also includes a mechanism to delete a call after an inordinate number of routing attempts. These scenarios include the following:

Alternate Routing for Stranded Calls

SIP Server offers alternate routing for stranded calls (calls left waiting in a queue after the last agent logs out) and stranded-on-arrival calls (calls arriving at a queue with no remaining logged-in agents). Two new configuration options, `stranded-calls-overflow` and `stranded-on-arrival-calls-overflow`, are used to configure how SIP Server processes calls stranded in ACD queues. An additional option, `stranded-call-redirection-limit`, is used to limit the number of redirections that SIP Server can make when processing a single stranded call (to avoid looping the call indefinitely).

Feature Configuration

Table 19 describes how to configure stranded call routing.

Table 19: Configuring Stranded Call Routing

Objective	Related Procedures and Actions
Configure for all queues.	In the SIP Server `Application > Application Options > TServer` section, configure these options: • `stranded-calls-overflow`—Enter a list of actions that you want SIP Server to take for all stranded calls to any configured `ACD Queue` DN. **Note:** For a list of valid actions, see "Stranded Calls Overflow Valid Values" on page 107. • `stranded-on-arrival-calls-overflow`—Enter a list of actions that you want SIP Server to take for calls arriving on any empty `ACD Queue` DN. • `stranded-call-redirection-limit`—Set this to a value between `0` and `15`. SIP Server stops trying to redirect stranded calls after the configured number of attempts.

Table 19: Configuring Stranded Call Routing (Continued)

Objective	Related Procedures and Actions
Configure for individual queues.	Go to SIP Server `Switch` > `DNs` folder > individual `ACD Queue` DN > `TServer` section and configure these options: • `stranded-calls-overflow`—Enter a list of actions that you want SIP Server to take for stranded calls to this DN only. • `stranded-on-arrival-calls-overflow`—Enter a list of actions that you want SIP Server to take for calls arriving on this empty ACD Queue DN only (without any logged in agents). **Note:** The DN-level overflow options take precedence. The redirection limit, however, is applied only at the Application-level, globally for all stranded calls.

Stranded Calls Overflow Valid Values

Table 20 describes the valid values for the `stranded-calls-overflow` and `stranded-on-arrival-calls-overflow` options, as well as their related SIP Server actions.

Table 20: Stranded Call Overflow Valid Values

Valid Value	Related Action
`<valid_destination_number>`	SIP Server redirects stranded calls to this number.
`default` or <empty string>	This is the default value which provides backwards-compatible behavior with previous versions of SIP Server. **DN-level:** • If the option is not specified, or contains an empty string or the value `default`, SIP Server uses the value of the Application-level option instead. **Application-level:** • If the option is not specified, or contains an empty string, SIP Server does not perform any stranded call routing. The call remains waiting in the queue.
`recall`	SIP Server sends stranded calls back to the previous distribution device as specified in the `OtherQueue` attribute of the call. If the call was not distributed from a previous device, SIP Server disregards this value, continuing with other values in the list (if available).
`release`	SIP Server releases calls stranded in the queue.

Table 20: Stranded Call Overflow Valid Values (Continued)

Valid Value	Related Action
`none`	The stranded call remains in the queue. Use this value at the DN-level when you do not want the value of the Application-level option to apply to this ACD Queue.
General Rules About the Overflow Values The following general rules apply to both `stranded-calls-overflow` and `stranded-on-arrival-calls-overflow` options: • Valid values are case-sensitive. • If the overflow destination points to the same queue where the call is already stranded, SIP Server skips this value in the list. Similarly, if a loop is detected, SIP Server skips the value. • The value `none` cannot be included in the comma-separated list (valid as a single action only). The remaining values can be combined. If included, the values release and default should be placed last in the list. • When configured on the application-level, this option applies to all ACD Queues on the switch *except* for the overflow destination queue. • The DN-level option takes precedence over the Application-level option.	

Alternate Routing for Unresponsive DNs

SIP Server supports alternate routing for new calls to Genesys SIP endpoints that fail to respond to an `INVITE` request. If the `INVITE` request to a particular endpoint fails to respond before the `sip-invite-timeout` setting expires, SIP Server sends the call to the alternate location specified in the `no-response-dn` option, as configured in the unresponsive DN.

Introduced in SIP Server 8.1.102.29

Setting SIP INVITE Timeout for Individual DNs

With this enhancement, you can limit how long a SIP transaction will remain in Proceeding state if the only provisional response received was `100 Trying`. When this timeout expires, the call is either sent to the DN configured in the `no-response-dn` option, or released if that option is not configured.

The `sip-invite-timeout` option set at the Application level specifies the number of seconds SIP Server waits for a response to the `INVITE` message; if no response is received in that interval, the call times out. The maximum value of this option is `34` seconds. To extend the waiting period of time for SIP Server after the `100 Trying` is received before the call times out, configure the `sip-trying-timeout` option for individual DNs, which offers the maximum value of `256` seconds.

Feature Configuration

Table 21 describes how to enable alternate routing for unresponsive DNs.

Table 21: Configuring Default Routing for Unresponsive DNs

Objective	Key Procedures and Actions
1. Configure the SIP Server Application.	In the SIP Server `Application` > `Application Options` > `TServer` section, configure this option: • `sip-invite-timeout`—Specifies the number of seconds that SIP Server waits before an `INVITE` times out. After this timeout, if the unresponsive DN is configured for it, SIP Server sends the call to the alternate DN. **Note:** This option affects the timeout for all `INVITE` messages sent by SIP Server. Genesys recommends that you not change this option without considering all the scenarios it may affect.
2. Configure the DN.	Go to SIP Server `Switch` > `DNs` > individual DN > `TServer` section and configure the following option: • `no-response-dn`—Enter the DN where SIP Server will send the timed-out call. To set the SIP `INVITE` timeout for individual DNs, configure the `sip-trying-timeout` option.

Alternate Routing for Unresponsive URS/ORS

Enhanced support introduced in SIP Server 8.1.101.75

SIP Server supports delivering calls to an alternative location in situations in which the Universal Routing Server (URS) or Orchestration Server (ORS) becomes non-operational or unresponsive. If enabled, SIP Server sends the call to a specified alternate DN if URS/ORS fails or if the call waits too long on a Routing Point.

In multi-site deployments, calls can be routed by using `route` or `direct-uui` ISCC transaction types, or by using the ISCC Call Overflow mechanism. If `route` or `direct-uui` transaction types are used, Genesys recommends configuring inbound trunks with OOSP (Out Of Signaling Path) for efficient use of alternate routing. That way, a call is removed from SIP Server, minimizing its load.

In addition, with this enhancement:

- When multiple alternate destinations are configured, including those located on different switches, SIP Server load balances them in a round-robin manner.
- SIP Server prevents loops in the routing path by ignoring all destinations that were already tried, and rejects the call if none are available.
- SIP Server supports standard log event 52053 for an alternate routing indication.

Feature Configuration

Table 22 describes how to enable alternate routing for unresponsive URS/ORS.

Notes:
- Alternate routing with attached data is enabled when alternate destinations are configured in a Default DNs list of the Routing Point DN configuration. However, if you configure the alternate destination using the `default-dn` option (on either the Application or the DN level), the alternate destination will be taken from that `default-dn` option. The alternate destination configured in `alternate-route-profile` will be ignored and not used.
- The Default DNs list in the Routing Point configuration is also used by URS to route an interaction to the default destination. See the *Universal Routing Reference Manual* for more information.

Table 22: Configuring Default DN for Unresponsive URS/ORS

Objective	Key Procedures and Actions
Configure for all Routing Points.	In the SIP Server `Application` > `Application Options` > `TServer` section, configure these options: • `default-dn`—Enter the alternate DN to which SIP Server sends calls in case of URS failure/timeout. **Note:** Applies to all `Routing Point` DNs on the switch, unless configured otherwise at the DN-level. • `router-timeout`—Enter the max time (in seconds) that a call waits on a Routing Point before SIP Server sends the call to the `default-dn`. Multi-site deployments: • Use the Application-level option `alternate-route-profile` to define a valid Routing Point DN that contains a Default DNs list. SIP Server uses that list when it encounters a Routing Point with an empty Default DNs list. • Set the parameter `alternate-route-cof=<true, false>` to `true` to specify that alternate routing uses the ISCC Call Overflow feature.
Configure for individual Routing Point.	In the SIP Server `Switch` > `DNs` > individual `Routing Point` DN > `TServer` section, configure the following option: • `default-dn`—Enter the alternate DN to be used for URS failures/timeout on this Routing Point only.

Feature Limitations

- Alternate routing does not support default access codes.

- SIP Server does not trigger alternate routing when the `router-timeout` timer is in progress and a URS disconnects from SIP Server, or when SIP Server submits a `TUnregisterAddress` request from the last T-Library client registered on this Routing Point. SIP Server triggers alternate routing only when the `router-timeout` timer expires.

Alternate Routing for Calls to an External Destination

SIP Server supports routing inbound 1pcc calls to a specified default location in cases where the incoming `INVITE` request is addressed to an external destination. With this feature enabled, as SIP Server receives an `INVITE` from an external source, it checks all configured DNs and registered endpoints. If the Request URI includes a number that does not match any of the configured DNs or registered endpoints, SIP Server sends the call to a specified default (DN configured in the `default-route-point` option), even if the destination might match a configured gateway. This feature is used to prevent SIP Server from looping the call back to the same gateway on which the call came in.

Note: This feature applies to inbound 1pcc calls only (initiated by `INVITE` request). This feature does not apply to ISCC calls or to 3pcc calls initiated by T-Library requests.

Feature Configuration

Table 23 describes how to enable this feature.

Table 23: Configuring Default DN for External Destinations

Objective	Key Procedures and Actions
Configure the SIP Server Application.	In the SIP Server `Application` > `Application Options` tab > `TServer` section, configure the following option: • `default-route-point`—Enter the DN where SIP Server will route calls addressed to an external destination. For example, a `Routing Point` DN that applies a treatment, then rejects the call.

Application Failure Detection

Application Failure Detection is a Management Layer feature, where a particular application is configured for monitoring by the Local Control Agent (LCA), so that corrective action can be taken if and when the application becomes unresponsive (hangs up).

For SIP Server, you can configure failure detection for the application itself as well as for any of its individual threaded modules. For more information about multi-threading, see "Multi-Threaded Architecture" on page 49.

How Failure Detection Works

1. With failure detection enabled, the Management Layer monitors heartbeat messages (UDP packets) sent by the SIP Server application to the LCA.
2. If the LCA discovers that the application becomes unresponsive, it sends a notification to the Solution Control Server (SCS) stating that a hang-up has occurred, and what caused it. SCS then issues a log message:

   ```
   5160|STANDARD|GCTI_SCS_APP_HANG_UP_DETECTED| Application hang-up
   detected, reason %s
   ; Produced by SCS on behalf of application when LCA reports that
   application
   ; hang-up detected
   ; %s - reason of hang-up detection (application or thread hangup)
   ```

3. If `hangup-restart` is enabled on SIP Server, the LCA will restart SIP Server.

   OR

   If `hangup-restart` is set to `false`, monitoring continues. If the situation resolves itself, at the next successful heartbeat message SCS generates the following log message:

   ```
   5161|STANDARD|GCTI_SCS_APP_RESTORED_AFTER_HANG_UP| Application
   restored after hang-up
   ; Produced by SCS on behalf of application when LCA reports that
   application
   ; restores correct behavior after hang-up
   ```

Feature Configuration

Table 24 describes how to enable this feature.

Table 24: Configuring SIP Server for Hang-Up Detection

Objective	Key Procedures and Actions
Configure the SIP Server Application.	In the SIP Server `Application` object > `Options` tab > `sml` section, configure the following options: • `heartbeat-period`—Set this option to the length of time, in seconds, that Management Layer will wait before taking corrective action. • `hangup-restart`—Set this option to `true` to restart SIP Server in case it becomes unresponsive, `false` to send a notification only.

Feature Limitation

In 8.0.3, SIP Server supports application hang-up detection for the main thread class only.

Associating an ACD Queue with a Routing Point

SIP Server is able to associate an ACD Queue with a Routing Point by specifying the `Routing Point` DN in the `Association` field in the `Properties` dialog box of the `ACD Queue` DN object in the Configuration Layer.

The call flow for this functionality is as follows:

- Agents log into the ACD Queue.
- An inbound call arrives at the ACD Queue and at the associated Routing Point. The call is not auto-distributed to an agent in that ACD Queue.
- A Universal Routing Server (URS) strategy on the Routing Point selects an available agent in the ACD Queue.
- The call is routed to an agent's DN, which responds with a SIP Ringing message. As a result, an `EventDiverted` message is distributed against the ACD Queue and `EventRouteUsed` and `EventDiverted` messages are distributed against the Routing Point.
- The agent answers the call.

Notes: The inbound call will be treated as a regular call to the ACD Queue if no URS application has registered for the Routing Point associated with the ACD Queue.

The inbound call will be treated as a regular call to the ACD Queue if a URS application has registered for the Routing Point associated with the ACD Queue, but the routing timeout expires.

Automatic Inactive Agent Logout

SIP Server can automatically log an agent out after a specified period of inactivity, so as to ensure the accurate reporting of agent activity. Automatic agent logout can be configured for agents who are in a `NotReady` status, or more strictly for agents who are in either a `NotReady` or a `Ready` status in a work-related mode (for example, `AfterCallWork`).

Agent activity is determined by monitoring the following:

- Changes in the agent state.
- Calls that are made or received at the DN from which the agent is logged in (SIP Server will not log out an agent who is currently on a call).

Feature Configuration

Table 25 describes how to enable the automatic agent-logout feature. For highest priority, set the following options in the `Agent Login` object.

Table 25: Enabling Auto Agent Logout

Objective	Related Procedures and Actions
1. Enable automatic agent logout.	In the `TServer` section of the applicable configuration object, set the `auto-logout-timeout` option to a value of 1 or greater.
2. (Optional) Enable a stricter logout policy.	In the `TServer` section of the applicable configuration object, set the `auto-logout-ready` option to `true`.

Call Completion Features

SIP Server supports Call Completion on Busy Subscriber (CCBS) and Call Completion on No Reply (CCNR) when offered by the Siemens OpenScape Voice switch. This feature provides a callback mechanism, where a caller is able to request a call back from the switch when a line they have tried to reach (but is busy or does not answer) later becomes available.

How It Works

For this feature to work, both the caller DN and the destination DN must be behind the same switch.

Call Completion on Busy Subscriber/No Answer

A sample call flow for a CCBS scenario is as follows:

1. DN1 and DN2 are both behind the switch that provides the CCBS/CCNR feature.
2. DN1 places a call to DN2, but DN2 is either busy or there is no answer. DN2 includes the `Allow-Events` header in its SIP response, requesting the feature:
   - `486 Busy Here` includes `Allow-Events: CCBS`
   - `180 Ringing` includes `Allow-Events: CCNR`
3. SIP Server passes the `Allow-Events` to the switch, which then presents to DN1 the option to start the callback feature, as per switch functionality. DN1 accepts the callback.
4. To establish the subscription between the DNs, DN1 sends a `SUBSCRIBE` message to DN2 through SIP Server:

   ```
   SUBSCRIBE
   Event: CCBS;queue=tru;service-retention=service-retained
   Contact: URI
   ```

5. DN2 responds with a `NOTIFY` message confirming the subscription:

   ```
   NOTIFY
   Event: CCBS;queue=tru;service-retention=service-retained
   Subscription-state: active
   Contact: URI
   ```

6. With the subscription established, the original call ends.
7. When DN2 becomes available, it sends a `NOTIFY` (user-free) message as per the subscription. On receiving this `NOTIFY`, the switch sends an `INVITE` to DN1. If DN1 answers, the switch then sends an `INVITE` to DN2 (now free) and a call between the two parties is established.

Feature Configuration

SIP Server does not provide this functionality itself, but instead supports this functionality when offered by Siemens OpenScape Voice version 6.0.

Table 26 describes how to enable the call completion features.

Table 26: Configuring Call Completion

Objective	Related Procedures and Actions
Configure a SIP Server Application.	In the `TServer` section of the SIP Server `Application` object, configure the following options: 1. `internal-registrar-enabled`—Set this to `false`. 2. `external-registrar`—Set this to the same value as the `contact` option configured on the softswitch DN (`Voice over IP Service` DN with `service-type` set to `softswitch`). With this configuration, SIP Server processes the `Allow-Events: CCBS` and `Allow-Events: CCNR` headers if they are included in the `INVITE` request.

Feature Limitation

This functionality is available for 1pcc calls only (not applicable for 3pcc calls).

Call Divert Destination

SIP Server supports routing the caller to a specific destination when, after an initial leg of the call is completed, only the caller remains on the line. For example, this feature could be used to route the caller to a post-call survey.

Feature Configuration

To enable this feature, configure the DN-level option `after-call-divert-destination` on the Routing Point DN. You can also enable this feature by passing the `after-call-divert-destination` parameter in the `Extensions` attribute of a `TRouteCall` request. Parameters passed in the `Extensions` attribute override the value of the configured option.

Feature Limitations

- This feature is supported only in single-site deployments.
- This feature is supported only for the calls initiated by a `TMakePredictiveCall` request on behalf of a Routing Point. In all other cases, calls initiated by `TMakePredictiveCall` requests are not supported.

Caller Information Delivery Content for AT&T Trunks

Introduced in SIP Server 8.1.101.66

Support for GVP added in 8.1.102.00

SIP Server can pass the multipart body content received in `INVITE` messages (as described in RFC 5621) to make it available to URS/ORS and/or GVP. The only content type currently supported is Caller Information Delivery (CID), as defined in the AT&T specification for AT&T IP Toll Free Service SIP trunks.

SIP Server communicates with URS/ORS using T-Library to pass the CID content that it receives in a multipart `INVITE` body, as an attribute of an `EventRouteRequest` message. See "Passing CID Content to T-Library Clients (URS/ORS)" on page 117.

SIP Server communicates with GVP using SIP to pass the CID content that it receives in a multipart `INVITE` body in the relayed `INVITE`. See "Passing CID Content to SIP Destinations (GVP)" on page 119.

Notes:
- CID content that is received in a multipart `INVITE` body is still delivered following an MCP failure or a SIP Server failure in HA hot-standby mode.
- CID content is handled in Presence Information Data Format (PIDF), as RFC 3863 describes in detail.

Enabling CID Content Retrieval

Configure the DN-level option `sip-accept-body` to enable SIP Server to retrieve CID content from the `INVITE` that it receives from a Trunk DN.

Passing CID Content to T-Library Clients (URS/ORS)

SIP Server previously mapped the SDP portion of a SIP message body to a T-Library event attribute; see the section "Mapping SIP Headers and SDP Messages" on page 261. Now it can also perform CID mapping to T-Library clients (URS/ORS). SIP Server sends `EventRouteRequest` with the CID content passed in `AttributeExtensions`.

To enable CID mapping to T-Library clients, add this configuration option to the `INVITE` section:

- `extensions-1 = CID`

By default, CID content is passed to T-Library clients unchanged (UTF-8 encoding). If conversion to a local charset is enabled for SIP-to-TLib mapping

(set with the encoding option), then this conversion is also applied to CID content.

Note: CID can be mapped to `AttributeExtensions` only. CID mapping to `AttributeUserData` is not supported.

Example

```
message EventRouteRequest
 AttributeThisDN '5001'
 AttributeThisDNRole 2
 AttributeThisQueue '5001'
 AttributeOtherDN '31001'
 AttributeOtherDNRole 1
 AttributeConnID 2266025dfcd2c001
 AttributeExtensions
       'CID'
       'Content-Type: application/pidf+xml
       <presence xmlns="urn:ietf:params:xml:ns:pidf"
       xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
       xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
       xmlns:gml="http://www.opengis.net/gml"
       xmlns:gs="http://www.opengis.net/pidflo/1.0"
       xmlns:cl="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"
       xmlns:tf="http://www.att.com/iptf"
       entity="pres:tfas1@att.net">
       <tf:dataresponse status="available"/>
       <dm:device id="3754348893">
           <gp:geopriv>
               <gp:location-info>
                   <gs:Circle srsName="urn:ogc:def:crs:EPSG::4326">
                       <gml:pos>40.3958 -74.1322</gml:pos>
                       <gs:radius
uom="urn:ogc:def:uom:EPSG::9001">113</gs:radius>
                   </gs:Circle>
                   <cl:civicAddress>
                       <cl:A1>Daly City</cl:A1>
                       <cl:A3>CA</cl:A3>
                       <cl:PC>94014</cl:PC>
                       <tf:streetaddress>2001 Junipero
Serra</tf:streetaddress>
                       <tf:name>Genesys</tf:name>
                       <tf:givenName></tf:givenName>
<tf:mailableVerified>true</tf:mailableVerified>
                       <tf:listType>Bus</tf:listType>
                   </cl:civicAddress>
               </gp:location-info>
           </gp:geopriv>
       </dm:device>
       </presence>'
```

Passing CID Content to SIP Destinations (GVP)

Configure the DN-level option `sip-pass-body` to specify that the CID content (taken from one of the call parties) is included in the initial `INVITE` that is sent to the DN.

Configure the Application-level option `cid-enable-on-vtp` to simplify provisioning of the IVR configured through the Voice Treatment Port (VTP) DNs. Set to `true` to specify that CID content is passed to the VTP DN in the initial `INVITE`.

Call Park/Retrieve

This feature lets SIP Server support Call Park and Call Retrieve features provided by various Private Branch Exchange (PBX) vendors. This feature lets users (agents) park a call for a period of time—for example, to change phones—and then retrieve the call later.

How It Works

1. A call is established between an agent and a caller.When the agent wants to park the call, he or she initiates a transfer using a 1pcc request to a specially-configured "Call Park" star code—for example, `*10`.
2. SIP Server parks the call on the internal `gcti::pbxpark` device (the device does not need to be created in the Configuration Layer because it is an internal SIP Server device). While the call is parked, the agent can hang up their phone if the agent needs to; the caller remains parked and not disconnected from the contact center.
3. When the agent wants to retrieve the call, he or she dials a specially-configured "Retrieve Call" star code, plus the number of the DN from which the call was parked at the internal `gcti::pbxpark` device. For example, `*11 1001`, where `*11` is the star code, and `1001` is the DN from which the call was parked.
4. Based on the provided DN, SIP Server retrieves the parked call from `gcti::pbxpark` and re-connects the caller with the agent.
5. If, while parking the call, the agent enters the wrong "Call Park" star code, the caller will be placed on hold. If, while unparking the call, the agent enters the wrong "Call Park" star code, SIP Server identifies that there is no associated parked call, checks for other star code features, and applies standard call processing for unknown dialed numbers if no star code features are found.
6. If a call remains parked for longer than the configured `max-parking-time` option, SIP Server returns the call to the original DN.

Feature Configuration

Table 27 describes how to configure Call Park/Retrieve.

Table 27: Configuring Call Park/Retrieve

Objective	Related Procedures and Actions
Configure the SIP Server `Application`.	In the SIP Server `Application > Application Options > TServer` section, configure the following options: • `feature-code-park`—Enter the number part of the star code to be used to park the call. • `feature-code-retrieve`—Enter the number part of the star code to be used to retrieve the parked call. • `max-parking-time`—Set the timeout after which a parked call will be reconnected with the initial DN. • `music-on-pbxpark`—Enter the name and path to a valid audio file for the audio that will be played to remote parties connected to the SIP Server internal `gcti::pbxpark` device.

Feature Limitations

- Call Park and Call Retrieve functionality is only supported in single-site deployments.
- This functionality only works with 1pcc call flows. SIP Server does not support 3pcc requests (`TSingleStepTransfer, TMakeCall`) to star codes.

Call Pickup

SIP Server supports the Call Pickup feature. When enabled, calls ringing at an agent device may be picked up by another agent from his or her device by dialing a following combination:

`*<pickup code><DN where a call is ringing>`

For example, if the pickup code is `12` and a call is ringing at DN `1001`, the other agent can dial `*12 1001` from his or her current device to pick up the ringing call at DN `1001`.

Feature Configuration

Table 28 describes how to configure Call Pickup.

Table 28: Configuring Call Pickup

Objective	Key Procedures and Actions
1. Configure the SIP Server Application.	In the SIP Server `Application` > `Application Options` > `TServer` section, configure the following option: • `feature-code-pickup`—Set this option to the numeric string that will be used as the pickup code. By default, it is set to `12`.
2. Configure a DN.	To enable call pickup on a particular DN, go to `Switch` > `DN` object, `Options` > `TServer` section and configure the following option: • `enable-direct-pickup`—Set this option to `true`.

Feature Limitation

This functionality only works with 1pcc call flows. SIP Server does not support 3pcc requests (`TSingleStepTransfer, TMakeCall`) to star codes.

Call Recording—NETANN-Based

SIP Server supports call recording using two different methods:

- MSML-based call recording—SIP Server invokes Genesys Media Server to record calls using Media Server Markup Language (MSML), as part of an overall recording solution with one of the following:
  - Genesys Interaction Recording
  - Genesys Quality Management
  - A third-party voice recorder (requires an appropriate Genesys Connector license for recording)

  For details about this kind of recording, see "Call Recording—MSML-Based" on page 125.
- NETANN-based call recording—SIP Server invokes Genesys Media Server (MCP) to record calls to a local file using the NETANN protocol.

SIP Server supports both regular call recording and emergency call recording.

Regular Call Recording

Call recording is performed by passing a Real-Time Transport Protocol (RTP) stream through a media server (Genesys Media Server (MCP)). This Media Server acts as a proxy for the media stream, recording all media packets into a file. Depending on the configuration, the media server may perform media mixing, or it may save the RTP packets as is, thus improving call recording

performance. (See the *Genesys Media Server Deployment Guide* for details.) Call recording is always enabled on a single call leg, such as with a gateway or a SIP phone.

Call recording starts after a call becomes established. It does not result in any changes to the call itself, to event processing, or to any other generated TEvent.

When call recording starts, SIP Server creates two new SIP dialogs with the media server, SIP Server sends re-`INVITE` requests to all call participants with the SDP from the media server. As a result, the RTP stream between call participants is passed via the media server.

Call recording has the highest priority compared to other operations that can be performed on the call when it is established. That is, when the `EventEstablished` message is generated on the destination DN, the operations on the call are performed in the following order:

1. Call recording, if enabled.
2. Personal greeting, if enabled.
3. Supervisor monitoring, if enabled

Recording is only available for calls with audio media. If a call contains video or IM media, recording will not start.

Only one recording is allowed on a call. If the configuration enables recording on more than one device in the call, recording will only start on the device for which the first `EventEstablished` is distributed.

Consultation Calls

Regular call recording is also available for consultation calls. If re-enabled for it, SIP Server starts call recording when any DN that is involved in the consultation call is set for recording. Once recording is initiated, it continues for as long as at least one party that is set for recording remains in the call. Recording ends when no more recording-enabled parties are left. For example, if recording for both main and consultation calls is initiated by a single party, and that party then initiates a `TCompleteTransfer`, recording on both calls is terminated once the transfer is completed.

Reassigning Recording

In cases where recording is initiated by a party that then leaves the call, while another party remaining on the call is also configured for call recording, SIP Server will continue recording the call for the same file. The actual RTP stream is reassigned from the original initiating party to any other party on the call that is configured for call recording. If cases where no remaining party is configured for call recording when the initiating party leaves the call, the recording is terminated.

Building the Request URI for the Recording

SIP Server builds the Request URI for the call recording in a number of ways, depending on configuration and type of call recording:

- For regular call recording, SIP Server builds the Request URI automatically at run-time, so long as the `recording-filename` option on the SIP Server application is correctly configured. SIP Server does not use the value of the `request-uri` option on the `recording-service` DN, except as a backup in case the `recording-filename` is wrong or not configured.
- For emergency call recording, SIP Server builds the Request URI using both the value of the `request-uri` option on the Application (if configured), as well as the value of the `emergency-recording-filename` option. If `request-uri` is not configured, then the resulting URI is formed as follows:

  `<RECORDER-DN>@<SIP-ADDRESS>:<SIP-PORT>`

  For example,

  `REC@192.168.1.2:5060`

Feature Configuration

Table 29 describes how to enable NETANN-based call recording.

Table 29: Configuring NETANN-Based Call Recording

Objective	Related Procedures and Actions
1. Configure a DN.	To enable call recording on a particular DN, in the `TServer` section of the `DN` object, set the configuration option `record` to `true`. To record all inbound calls coming for a particular media gateway, set the `record` option to `true` on the `Trunk` DN that represents this gateway.
2. Configure a SIP Server Application.	In the SIP Server `Application` > `Application Options` > `TServer` section configure the following options: 1. `recording-filename`—Enter the name of the recorded file. For example: `call-$ANI$-$DNIS$-$DATE$-$TIME$-$CONNID$-$UUID$-$AGENTDN$-$AGENTID$` 2. `record-consult-calls`—To enable recording for consultation calls, set this option to `true`.
3. Configure an `Extensions` attribute.	Specify an `Extensions` attribute with a `record` key in the `TRouteCall` request. See the key values in Table 108, "Use of the Extensions Attribute," on page 414. The routing strategy will determine whether call recording is needed.
4. Configure the Media Server application.	Configure and tune the Genesys Media Sever (MCP) application according to the *Genesys Media Server Deployment Guide*.

Table 29: Configuring NETANN-Based Call Recording (Continued)

Objective	Related Procedures and Actions
5. Configure a recording service.	Configure a DN of type `Voice over IP Service` with the following configuration options: • `contact`—Set this option to the device's IP address and port used for recording. **Note:** The device can be MCP. • `request-uri`—Set this option to the SIP URI. • `service-type`—Set this option to `recorder`. See "Configuring a Recording Service" on page 89 for details.

Emergency (Manual) Call Recording

SIP Server performs emergency call recording when processing a single-step conference call request that specifies `AttributeOtherDN` as a `Trunk` DN specifying a `gcti::record` number. When this attribute is set, SIP Server recognizes this special request and initiates call recording as follows:

- Selects one of the available call recording units that are configured in the Configuration Layer. See "Configuring a Recording Service" on page 89 for more information.
- Performs a single-step conference call and adds the selected call recording unit to the call.
- Creates the file name as configured in the `emergency-recording-filename` option that is described on page 460.

To stop emergency call recording, the agent must issue the `TDeleteFromConference` request using the `gcti::record` number.

Feature Limitations

Emergency call recording cannot be activated on a consultation call if it has already been activated from the same DN on the primary call. Emergency call recording can only be activated on both primary and consultation calls if initiated from different DNs.

Call Recording—MSML-Based

SIP Server supports call recording using two different methods:

- MSML-based call recording—SIP Server invokes Genesys Media Server to record calls using Media Server Markup Language (MSML), as part of an overall recording solution with one of the following:
  - Genesys Interaction Recording
  - Genesys Quality Management
  - A third-party voice recorder (requires an appropriate Genesys Connector license for recording)
- NETANN-based call recording—SIP Server invokes Genesys Media Server to record calls to a local file, using the NETANN protocol. For details, see "Call Recording—NETANN-Based" on page 121.

About Genesys Media Server

The Genesys Media Server is a module that provides MSML-based media services offered by the Genesys Voice Platform. When integrated with SIP Server, it supports MSML-based call recording, where the Genesys Media Server acts as a proxy, replicating the media stream in a new recording session with a third-party voice recorder that does the actual recording. In case of file-based call recording, the actual recording is processed by MCP.

Note: For more information about integrating SIP Server with Genesys Media Server, see "Configuring Genesys Media Server" on page 92.

How Call Recording Works

Depending on how call recording is configured, the basic call flow for it is as follows:

1. Call recording is initiated in one of the following ways:
   - Static configuration—Recording is enabled through static DN-level configuration on either the customer side (`Trunk` DN) or on the agent side (`Extension` DN or `Agent Login`).
   - Routing strategy—The routing strategy initiates recording through the `TRouteCall` request that it sends to SIP Server.
   - T-Library client or 3rd-party recorder—A T-Library client initiates recording through a `TPrivateService` request that it sends to SIP Server.
2. Based on this trigger, SIP Server builds a request URI that includes key recording-related parameters. It then sends this request URI in an `INVITE` to Resource Manager.

3. Resource Manager determines the right MCP to provide the service, and forwards the `INVITE` to the selected MCP to set up the service.
4. SIP Server sends additional MSML instructions in SIP `INFO` messages, telling the media server to start the recording.
5. For additional control over the established recording session, T-Library `TPrivateService` requests can be used to initiate new actions—for example, to pause or resume recording. SIP Server forwards the resulting MSML instructions in new `INFO` messages.

Supported Media File Format

MSML-based call recording supports the `wav` and `MP3` file formats (NETANN-based recording supports both `wav` and `pcap`).

Building the Request URI for the Recording

SIP Server builds the Request URI for call recording in a number of ways, depending on configuration and type of call recording:

```
sip:msml=<conf-id>@<resource-managaer>;<dn>=<DN>;record
```

where,

`msml`	Fixed part of the URI. Identifies the protocol as MSML.
`conf-id`	Unique identifier for the MSML/recording session. Ensures that all users are connected to the correct media server.
`DN`	The DN of the endpoint that SIP Server will record.
`record`	Identifies “record” as the type of MSML session. Genesys Media Server can then properly handle recording separately from other MSML services.

Dynamic Call Recording

Call recording can be started on an as-needed or “emergency” basis during an ongoing call. To initiate dynamic recording, recording-related parameters are included in the `Extensions` attribute in either of the following T-Library messages:

- `TRouteCall`
- `TPrivateService`

TRouteCall

The URS routing strategy must be configured to include recording-related parameters in the `TRouteCall` request that it sends to SIP Server.

The Extensions attribute must include the key record, with one of the following values:

- source—The recording will be initiated on the DN that sent the call to the Routing Point (customer) and will continue as long as the customer stays in the call.
- destination—The recording will be initiated on the routing destination DN (agent) and will continue as long as the agent stays in the call.

TPrivateService

The T-Library client or 3rd-party recorder must include recording-related parameters in the TPrivateService request that it sends to SIP Server.

To initiate dynamic recording with TPrivateService, the request uses the following parameters:

Table 30: Dynamic Call Recording Extensions in TPrivateService

Attribute	Value
PrivateMsgID	Specifies the type of recording operation to be performed: • GSIP_RECORD_START (3013)—Starts the recording
ThisDN	Specifies the DN on behalf of which recording operation is requested. This DN must be registered by the T-Library client.
ConnectionID	References the ID for the call that is currently being recorded.
Extensions	Specifies key-value pairs used to control the recording session: • record—Set to source or destination. • id—Adds a recording identifier to the recording session. This identifier must be globally unique; it is passed back in the recording session. If this parameter is not included in the request, SIP Server will construct a unique identifier based on the recording-filename option. • dest—Overrides the default location of the 3rd party recording server. • params—Adds additional parameters that are passed as general key-value pairs in the request. There parameters will appear in the recording session. For example, AttributeExtensions... 'record' 'source' 'id' '32980asdf320990ad' 'dest' 'sip:172.24.129.75:5070' 'name1' 'value1' 'name2' 'value2'
Reasons	Specifies any reasons. Processed the same as for all other T-Library requests.

Mid-Call Control of the Recording Session

Using `TPrivateService` requests, T-Library clients can control in real-time an ongoing recording session. The client can pause, resume, or stop the recording. SIP Server translates recording-related parameters from the request to `INFO` messages that it sends to Genesys Media Server.

Supported mid-call actions are as follows:

- Stop the recording.
- Pause the recording.
- Resume a paused recording.

To control mid-call recording, the `TPrivateService` request uses parameters described in Table 31.

Table 31: Mid-Call Recording Extensions in TPrivateService

Attribute	Value
`PrivateMsgID`	Specifies the type of recording operation to be performed: • `GSIP_RECORD_STOP (3014)`—Stops the recording • `GSIP_RECORD_PAUSE (3015)`—Pauses the recording. • `GSIP_RECORD_RESUME (3016)`—Resumes the recording
`ThisDN`	Specifies the DN on behalf of which recording operation is requested. This DN must be registered by the T-Library client.
`ConnectionID`	References the ID for the call that is currently being recorded.
`Reasons`	Specifies any reasons. Processed the same as for all other T-Library requests.

Recording During Transfers and Conferences

SIP Server supports continuous recording for conference calls if the party (where the record was initiated) is dropped from the conference, and any party remaining on the call requests the recording (by DN configuration, routing strategy, or T-Library client). Recording ends when no more recording-enabled parties remain. Recording can be stopped by a respective T-Library request.

Feature Limitation

Continuous recording applies only to two-step transfers.

Recording Calls Without Music-on-Hold Treatment

SIP Server provides the ability to record a call without recording a music-on-hold treatment when a call is placed on hold. SIP Server sends corresponding

MSML information in `INFO` messages to Genesys Media Server to pause the recording (`gvp:recorder state="pause"`) when the call is placed on hold and to resume the recording (`gvp:recorder state="start"`) when the call is retrieved.

This functionality also applies to call transfers: the recording is paused when a transfer is initiated, and resumed when the transfer is completed.

When several agents are involved in a call and the call is placed on hold, SIP Server pauses the recording at the first invocation of the hold operation and resumes the recording at the first invocation of the retrieve call operation.

If an agent pauses the call recording and then places the call on hold, SIP Server resumes the recording when the call is retrieved from hold.

If SIP Server receives an error message from Genesys Media Server in response to pause or resume the recording, it will not resubmit the request. Recording will be left in the previous state.

In multi-site deployments (see Figure 7), where the recording and music-on-hold treatment might happen on different SIP Servers, the SIP Servers will use an `EventNetworkPrivateInfo` message containing the `AttributePrivateMsgID` to pass the recording control from one SIP Server to another SIP Server. The value of the `AttributePrivateMsgID` indicates the recording state:

- `6004`—Active recording is paused
- `6005`—Active recording is resumed
- `6006`—Active recording is stopped

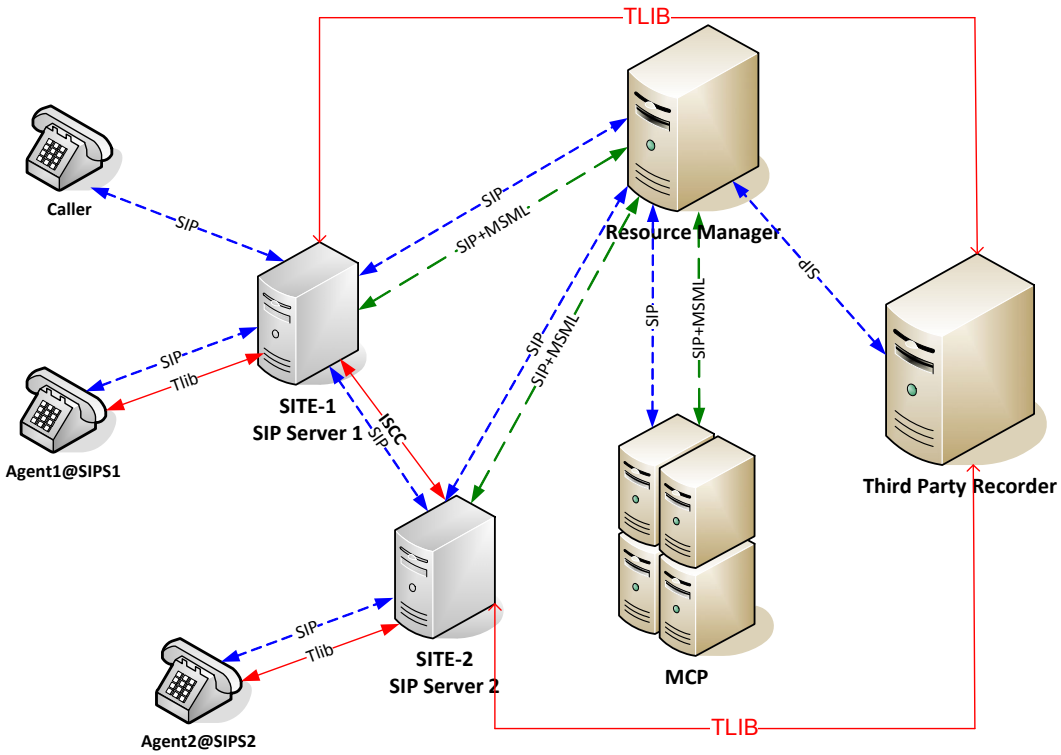

Figure 7: SIP Server Multi-Site Deployment with Dynamic Recording Capabilities

When dynamic pause of recording (explicit pause) and hold call (implicit pause) are invoked at the same time, recording is paused only once. This is done to remove the repeated pause INFO message to Genesys Media Server. In this case, recording is resumed when either recording is resumed or call is retrieved whichever is performed first.

Feature Configuration

The `record-moh` configuration option enables this feature. See "Configuring MSML-Based Call Recording" on page 134.

Feature Limitations

- This feature is supported only for MSML-based call recording.
- This feature is supported in multi-site deployments (ISCC calls) only when `event-propagation` is set to `list` in the `extrouter` section of the SIP Server application.
- Recording can be paused when parties on the call are still speaking. It can happen in the following scenario:
  a. Caller (at SIP Server 1), Agent1 (at SIP Server 1), and Agent2 (at SIP Server 2) are talking.
  b. Recording sessions are activated on the caller's leg and on the Agent2 leg.
  c. Agent1 or Agent2 places the call on hold.
  d. Recording session is paused on both SIP Servers.
- Recording status can be reported inaccurately if the global command fails on one of the recording SIP Servers. For example, the `PAUSE` command is submitted. It is executed successfully on SIP Server 1 and fails on SIP Server 2. An agent connected to SIP Server 2 may see the recording status as "`PAUSED`" even though recording is still in progress on SIP Server 1.

Video Call Recording

For video calls, recording applies to the audio portion of the call only. The video part of the dialog remains unaffected by the audio recording process.

Call Recording Alarms

SIP Server supports a standard log event, `01-52051`, for unsuccessful call recording scenarios. A recording scenario is considered unsuccessful if one of the recording operations fails and cannot be recovered by SIP Server. For example, SIP Server tries to start recording on a DN and fails. SIP Server then makes a second attempt to start this recording, using a different MCP. If the second attempt is successful, the alarm is not generated. If the second attempt

also fails and recording is not started, then the recording scenario is considered unsuccessful and the alarm is raised.

An alarm can be issued for the start, pause, and resumption of MSML-based recording operations regardless of the method selected as a trigger. For example, an alarm can be triggered for the start recording command if it is submitted from an agent desktop or from the routing strategy in a `TRouteCall` request, or if it is activated internally based on a DN configuration. An alarm is not raised for the stop recording operation. If an attempt to stop the recording fails, SIP Server terminates recording dialogs without raising an alarm.

With this feature enabled, when the first call recording failure is detected, SIP Server generates a `01-52051` alarm message and starts the timer using the interval defined in the `recording-failure-alarm-timeout` configuration option. Each consecutive call recording failure detected during this period increments the counter. When the timer expires, SIP Server generates an alarm message with the number of failures detected in the past interval and resets the counter. If the timer expires and no recording failures have been detected within the past interval, SIP Server does not generate an alarm message.

This call recording alarm is designed as a persistent alarm. An administrator can clear this alarm manually or use the Clearance Timeout timer in GAX.

Here are some examples of alarm messages:

```
14:34:38.906 Std 52051 1 call recording sessions failed
14:35:38.913 Std 52051 571 call recording sessions failed
```

Call Recording Failure Count

SIP Server maintains the call recording failure count by using the SIP Server 1536 logs.

When SIP Server starts, a log with suffix `1536` is generated where SIP Server operational statistics are periodically logged. Operational statistics are generated properly only if the `x-sip-log` option is configured. The frequency with which statistics getting printed in the log is controlled by the `operational-stat-timeout` configuration option.

The cumulative number of call recording failures is reset after SIP Server restarts.

A SIP Server active call recording failure count is printed in the log of the Call Manager section. Here is an example of a message containing the call recording failure count that is logged in the SIP Server 1536 log.

```
10:30:29.945: --------- SIP Server Operational Statistics ------------
.
.10:30:29.945: <sipCallManager>
..
10:30:29.945: NCALLRECORDINGFAILED=10
10:30:29.945: </sipCallManager>
..
10:30:29.945: ------------ End of Operational Statistics ------------
```

Recording in Outbound Call Scenarios

In Outbound call scenarios (ASM, Transfer, or Proactive modes), recording will be started only after a caller is connected to an agent in both regular (static) and dynamic recording.

DN Recording Override

Call recording functionality can be enabled statically on a DN by setting the `record` configuration option to `true`, or dynamically by using the `record` key in the `Extensions` attribute of a `TRouteCall` request.

With this feature, call recording can be selectively disabled through a routing strategy by overriding the `record` option configured on a DN. Call recording can be disabled on either the origination DN or destination DN when a routing strategy issues `TRouteCall` containing the `record` extension key set to `disable_source` or `disable_destination`, respectively.

When recording is disabled by the `TRouteCall` request, recording can be started on the DN by issuing a `TPrivateService` request after the call is established.

DN Recording Override is supported with MSML-based call recording, for single-site, multi-site, and Business Continuity deployments. DN Recording Override is not supported with NETANN-based call recording.

General Rules for DN Recording Override

- If a recording configuration is overwritten for a DN, recording does not start when a call is answered on this DN. Recording can still be activated on this DN when the call is already established using the `TPrivateService(GSIP_RECORD_START)` request.
- It is not possible to disable recording on both origination and destination DNs using the same `TRouteCall` request.
- Extension key values provided in a `TRouteCall` request are not carried forward to the subsequent requests.
- Call recording that is already in progress cannot be stopped.

Table 32 describes how the `record` key is processed if a call is routed to the remote SIP Server.

Table 32: Record Key Processing in Multi-Site Scenarios

Record Extension Key Value in TRouteCall	ISCC Transaction Type route
`destination`	Recording is started at the remote site
`disable_destination`	Recording configuration is overridden at the remote site

Table 32: Record Key Processing in Multi-Site Scenarios (Continued)

Record Extension Key Value in TRouteCall	ISCC Transaction Type route
source	Recording is started at the local site
disable_source	Recording configuration is overridden at the local site

Note: For multi-site REFER (OOSP) transfers, only the record key extension values destination and disable_destination with the ISCC transaction type Route are supported.

Multi-Site Call Flow Examples

These call flow examples show how DN Recording Override works in multi-site deployments.

Example 1: record='disable_source'

1. Agent 1 with record=true at Site 1 dials internally to a Routing Point at Site 1.
2. TRouteCall containing record='disable_source' with ISCC transaction type route is issued to Agent 2 at Site 2.
3. Call recording is disabled for Agent 1 at the origination site (Site 1).

Example 2: record='disable_destination'

1. An inbound call arrives at a Routing Point at Site 1.
2. TRouteCall containing record='disable_destination' with ISCC transaction type route is issued to Agent 2 with record=true at Site 2.
3. Call recording is disabled for Agent 2 at the destination site (Site 2).

Configuration Notes

This feature applies only if the following configurations are enabled:

- Application-level options must be set to true:
  - msml-support=true
  - msml-record-support=true
- Multi-site deployment:
  - The destination site must be controlled by SIP Server (sip-server-inter-trunk=true).
  - ISCC transaction type must be set to route.

Feature Configuration

Table 33 describes how to enable MSML-based call recording.

Table 33: Configuring MSML-Based Call Recording

Objective	Key Procedures and Actions
1. Integrate SIP Server with Genesys Media Server.	Complete the steps described in: • Table 14: Integrating Media Server for MSML, on page 94
2. Configure the SIP Server Application.	In the `TServer` section of the SIP Server `Application` object, configure the following options: • `resource-management-by-RM`—Set this option to `true`. • `recording-filename`—Enter the name of the recorded file. For example: `call-$ANI$-$DNIS$-$DATE$-$TIME$-$CONNID$-$UUID$-$AGENTDN$-$AGENTID$` • `msml-support`—Set this option to `true`. • `msml-record-support`—Set this option to `true`. *Required for backward compatibility for local recording with GVP.* • `record-consult-calls`—To enable recording for consultation calls, set this option to `true`. • `record-moh`—To enable recording without a music-on-hold treatment, set this option to `false` (in case if music-on-hold is enabled with the `sip-enable-moh` set to `true`). • `recording-failure-alarm-timeout`—To enable call recording alarms, set this option to a value other than 0 (zero). • `record-after-merge`—To enable call recording after a transfer or conference is completed, set this option to `true`.
3. Configure the MSML DN.	1. Open the `Voice over IP Service` DN that you created in the SIP Server-Genesys Media Server integration. 2. In the `TServer` section, configure the following additional options: ◆ `refer-enabled`—Set this option to `false`. ◆ `make-call-rfc3725-flow`—Set this option to `1`. ◆ `ring-tone-on-make-call`—Set this option to `false`.
Enabling Call Recording Features	
Enable full-time call recording.	To start recording based on static DN-level settings, set the `record` configuration option to `true`, in the `TServer` section, in any of the following: • `Extension` DN for agent-side recording • `Agent Login` for agent-side recording • `Trunk` DN for customer-side recording

Table 33: Configuring MSML-Based Call Recording (Continued)

Objective	Key Procedures and Actions
Enable dynamic call recording.	To start recording during an ongoing conversation, configure either of the following: • In the routing strategy, configure the `TRouteCall` request to include the key `record`, with the values: ◆ `destination` for agent-side recording ◆ `source` for customer-side recording • In the T-Library client, configure the `TPrivateService` request to include the key `record`, with the values: ◆ `source` for recording `ThisDN` ◆ `destination` for recording `OtherDN` You can also add the following optional key-value pairs: ◆ `id`—A string used to add an identifier to the recording session. Must be globally unique. If not configured, Media Server constructs a unique identifier itself. ◆ `dest`—A string used to override the default location of the 3rd party recording server. ◆ `params`—A string used to add additional parameters that can be passed as generic key-value pairs. These parameters will appear in the recording session. **Note**: Full-time recording takes precedence over dynamic recording. SIP Server rejects any dynamic recording request that arrive while recording is already underway.
Enable mid-call recording control.	To control the recording during an established session, configure `TPrivateService` to include the key `AttrPrivateMsgID`, using one of the following values: • `GSIP_RECORD_STOP (3014)` • `GSIP_RECORD_PAUSE (3015)` • `GSIP_RECORD_RESUME (3016)`
Enable DN recording override.	To disable call recording, in the routing strategy, configure the `TRouteCall` request to include the `record` key with the appropriate value, as follows: • `disable_source`—to disable recording on the origination DN. • `disable_destination`—to disable recording on the destination DN.

Call Recording—Geo-location

In Active Call Recording scenarios, SIP Server is able to select the Media Server and Recording Server based on its geographic proximity to either the caller or the agent. This minimizes WAN traffic and telecom costs. SIP Server does not select the service itself, it passes the geo-location information (in the `X-Genesys-geo-location` header) in the initial `INVITE` messages to Resource Manager, which then uses that information to select the closest Media Server to the caller or agent.

You can configure geo-location in any of the following places:

- Inbound `Trunk` DN
- `Routing Point` DN
- `AttributeExtensions` in `TRouteCall`
- Agent `Extension` DN

SIP Server selects and passes the `X-Genesys-geo-location` header using a different order of configuration precedence, depending on the call scenario.

Inbound Call Scenarios

Configuration Order of Precedence

For inbound calls, the order of precedence for the geo-location configuration is:

1. `AttributeExtensions` in `TRouteCall`
2. `Routing Point` DN where the incoming call arrives
3. Inbound `Trunk` DN where the call first arrives
4. Agent DN where recording is enabled

Outbound Call Scenarios

Outbound Solution—ASM, Proactive, or Transfer Modes

For outbound calls using TMakePredictiveCall (ASM, proactive, or transfer mode), the order of precedence for the geo-location configuration is:

1. `AttributeExtensions` in `TRouteCall` that is received from URS in response to SIP Server's `TRouteCall`, which is issued upon an answered outbound call arrival.
2. `Routing Point` DN from which an outbound call is originated or where the answered outbound call is transferred to from a `Trunk Group` DN for distribution to an agent.

3. Outbound `Trunk` DN where the call is sent to the customer.
4. Agent DN where recording is enabled.

Outbound Solution—Engaging Mode

For outbound calls that engage the agent before making the call out to the customer, the order of precedence for the geo-location configuration is:

1. `AttributeExtensions` in `TRouteCall` that is received from URS in response to SIP Server's `TRouteCall`, which is issued upon an answered outbound call arrival.
2. `Routing Point` DN from which an outbound call is originated or where the answered outbound call is transferred to from a `Trunk Group` DN for distribution to an agent.
3. Outbound `Trunk` DN where the call is sent to the customer.
4. `Trunk Group` DN for Engaging mode.

Regular Outbound Calls

For outgoing calls using `TMakeCall` (Agent makes a 3pcc or 1pcc outbound call to the customer/external party through a media gateway `Trunk` DN), the order of precedence for the geo-location configuration is:

1. `AttributeExtensions` in `TRouteCall`
2. `Routing Point` DN
3. Agent DN
4. `Trunk` DN

Feature Limitation

Geo-location for call recording may not work in cases where multiple MSML media services are required.

Call Release Tracking

If configured, SIP Server can provide information about which party—agent or customer—initiated the release of a call. Added to historical and real-time reporting, this information is useful for different applications.

Call release tracking is available for all 3pcc and 1pcc call release scenarios.

DN-Based Reporting

In DN-based reporting, information about who released the call is reported in the `AttributeExtensions` using extension key `ReleasingParty` in `EventReleased` and `EventAbandoned` events, when those events are distributed.

SIP Server includes one of the following values in the `ReleasingParty` key:

- `1 Local`—The call is released because the `ThisDN` value in the `EventReleased` requested the release.
- `2 Remote`—The call is released because the other party (which is remote to `ThisDN`) in the `EventReleased` or `EventAbandoned` events requested the release.
- `3 Unknown`—The call is released, but SIP Server cannot determine the release initiator.

Feature Configuration

Table 34 describes how to enable call release tracking.

Table 34: Configuring Call Release Tracking

Objective	Key Procedures and Actions
Configure the SIP Server Application.	In the `TServer` section of the SIP Server `Application` object, configure the following option: • `releasing-party-report`—Set this option to `true`.

Call Supervision

Call supervision functionality is designed to enable contact center managers to monitor agent DNs, and it also enables agents to invite their supervisors to the call when dealing with a customer.

SIP Server supports the following call supervision scenarios:

- Standard Call Supervision—Enables supervisors to monitor agent DNs where supervisors and agents are located on the same site.
- Multi-Site Supervision—Enables supervisors at a local site, from an endpoint controlled by a local SIP Server, to monitor remote agents, whose endpoints are controlled by another SIP Server. See "Multi-Site Supervision" on page 150.
- Remote Supervision feature—Enables supervisors to monitor agents from outside the contact center—for example, from an off-premise cell phone. See "Remote Supervision" on page 153.

Overview

There are two types of call supervision that SIP Server supports:

- *Subscription* monitoring enables supervisors to subscribe and monitor one agent DN. If the subscription is active, SIP Server automatically invites the supervisor to all calls where the agent DN participates. SIP Server stops working in this mode when the subscription is cancelled.
- *Assistance* monitoring is activated by an agent by issuing an assistance request sent to the supervisor. The agent can issue this while he or she is on a call with a customer.

Supervision Modes

Call supervision is performed in three different modes:

- *Silent monitoring* hides the supervisor's presence from all call participants, including the monitored DN for the agent who is the target of the supervisor's attention.
- *Whisper coaching* hides the supervisor's presence from all call participants but the monitored agent. Only the agent can hear the supervisor.
- *Open supervisor presence* invites the supervisor to the call through subscription or assistance call supervision scenarios, but all call participants are aware of the supervisor's presence and can hear the conversation.

The supervisor can choose any of these three modes for the call supervision subscription, but the agent can only use the last two modes for an assistance request.

Supervision Scopes

The call supervision scope specifies the time frame when the supervisor must participate in the call. There are two supervision scopes available:

- *Agent scope* allows the supervisor to monitor the agent. The supervisor joins the call when the call is established on the agent's monitored DN. The supervisor leaves the call immediately after the agent leaves the call.
- *Call scope* allows the supervisor to control the customer's experience. The supervisor joins the call when the call is established on the agent's monitored DN, or when the supervisor receives the assistance request from the agent. SIP Server keeps the supervisor as part of the call as long as either a customer or monitored agent remains in the call.

The supervisor can choose either of these scopes for the monitoring subscription.

An assistance request issued by the agent does not specify the supervision scope, so the scope always contains the `call` value. Therefore, if a supervisor is invited to a call through an assistance request, he or she will stay on the call until the call is finished.

Supervision Types

The call supervision type specifies the number of calls to be monitored—either one call or all calls.

- If *one call* is chosen for the subscription, the subscription is cancelled automatically when the supervisor finishes monitoring the first call on the monitored DN.
- If *all calls* is chosen for the subscription, the supervisor must cancel the subscription manually when he or she wants to stop monitoring the agent's calls.

The call supervision type cannot be specified for an assistance request. The `one call` type is always used when call supervision is initiated through an assistance request. The type cannot be changed through the configuration settings.

Monitoring Session

A *monitoring session* is the process in which a supervisor listens to an agent-customer conversation. There are two types of monitoring sessions that are defined by the session creation scenario:

- A *subscription session* is created by SIP Server automatically when a call is delivered to an agent's DN using the existing call supervision subscription.
- An *assistance session* is created as a result of the assistance request sent by an agent to a supervisor.

A monitoring session of any type must be initialized with the following three parameters when it is created:

- Supervision type
- Supervision mode
- Supervision scope

These parameters in the subscription session are initialized with the values of the corresponding parameters in the subscription from which this session was derived. An assistance session uses information passed in the assistance request and includes some configuration parameters for the initialization purpose. See "Call Supervision Configuration" on page 144 for more information.

A monitoring session begins when a supervisor joins a call, and ends when the supervisor disconnects from the call.

One call can have monitoring sessions of both types, which are active at the same time. Each monitoring session is uniquely identified by the supervisor involved. As a result, the supervisor can participate in only one monitoring session at a time, but one agent can be part of multiple monitoring sessions where one of the sessions is subscription-based and other sessions are assistance request-based.

The following example demonstrates how multiple monitoring sessions are created in one call:

- Agent1 answers an incoming call and Supervisor1 is invited to the call based on the existing subscription.
- Agent1 sends an assistance request to Supervisor2 who also joins the call.

This call has two monitoring sessions active at the same time: the first session has a subscription type, and the second session is an assistance session.

Intrusion

Intrusion occurs when a supervisor activates a new call supervision subscription to monitor an agent who is currently on a call. SIP Server creates the requested subscription and immediately invites the supervisor to join the existing call.

Monitoring Consultation Calls

SIP Server supports the monitoring of consultation calls made to or from a DN under call supervision. This feature is disabled by default. To enable this feature, use the option `monitor-consult-calls`.

If enabled, monitoring can take place in the following two ways:

- Consultation call initiated by the agent—If the agent under supervision with scope set to `Agent` initiates a consultation call, the supervisor will continue listing to the held call until the consultation call is established. At

that time, the supervisor will be re-invited to the consultation call using the same dialog. The agent under supervision can alternate with or reconnect to the main call, in which case the supervisor will monitor the "active" call. After the transfer is complete, the agent releases from the call and supervision is terminated. After the conference is completed, the supervisor will continue to monitor the conference call. If the agent under supervision is instead using the `Call` scope, the supervisor will then continue to monitor the main call even after the consultation call is established.

- Consultation call received by the agent—If the agent under supervision receives a consultation call (`Call` or `Agent` scope), the supervisor is connected to the consultation call and will later be connected to the main call when the transfer/conference is completed.

When enabled, consultation call monitoring is supported for both supervision scopes (`Agent` and `Call` scope), both supervision types (*all calls* and *one call*), and all supervision modes (`Open`, `Silent`, and `Whisper`). Remote and multi-site supervision also support consultation call monitoring.

Note the following:

- Intrusion—If a supervisor intrudes on an agent currently in a consultation call, the supervisor will monitor the 'active' (non-held) call. If intrusion is made with *agent* scope, the supervisor will continue to monitor the 'active' call of the agent after alternate/reconnect. For *call* scope, the supervisor will continue to monitor the same call. To allow intrusion, the option `intrusion-enabled` must be set to `true`.
- Supervision Type—For "one call" type, the consultation call and main call are considered the same. This means that if the main call is currently under "one call" type supervision, supervision will still be performed on the consultation call if initiated by the agent under supervision.
- Assistance Request—If an agent requests assistance during a consultation call, then alternates back to the main call, the supervisor will listen to hold music and will not reconnect with the main call.
- Double monitoring—The calling and called party cannot both be monitored at the same time. When a consultation call to a monitored destination is made by a monitored agent, the supervisor monitoring the agent (not the destination) will join the call.
- If a consultation call is answered, but the supervisor has not yet answered the alerting call, the call to the supervisor will be dropped on transfer complete.

Switching Between Supervision Modes

Introduced in SIP Server 8.1.101.38

A supervisor can switch between any supervision modes—silent monitoring, whisper coaching, or open supervisor presence—in MSML-based call monitoring, as follows:

- To switch from any mode to `connect` (or open supervision), the supervisor uses a `TSetMuteOff` request.

- To switch from any mode to `mute`, the supervisor uses a `TSetMuteOn` request.
- To switch from `mute` to `coach`, the supervisor uses a `TSetMuteOff` request with the `MonitorMode=coach` extension key.
- To switch from `connect` to `coach`, the supervisor uses a `TSetMuteOn` request with the `MonitorMode=coach` extension key.

When a supervisor changes the supervision mode using the `TSetMuteOff` or `TSetMuteOn` request, SIP Server generates an `EventPrivateInfo(4024)` message with the `MonitorMode` key in `AttributeExtensions` to the supervisor and agent DNs, and all of subscribed T-Library clients.

Switching between supervision modes can be performed only during an established supervision call (with a supervisor present on the call), and from the same supervisor DN from which the `TMonitorNextCall` request was sent.

Notes:
- This feature is supported for Assistance Supervision and Multi-site Supervision, and for both monitoring scopes `agent` and `call`.
- This feature depends on support from specific versions of Workspace Desktop or a T-Library client. Consult corresponding documentation for the availability of this new feature in those components.

Feature Configuration

In the `TServer` section of the SIP Server Application, configure the following options:

- `msml-support`—Set this option to `true`.
- `sip-enable-call-info`—Set this option to `true`.

Feature Limitation

Supervision modes cannot be changed during the remote supervision session.

Customer-on-Hold Privacy

Introduced in SIP Server 8.1.101.87

Some countries require that a customer who is on hold must be muted to the supervisor and agent(s) who are sharing the call.

Conference Behavior

In these examples: a customer, one or more agents, and a supervisor share a conference call.

Example 1

ON THE CALL: Customer, agent, supervisor (in Whisper mode).

ACTION: The agent puts the customer on hold.

RESULT: The customer hears music, and is muted to everyone else.

Example 2

ON THE CALL: Customer, agent, supervisor (in Open mode).

ACTION: The agent puts the customer on hold.

RESULT: The customer and the supervisor can still converse.

Example 3

ON THE CALL: Customer, two agents, supervisor (in Whisper mode).

ACTION: The first agent puts the customer on hold.

RESULT: The customer and the second agent can still converse.

Notes:
- This feature applies to MSML mode only.
- If the recording is activated on the inbound (customer) trunk, the customer will be recorded even while on hold. If the recording is activated on the agent leg, the customer will not be recorded while on hold.

Feature Configuration

1. In the `TServer` section of the SIP Server Application, configure the following options:
   - `sip-enable-call-info`—Set this option to `true`.
   - `monitor-party-on-hold`—Set this option to `false`.
   - `msml-support`—Set this option to `true`.
2. Verify that the `sip-enable-call-info-extended` is set to `true`.
3. In the `TServer` section of `Trunk` DNs (for all trunks between SIP Servers participating in the call flow), set the `sip-server-inter-trunk` option to `true`.

Call Supervision Configuration

This section describes how to configure call supervision. It covers the following topics:

Subscription

Call supervision subscription is controlled by two T-Library requests:

- `TMonitorNextCall`
- `TCancelMonitoring`

The supervisor's desktop must be able to process these two requests to perform call supervision.

The first request creates a new subscription, and the second request cancels the existing subscription. These requests use `AttributeThisDN` to identify the supervisor and `AttributeOtherDN` to identify the monitored agent DN.

Subscription Creation

SIP Server creates a new subscription based on the `TMonitorNextCall` request from the supervisor. The request is either accepted or rejected.

SIP Server rejects the request in the following scenarios:

- The supervisor or the monitored agent DN already has an active subscription.

  However, if the `TMonitorNextCall` request tries to activate a monitoring subscription that is already active (for example, the supervisor who submitted this request is already set up to monitor the agent), SIP Server responds with standard `EventMonitoringNextCall` messages sent to the agent and supervisor DNs. This request is not rejected, because it does not create multiple subscriptions on one DN.
- The supervisor or the agent DN is not configured in the Configuration Layer.

If the request is accepted, SIP Server creates a new subscription and initializes it with the type, mode, and scope information that was defined in the request.

This information is part of the request as the following attributes:

- `AttributeMonitorNextCallType`, which defines the type of call supervision. Its possible values are `MonitorOneCall` and `MonitorAllCalls`.
- `AttributeExtensions/MonitorMode`, which defines the mode of call supervision. Its possible values are `normal`, `mute`, `coach`, and `connect`.
- `AttributeExtensions/MonitorScope`, which defines the scope of call supervision. Its possible values are `call` and `agent`.

If one or both of the monitoring extensions are missing or incorrect, the following values are used:

- `default-monitor-scope` for `MonitorScope`
- `default-monitor-mode` for `MonitorMode`

SIP Server confirms the new subscription for both the supervisor and the agent by sending an `EventMonitoringNextCall` message to both destinations. This event always contains `AttributeExtensions` that include both monitoring extensions. These extensions represent the monitoring configuration for a new subscription.

See "Using the Extensions Attribute" on page 414 for more information.

Notes:
- SIP Server identifies the agent to whom call supervision will be applied by the agent DN specified in the `OtherDN` attribute of the `TMonitorNextCall` request. The agent's login ID is not used for this purpose. In particular, this means that SIP Server does not try to identify the agent who is logged in on the monitored DN, or to analyze the agent's state to decide if supervision should be activated for a call. SIP Server monitors calls made to or from the specified DN, regardless of the person using this DN, until supervision scope expires (see "Supervision Scopes" on page 140).
- Supervision starts only when a call is delivered to an agent from a Routing Point/ACD Queue.

Subscription Cancellation

SIP Server can cancel active subscriptions using the following methods:

- Manual, where a supervisor submits a `TCancelMonitoring` request.
- Automatic, where SIP Server cancels the subscription when a `MonitorOneCall`-type monitoring session is terminated.

A supervisor can submit a `TCancelMonitoring` request at any time. SIP Server identifies a subscription by the pair of supervisor and agent DNs. If this subscription exists, then it will be cancelled. Otherwise, SIP Server returns an `EventError` message.

SIP Server generates `EventMonitoringCancelled` events for both the supervisor and the agent to inform them that the subscription was cancelled.

Assistance Request

An assistance request is a `TSingleStepConference` request containing the `AssistMode` parameter in the extensions. SIP Server creates a new monitoring session based on the assistance request, but a monitoring subscription is not created.

The `AssistMode` extension is identical to the `MonitorMode` extension used in the `TMonitorNextCall` request. The difference is that `AssistMode` can contain only the `connect` and `coach` values.

There are no parameters to define the scope and type of monitoring in an assistance request, so the following monitoring parameters are used:

- `MonitorScope` set to `call`
- `MonitorType` set to `MonitorOneCall`

These two settings are hard-coded and cannot be changed.

Supervisor Auto-release

Depending on the type of monitoring scope and mode, SIP Server determines whether to release a supervisor from the call. If the monitoring scope is `agent`, SIP Server releases the supervisor from the call at the same time that the monitored agent leaves the call. If the monitoring scope is `call` and the other party of the call is aware of the supervisor's presence on the call and can hear this supervisor, SIP Server does not release the supervisor from the call.

Call Scenarios

This section presents two-party and three-party call scenarios to demonstrate how auto-release rules work.

Example 1 Three-party call, `MonitorScope=call`:

1. A call is established with three parties: a caller, a supervisor, and Agent 1 (a monitored target of the supervisor).
2. Agent 1 transfers the call to Agent 2 (whose DN is not monitored by the supervisor).
3. The call now has the following parties: the caller, the supervisor, and Agent 2.

   The supervisor is not released in this step, because `MonitorScope` is set to `call`, and the call is not finished yet (the monitor scope has not expired).
4. The caller hangs up. Now this call contains only two parties.
5. One of the following happens:
   - If `MonitorMode` is set to `mute` or `coach`, SIP Server will release the supervisor and the call, because the supervisor is on the call with the agent (Agent 2) whose DN is not the monitoring target of this supervisor; and the agent is not aware of the supervisor's presence.
   - If `MonitorMode` is set to `connect`, SIP Server will not release the supervisor, so Agent 2 can hear the supervisor.

Example 2 Three-party call, `MonitorScope=agent`:

1. A call is established with three parties: a caller, a supervisor, Agent 1 (whose DN is a monitored target of the supervisor).
2. Agent 1 transfers a call to Agent 2 (whose DN is not monitored by a supervisor).
3. SIP Server releases the supervisor from the call. The caller and Agent 2 remain on the call.

Example 3 Three-party call with recording, `MonitorScope=call, MonitorMode=mute`:

1. A call is established with three parties and a recorder: a caller, a supervisor, Agent 1 (whose DN is a monitored target of the supervisor), and the recorder.

2. The caller hangs up. Now this call contains three parties: Agent 1, the supervisor, and the recorder.
3. SIP Server releases the supervisor and the call, because `MonitorMode` is set to `mute` and the agent cannot talk to the supervisor.

Example 4 Three-party call with recording, `MonitorScope=call, MonitorMode=connect`:

1. A call is established with three parties and a recorder: a caller, a supervisor, Agent 1 (whose DN is a monitored target of the supervisor), and the recorder.
2. Agent 1 transfers the call to Agent 2 (whose DN is not monitored by the supervisor).

   Now the call has the following parties: the caller, the supervisor, Agent 2, and the recorder. The supervisor is not released in this scenario, because `MonitorScope` is set to `call,` and the call is not finished yet (the monitor scope is not expired).
3. The caller hangs up.
4. SIP Server does not auto-release the call, but will enable the supervisor to continue talking to Agent 2.

Hiding Supervisor's Presence

A supervisor who is performing silent monitoring or whisper coaching must be hidden from other call participants. If the scenario involves whisper coaching, only the monitored agent (who can hear the supervisor) must be aware of his or her presence on the call.

Call participants receive information about other participants joining or leaving the call from the corresponding T-Library events distributed by SIP Server. Supervisor presence is not shown to any new participant joining the call. The T-Library desktop applications used by call center employees must be able to process T-Library events and indicate recent changes in a call status. For example, they can show that new participant has just joined or left the call.

Hiding a supervisor's presence means filtering out any events that inform other participants about the supervisor's activity. SIP Server inserts specific information into the T-Library events that allow T-Library clients to decide if a particular event must be shown to the customer or it must be suppressed. SIP Server makes modifications to the events if at least one monitoring session is active on a call. The following attributes support this functionality:

- `AttributeCallState`
- `AttributeOtherDNRole`
- `AttributeThirdPartyDN`
- `AttributeThirdPartyDNRole`

The details on how those attributes are modified are found in the *Genesys Events and Models Reference Manual*.

Configuration Options

The following SIP Server Application-level options support call supervision functionality:

- `cancel-monitor-on-disconnect`
- `default-monitor-mode`
- `default-monitor-scope`
- `intrusion-enabled`
- `monitor-internal-calls`
- `monitor-consult-calls`

Feature Limitations

The following known limitations currently apply to call supervision:

- Genesys recommends that you not configure agent-greeting functionality for a supervisor that is currently configured for Supervisor Monitoring.

  For NETANN-based call monitoring, when agent-greeting functionality is enabled for the supervisor and silence monitoring is requested, then both agent and caller will hear the greeting when the supervisor joins the call.
- For NETANN-based call monitoring, if the supervisor changes the supervisor mode during a call (with MuteOn/MuteOff), then the mode will be changed for both the consultation call and the main call. For MSML-based call monitoring, the supervisor mode will be changed only for the call where a corresponding T-Library request is submitted.
- Call supervision functionality is disabled for video calls.
- A supervisor participating in a monitoring session should not initiate a 1pcc or 3pcc call transfer or conference call because this can change either the supervisor's status in the conference call or the status of a new party added to the call because of the conference or transfer.
- If a supervisor is already engaged in a call when an agent DN that it is targeting joins a new call (which requires monitoring), SIP Server does not invite the supervisor to monitor the new agent conversation. Even if the supervisor disconnects from its current call, the monitoring session for the new agent conversation will not start. SIP Server will activate monitoring on the next call on the targeted DN. This limitation is applied to supervision initiated through subscription monitoring (`MonitorMode`) and does not apply to the assistance monitoring (`AssistMode`).
- Call supervision functionality is supported only when Genesys Media Server is used as the MCU. This is because SIP Server sends proprietary information in the SIP messages to set up a specific conference mode that can only be interpreted by Media Server.

- When two agents are monitored by two different supervisors, and one agent calls the other agent, SIP Server invites only one supervisor to the call.
- Call supervision of the ACD Queue is not supported.
- Call supervision of the Hunt Group is not supported. Hunt Group members can only be monitored.
- If a supervisor is using whisper coaching with `MonitorScope` set to `call`, and the agent under supervision consults to an agent who is also under coach supervision, then after the transfer or conference is complete, the new agent will also be able to hear the supervisor on the main call. Normally, in all other conditions, only the specific agent under supervision can hear the supervisor when using whisper coaching, even if another agent joins the call.
- Call supervision of outbound predictive calls in ASM mode is not supported.

See also: known limitations that apply to multi-site supervision on page 153 and remote supervision on page 160.

Multi-Site Supervision

When SIP Servers operate in a multi-site environment, a supervisor at a local site, from an endpoint controlled by a local SIP Server, can monitor remote agents whose endpoints are controlled by another SIP Server.

A supervisor can switch between modes as described in "Switching Between Supervision Modes" on page 142.

Feature Configuration

To enable this feature, both the supervisor's SIP Server and the agent's SIP Server must be configured for mutual multi-site access with the ISCC transaction type `route` or `direct-uui` (see Chapter 9, "Multi-Site Support," on page 659). Additionally, a special `Routing Point` DN, dedicated for multi-site supervision, must be configured under the agent's `Switch` object. The `Routing Point` number must be specified in the `observing-routing-point` option of the agent's SIP Server `Application` object. A special routing strategy must be loaded on the observing Routing Point to route the observing call leg to the supervisor. (See "Routing Strategy Design Sample" on page 151.)

A multi-site monitoring session can be initiated by a T-Library client connected to the supervisor's SIP Server by issuing a `TMonitorNextCall` request. The request must contain:

- The `Location` parameter with the `remote` value
- (Optional) The `MonitorMode` parameter
- (Optional) The `MonitorScope` parameter

If optional parameters are not specified in the TMonitorNextCall request, the values will be taken from the default-monitor-mode and default-monitor-scope configuration options of the agent's SIP Server Application object.

The TMonitorNextCall request, issued by a T-Library client to the supervisor's SIP Server, is transmitted through the ISCC connection to the agent's SIP Server and registered on both servers.

After a call has been answered by an agent, the agent's SIP Server initiates the observing service by creating a call leg to the Routing Point specified by the observing-routing-point option.

The EventRouteRequest message generated by the agent's SIP Server reports the supervisor's switch name and the number in the Location and Number extensions respectively. The routing strategy loaded on the observing Routing Point must use this information to route the observing leg of the call to the supervisor's endpoint.

When the supervisor answers, he or she will be connected to the call in the mode defined by the MonitorMode parameter of the TMonitorNextCall request.

During a multi-site supervision session, the supervisor's connection to the monitored call can be changed between the initial MonitorMode and an open supervisor presence, with the TSetMuteOff and TSetMuteOn requests containing the supervisor's DN in the dn parameter. A supervision session can be canceled with the TCancelMonitoring request.

Note: SIP Server supports TSetMuteOn and TSetMuteOff for established conferences only to allow for service observing.

Routing Strategy Design Sample

This section provides a routing strategy design sample (see Figure 8), which should be loaded on the observing Routing Point at the agent's SIP Server to support multi-site supervision.

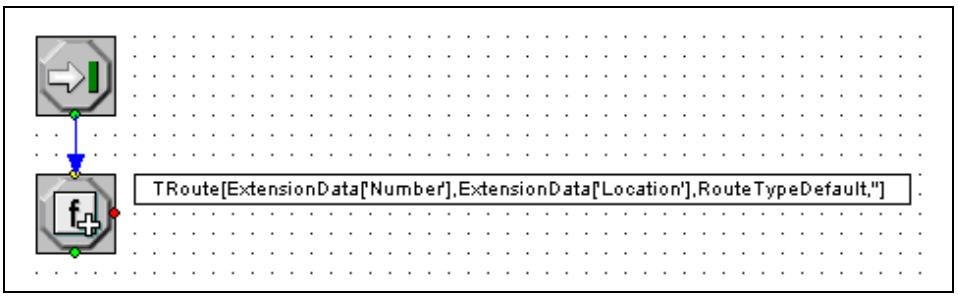

Figure 8: A Routing Strategy Design Sample

The sample strategy uses a single Multi Function routing object (see Figure 9). The supervisor's number and switch name are retrieved from the EventRouteRequest extensions by the ExtensionData function. These values are passed to the TRoute function in the Destination and Location parameters.

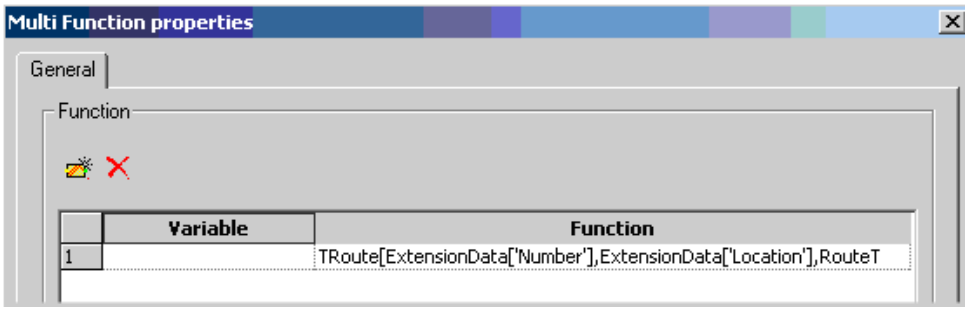

Figure 9: The Multi Function Object

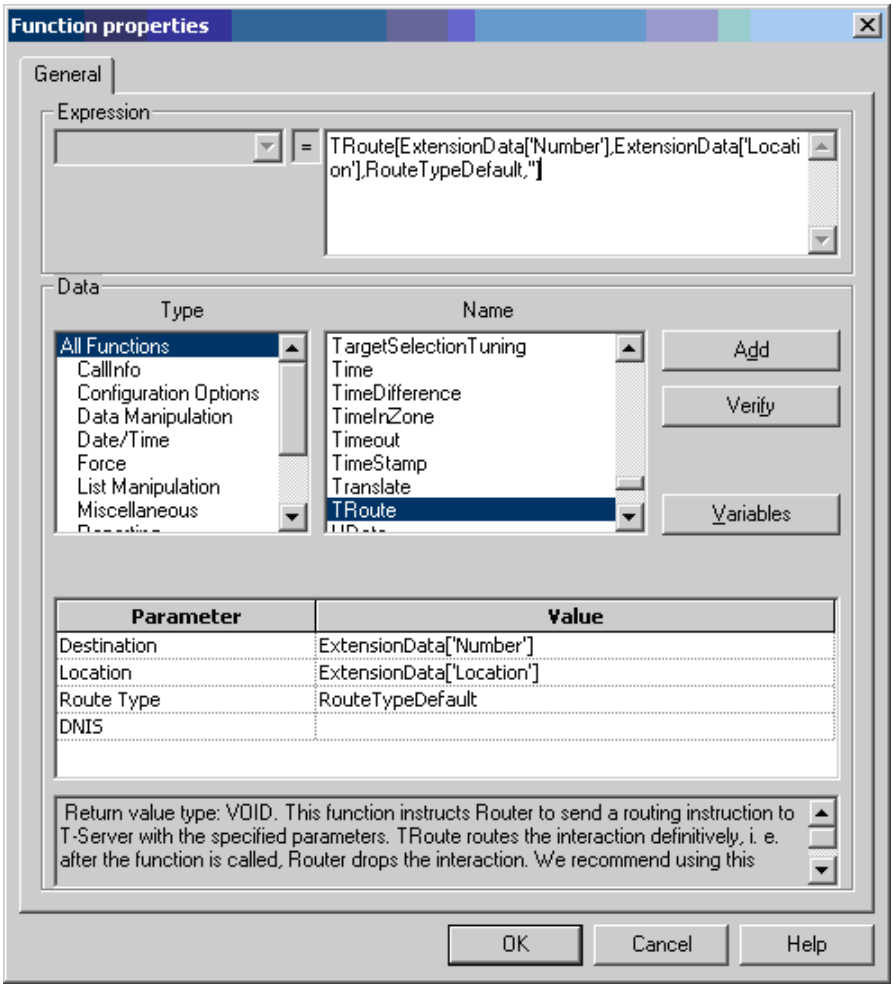

Figure 10: The Function Properties

Feature Limitations

The following known limitations currently apply to multi-site supervision:

- SIP Server will report a correct `DNRole Observer (10)` parameter in corresponding events for a supervisor's DN only if SIP Server operates in a pure SIP environment.
- If a supervisor connects to a multi-site call, the `EventPartyAdded` message is not delivered to the remote SIP Servers. Only T-Library clients registered on DNs configured on the same SIP Server as the Supervisor will receive the `EventPartyAdded` message.
- When SIP Server distributes `EventPartyAdded` toward an agent DN that is under multi-site supervision, this event contains a Routing Point specified in the `observing-routing-point` as `AttributeOtherDN` rather than the supervisor's DN.
- The `call-monitor-acw` option does not apply to the supervisor in multi-site supervision scenarios.

Remote Supervision

The Remote Supervision feature enables supervisors to monitor agent calls from outside the contact center—for example, from an off-premise cell phone. Call prompting when the supervisor first dials into the contact center is used to determine whether the supervisor is authorized to access the service, what target they want to monitor, and for how long.

The Remote Supervision feature includes the following functionality:

- Credentials check—SIP Server can check login and password credentials to verify that the supervisor is authorized to access the service.
- Targeted monitoring—The supervisor can choose to monitor either an individual agent or calls distributed to agents from a particular Routing Point or ACD Queue.
- Session persistence—The session can continue after the first monitored call ends, and for all consecutive calls (for the selected target), until the supervisor decides to hang up. In between calls, the supervisor's call is parked.
- DN translation—If configured for it, SIP Server can translate the supervisor's external DN to an internal DN so Reporting can monitor the call.
- Standard Call Supervision supported—Remote Supervision also supports the following Call Supervision functions: Supervision Modes and Supervision Scopes. For a description of these functions, see "Call Supervision" on page 139.

Feature Configuration

When using this feature, a remote supervisor dials *from outside the contact center to* a Routing Point with a special URS routing strategy. The strategy collects a caller's login information. Additionally, the strategy may collect the following information from the caller (or otherwise specify):

- A desired monitoring target number
- A supervision type (`AllCalls`), mode, and scope
- An associated internal DN, used for reporting purposes
- A post-feature destination DN

The strategy places these parameters as `Extensions` attributes in the `TRouteCall` request.

Monitoring session starts by routing a remote supervisor's call to the special pre-defined DN with the number `gcti::park`. This DN is used to park the supervisor's call before call monitoring starts, and between calls when several calls are monitored.

While the call is parked, the supervisor hears silence or a music file specified by the `parking-music` option.

Procedure: Configuring remote supervision

Start of procedure

1. In GAX, select the SIP Server `Application` object and add the `parking-music` option in the `TServer` section on the `Application Options` tab. This option specifies the music file, which will be played to a remote party parked on the `gcti::park` DN.
2. Plan the Remote Supervision routing strategy to meet your specific needs.

   The sample strategy described below is a simplified prototype. You may design your own strategy to include any custom logic available in URS, implement credential verification based on accessing enterprise databases, and utilize custom prompts. As a result, the strategy should park a call on the `gcti::park` device.
3. Prepare the recordings of the voice prompts, which are used in the routing strategy to collect the caller's login information and optional feature selection.
4. In Interaction Routing Designer (IRD), design your routing strategy. See "Routing Strategy Design Sample" on page 155.
5. Save the complete strategy in IRD.
6. Load the strategy into a Routing Point by using the `Loading` tab in IRD.

7. Test your strategy by placing a call from an external phone to the Routing Point number. Calls from internal DNs are not allowed.

End of procedure

Routing Strategy Design Sample

This section describes a strategy design sample (see Figure 11), followed by the explanation of each block called in the IRD terms *routing object.*

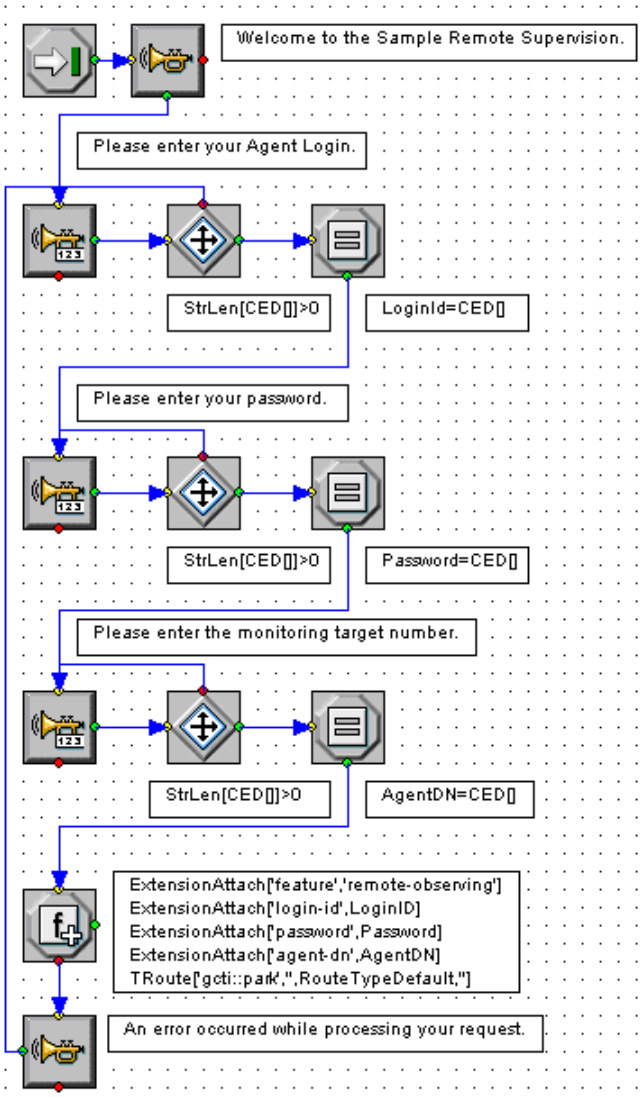

Figure 11: A Routing Strategy Design Sample

Initial Greeting

The first `Play announcement` routing object is used to play back the initial greeting (see Figure 12).

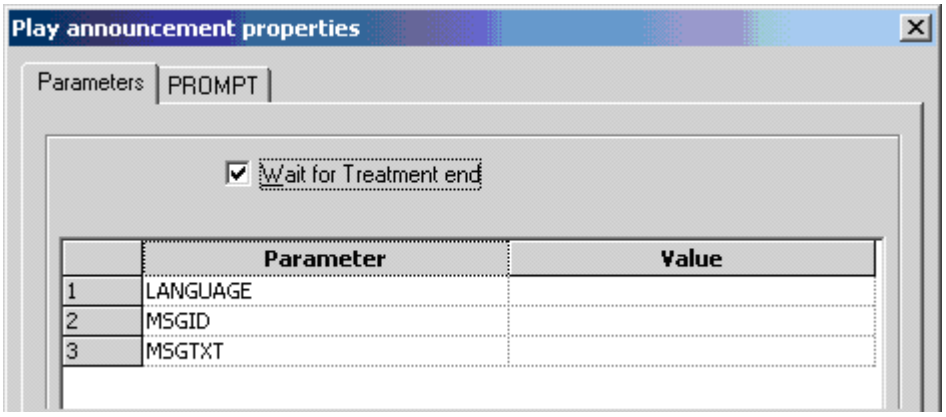

Figure 12: Initial Greeting: Parameters Tab

The `LANGUAGE`, `MSGID` and `MSGTXT` parameters are not used by SIP Server implementation of the Announcement treatment. The `Wait for treatment end` check box specifies to the URS that it should wait for the treatment to complete before proceeding to the next strategy step.

On the `PROMPT` tab (see Figure 13), the `ID` specifies the prompt to be played. The `Interruptible` flag allows the caller to skip the greeting message by pressing any key on the phone keypad.

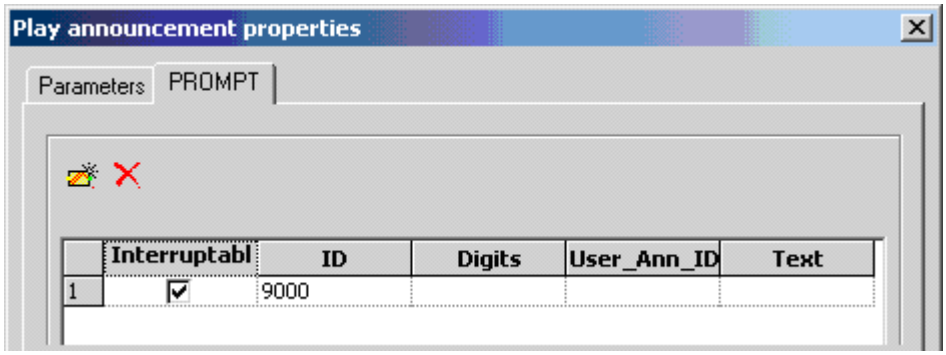

Figure 13: Initial Greeting: PROMPT Tab

Collecting Agent Login Code

Although your strategy may implement an arbitrary approach to perform caller verification, this sample strategy demonstrates the SIP Server's built-in functionality. SIP Server verifies the `login-id` and `password` information provided in corresponding extensions parameters against the Agent Login `Code` and `Password` specified in the `Agent Login` object in the Configuration Layer. Therefore, for the purpose of this strategy, ensure that you have configured `Agent Login` objects with the numeric-only Agent Login `Codes` and `Passwords` so they can be entered through the phone keypad.

After the initial greeting, the strategy uses the `Play Announcement and collect digits` routing object to request the caller's Agent Login (see Figure 14). It plays the "`Please enter your Agent login`" prompt and retrieves the caller's

digits input. The caller is expected to enter a numeric login code, up to 31 digits long, terminated by the "#" key.

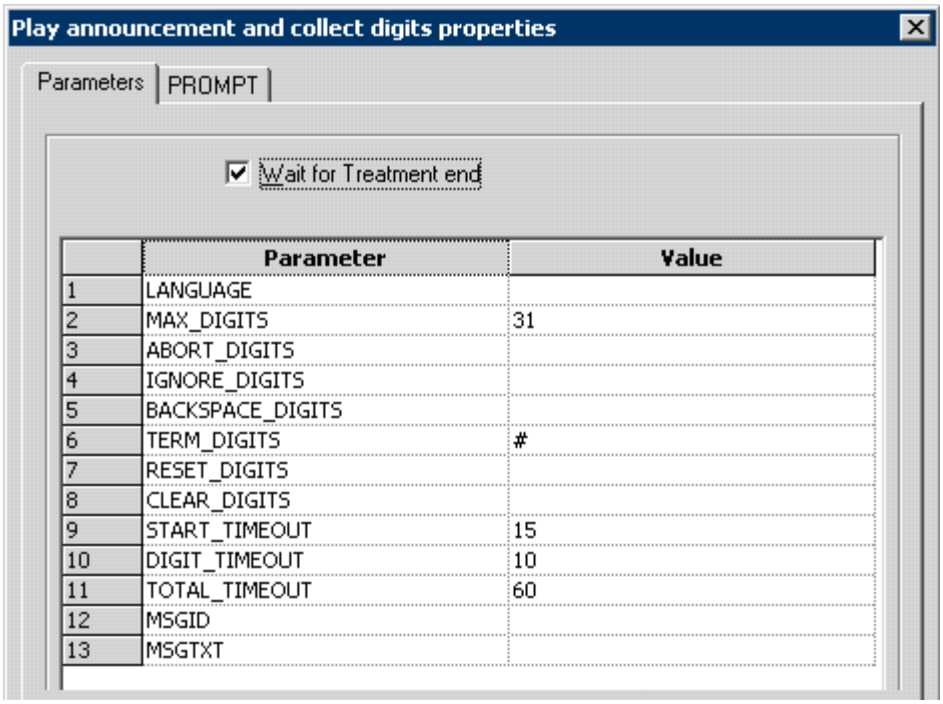

Figure 14: Collecting Digits: Parameters Tab

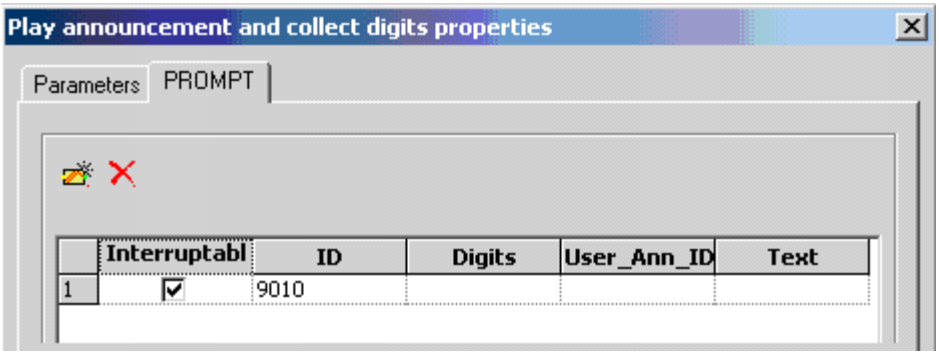

Figure 15: Collecting Digits: PROMPT Tab

Verifying Caller Input

The next `Generic segmentation` object verifies that the caller's input (returned by the `CED[]` function) is not empty (see Figure 16).

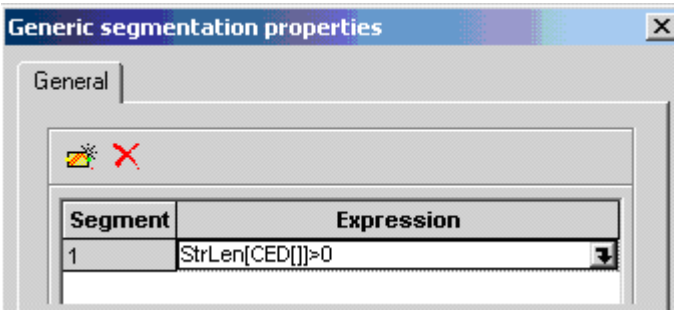

Figure 16: The Generic Segmentation Object

If the caller did not enter any digits within 15 seconds (as specified by the `START_TIMEOUT` parameter in the preceding `Play Announcement and collect digits` object, the segmentation object will repeat the previous prompt.

Storing the Entered Agent Login Code in a Variable

The next `Assign` routing object defines the `LoginId` internal strategy variable (use `Variables` button to define the variable) and assigns the caller's input value to this variable (see Figure 17).

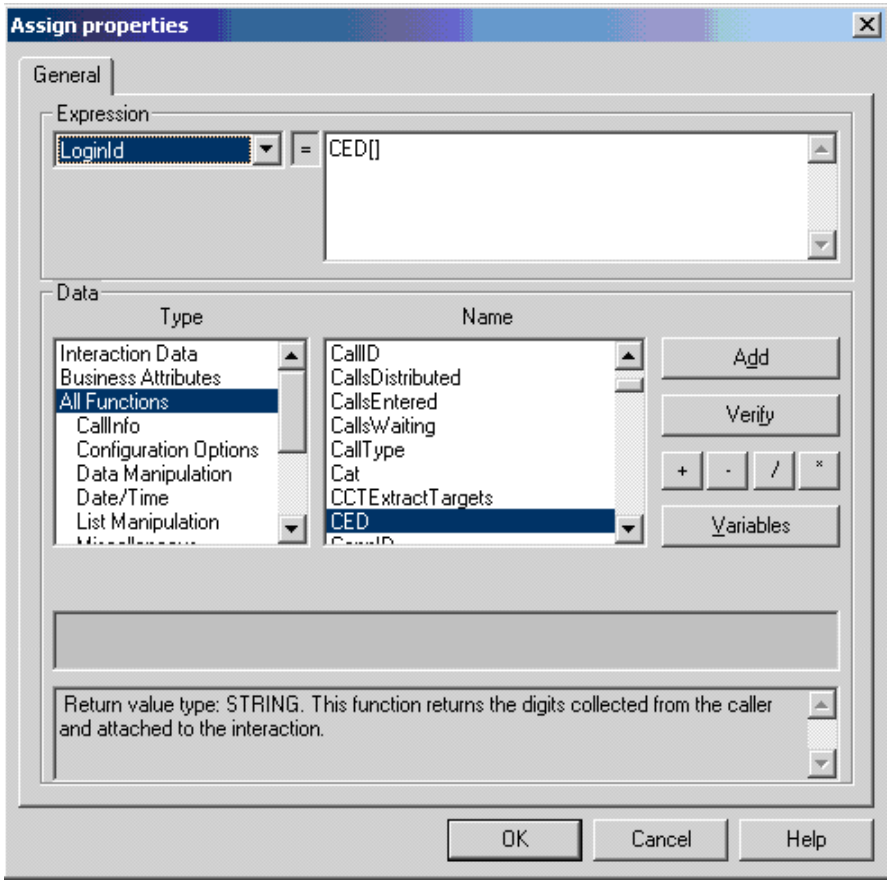

Figure 17: The Assign Object

Collecting Password and Monitoring Target

Subsequent routing strategy blocks prompt the caller to enter a password and the desired monitoring target DN number, and a sequence similar to the one used to collect the Agent Login information.

Collected input is placed into the `Password` and `AgentDN` internal variables.

Invoking Remote Monitoring

The key point of the strategy is the `Multi Function` object (see Figure 18).

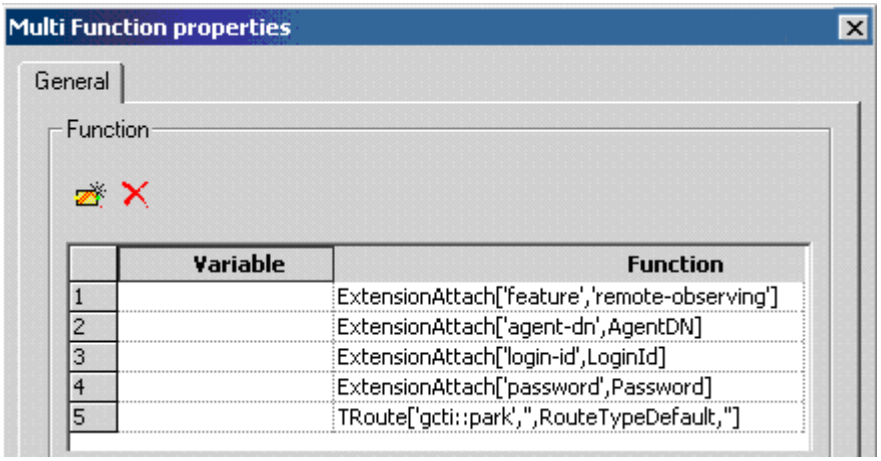

Figure 18: The Multi Function Object

The `Multi Function` object attaches the extensions parameters to the `TRouteCall` request as the request extensions, then routes the call to the `gcti::park` DN.

SIP Server determines the desired feature—remote observing from the value of the `feature` extension parameter—and verifies the user's login information.

If the supplied information is correct, SIP Server starts Remote Monitoring session. The supervisor hears silence or a music file until a call comes to the specified target. After that, the caller connects to the monitored call.

It the parameters are incorrect (for example, wrong login, password, or monitoring target information), SIP Server responds with the `EventError` message to the `TRouteCall` request. This will trigger the default branch of the `Multi Function` object.

Verifying the Result

If an error occurs for any reason, the strategy uses the subsequent `Play announcement` object to notify the caller about the error and returns the control flow to the point where the user is requested to enter the Agent Login again.

Use of the Extensions Attribute

The following `Extensions` attribute parameters can be used to configure the application that starts and defines the remote supervision session:

- `feature`—This key, with the `remote-observing` value, triggers registration of the routed party as a supervisor with parameters specified in additional `Extensions` described in this section.
- `dn`—An optional DN number that can be used during the monitoring session as a substitute for the external PSTN number that the supervisor used to dial in. If you do not include this parameter, no TEvents will be distributed for this DN.
- `login-id` and `password`—These optional parameters are used to establish that the supervisor is authorized for remote access to the feature. The treatment can prompt for the `login-id` alone, or for both the `login-id` and the `password`.
- `agent-dn`—The target that the supervisor wants to monitor. This can be a Routing Point or an agent DN.
- `monitor-type (AllCalls)`—An optional parameter that enables the supervisor to monitor all consecutive calls for the selected target, until the supervisor decides to hang up and end the monitoring session. In between monitored calls, the supervisor's call is parked.
- `post-feature-dn`—An optional parameter that specifies a Routing Point to which the supervisor will be connected after the supervision session.

Feature Limitations

The following known limitations currently apply to remote supervision:

- `MonitorMode` of remote supervision session cannot be changed during active supervision.
- `One Call` supervision type is not supported.
- When SIP Server distributes `EventPartyAdded` toward an agent DN which is under remote supervision, this event contains a Routing Point as `AttributeOtherDN` which was used during involvement of the supervisor rather than the supervisor DN.
- Remote supervision is not supported in IMS environments.
- There is no synchronization of parked remote supervisor's calls between primary and backup SIP Servers. After a switchover, the remote supervisor with a parked call is not able to monitor any calls. The supervisor must hang up and call back to the contact center to be able monitor calls.

Call Transfer and Conference

SIP Server supports the following call transfers:

- First-party call control (1pcc) transfers: single-step and two-step transfers.
- Third-party call control (3pcc) transfers: single-step and two-step transfers.

In these scenarios, the REFER request method is used. If an endpoint does not support the REFER method, the re-INVITE method can be configured for use in two-step transfers.

SIP Server can send a REFER message to the transferred party when the following scenarios occur:

- A REFER message was received from an endpoint.
- A single-step call transfer was received from a client.

This removes SIP Server from the SIP signaling loop.

SIP Server analyzes the destination specified in either scenario and then determines if a different contact is specified in the outgoing REFER message based on the following criteria:

- The destination is unknown to SIP Server (no regular DN and no Trunk DN contains the prefix that matches the specified destination).
- The destination refers to a DN of type Trunk that contains the oosp-transfer-enabled option set to true.

If either of the scenarios is true, SIP Server prepares a Contact for the Refer-To header of the outgoing REFER message, based on the following conditions:

- If there is no DN, and no DN of type Trunk is specified as the destination, the Contact information from the caller DN or the Trunk DN will be specified in the Refer-To header of the outgoing REFER message. It is the responsibility of the caller to determine where to transfer the call.
- If a Trunk DN is specified as the destination, and it contains the oosp-transfer-enabled option set to true, the contact information from this trunk will be specified in the Refer-To header of the outgoing REFER message.

Note: If the caller DN or the Trunk DN in either scenario contains the override-domain option specified, the value of this option will be specified in the Refer-To header of the outgoing REFER message.

Routing to External Destination Using REFER—Inbound Calls

SIP Server supports the routing of an inbound call to an external destination by using the REFER method. When the feature is activated, SIP Server places itself

in the Out Of Signaling Path (OOSP). This feature applies to the following scenarios:

- An inbound call is routed from a Routing Point to an external destination.
- An agent transfers an inbound call by using the single-step transfer to a Routing Point, then the call is routed to an external destination.
- An agent transfers an inbound call by using the blind transfer to a Routing Point, then the call is routed to an external destination.

Note: This feature is not applicable for scenarios (the second and the third, above) where a conference (supervision or emergency recording) is involved.

Table 35 describes how to enable routing of inbound calls to an external destination using `REFER`.

Table 35: Enabling Inbound External Routing by REFER

Objective	Related Procedures and Actions
Configure the `Trunk` DN.	In the `Trunk` DN for inbound calls, configure the following options in the `TServer` section: • `oosp-transfer-enabled`—Set this option to `true`. • `refer-enabled`—Set this option to `false`. This ensures the `REFER` for this `Trunk` DN is used only in OOSP (single-step transfer) scenarios.

Routing to External Destinations using REFER—Outbound Calls

SIP Server can also route outbound calls to an external destination using the `REFER` method, where SIP Server is placed out of the signaling path (OOSP) after the transfer.

Table 36 describes how enable routing of outbound calls to an external destination using `REFER`.

Table 36: Enabling Outbound External Routing by REFER

Objective	Related Procedures and Actions
Configure the Trunk DN.	In the Trunk DN for outbound calls, configure the following options in the TServer section: • oosp-transfer-enabled—Set this option to true. • refer-enabled—Set this option to true.

Selecting SIP Call Flows from the Routing Strategy

SIP Server supports dynamic selection of the SIP call flows for a particular call, based on the setting of the Transfer-Type key returned by a routing strategy in the TRouteCall request. The routing strategy can use this key to select the SIP call flow that will be used to deliver the call to a routing destination.

Feature Configuration

In the routing strategy, configure the TRouteCall to include the key Transfer-Type in AttributeExtensions with the value set to the type of transfer you want to enable for this call: invite, refer, or oosp.

The Transfer-Type key with values refer and invite has a higher priority than the refer-enabled option configured on a DN of type Trunk or Extension. In addition, the Transfer-Type key with the value of oosp has a higher priority than the oosp-transfer-enabled option configured on a DN of type Trunk.

If SIP Server receives a TRouteCall with Transfer-Type set to refer, but refer-enabled is set to false on the DN, SIP Server will send REFER anyway (routing strategy takes precedence). This allows the strategy to control the transfer type on a call-by-call basis.

However, SIP Server always monitors the endpoints to determine support for REFER. If the targeted endpoint does not announce support for REFER through the SIP dialog, SIP Server will not send REFER to this endpoint.

Feature Limitation

The request for a REFER method from the routing strategy is not used if an incoming call was not connected (the destination DN did not answer the call, or a treatment was not applied to an unanswered call), SIP Server uses the re-INVITE method instead of REFER. To ensure that REFER is actually used, change the strategy so the Transfer-Type is set to oosp instead. With this setting, SIP Server will use an Out-of-Signaling-Path REFER in cases where the call is currently being treated on the IVR, and not yet answered on the remote endpoint.

Single-Step Transfer Using re-INVITE

Scenarios in which single-step transfers use the re-INVITE request method require that the originating DN be configured with the refer-enabled option set to false. For external single-step transfers, the Trunk DN must be configured with oosp-transfer-enabled set to false.

Controlling Transfer Methods to External Destinations

To control SIP messaging (REFER or re-INVITE) that SIP Server uses to initiate transfers or routing to an external DN, configure the inbound Trunk DN according to the following rules:

- For two-step transfers, the refer-enabled setting on the Trunk DN takes precedence over oosp-transfer-enabled.
- For single step transfers, the oosp-transfer-enabled setting on the Trunk DN takes precedence over refer-enabled.

Note: Genesys recommends that you not use different values for the oosp-transfer-enabled option when configured on both the transfer destination Trunk and on the transferred party Trunk.

Table 37 shows how these two options control the SIP methods used.

Table 37: Trunk DN Configuration for External Transfers

refer-enabled	oosp-transfer-enabled	Single-Step Transfer/Route	Two- Step Transfer
true	true	REFER/302	REFER
false	true	REFER/302	INVITE
true	false	INVITE	REFER
false	false	INVITE	INVITE

Conference Calls

SIP Server supports third-party call control (3pcc) conferences with central mixing, using MCUs that support more than three participants.

If the initiator of the 3pcc conference drops the call, other participants will remain on the call. If any participant of the 3pcc conference issues TClearCall, the conference will end for all participants (the call will be dropped).

Single-Step Conference to External Destination

Starting with version 8.1.101.55, SIP Server supports a `TSingleStepConference` request to an external destination, which, for example, enables bringing an expert in to the conference without putting the caller on hold. The single-step conference operation can be performed from a two-way call or from a pre-existing conference. While waiting for the destination to answer the call, existing call parties will continue hearing each other. If the destination party does not respond or rejects the request, the call returns to the previous state.

Note: In multi-site deployments, the single-step conference operation to another site via ISCC is not supported.

Silence Treatment in Conference

SIP Server can provide a silent treatment for conference call participants when one of them places the call on hold. This will allow conference call participants to continue the conference without interruption (or hearing the music-on-hold treatment). This feature is applicable to conference calls where participants are located in single-site or multi-site environments.

For this feature to work, the `music-in-conference-file` option must be set to the valid name of the silent audio file to be played in applicable conferences (more than two active participants).

Deleting Party From Conference in Multi-site Deployments

Starting with version 8.1.101.57, SIP Server supports `TDeleteFromConference` requests in multi-site deployments in the same way as in single-site deployments; that is, any agent can remove any other party from the conference using a `TDeleteFromConference` request containing a targeted party DN.

Feature Configuration

To enable `TDeleteFromConference` requests support in multi-site deployments, configure the SIP Server Application, as follows:

1. In the `TServer` section, set the following configuration options:
   - `sip-enable-call-info`—Set this option to `true`.

     The Call Participant Info functionality must be activated, enabling SIP Server to maintain an `LCTParty` list containing DNs and their locations for all parties present in the call. The LCTParty list is distributed to a T-Library client in `EventUserEvent`.
   - `sip-remote-del-from-conf`—Set this option to `true`.

2. In the `extrouter` section, set the `use-data-from` configuration option to `current`. This enables Party Events propagation.

Feature Limitations

- In a multi-site conference in which two DNs have identical names, if `TDeleteFromConference` is requested to remove a DN with the duplicate name, either one or both parties can be deleted from the conference.
- In multi-site scenarios, real-time statistics related to call supervision (particularly `CallObserved`) may be incorrect if the supervisor is released from the call before the call is finished. See Stat Server documentation for details.

Private Conversations During Conference

Introduced in SIP Server 8.1.101.78

SIP Server supports T-Library requests `TListenDisconnect` and `TListenReconnect`. These requests can be used in a conference with three or more participants. Any agent who is using a T-Library desktop can submit a `TListenDisconnect` request to disconnect any other party from the conference temporarily. The disconnected party hears music and cannot hear the remaining participants, who can continue their conversation. Remaining conference participants also cannot hear the disconnected party. To return the disconnected party back to the conference, one of the agents in the call submits a `TListenReconnect` request.

If an agent disconnects another participant from the conference and then leaves the conference, the disconnected party remains disconnected until only one active participant exists in the conference. After that, SIP Server releases the conference and establishes the dialog between two remaining parties (the formerly disconnected and active parties).

SIP Server supports `TListenDisconnect` and `TListenReconnect` requests in accordance with the T-Library call model where SIP Server generates `EventListenDisconnected` and `EventListenReconnected` events in responses to the two corresponding requests. `EventListenDisconnected` is always distributed with `AttributeCallState` set to `CallStateHeld`, which indicates that the disconnected party cannot hear and cannot be heard by other members of the conference.

This feature must be used along with the LCTParty functionality enabled in SIP Server. The state of the disconnected party is reported to all call participants with the standard LCTParty `EventUserEvent`, which contains the `LCTParty<n>_state` extension key with a value set to `ListenDisconnectedHeld`, where *n* is a party index.

`TListenDisconnect` and `TListenReconnect` requests must have the `AttributeOtherDN` set to the party alias reported through the LCTParty `EventUserEvent`.

Note: This feature depends on support from specific versions of Workspace Desktop or a T-Library client. Consult corresponding documentation for the availability of this new feature in those components.

Feature Configuration

In the `TServer` section of the SIP Server Application, set the following configuration options:

- `sip-enable-call-info`—Set this option to `true`.
- `sip-enable-call-info-extended`—Set this option to `true`.
- `music-listen-disconnect`—Set this option to the path of any valid audio file.
- (Optional) `sip-hold-rfc3264`—Set this option to `false` to avoid MCP releasing the temporary disconnected party from a conference because of the RTP timeout.

Feature Limitations

- In multi-site deployments, Genesys recommends setting the `sip-enable-moh` to `false` on inter-trunk DNs, to avoid playing music to a remote party disconnected from the conference.

Muting/Unmuting a Party in a Conference

Starting with version 8.1.102.02, SIP Server allows any conference party on the call to mute or unmute any internal party in a conference. One party can mute several others. If a party mutes some other party and leaves the conference, the muted party remains muted if more than two participants remain in the conference. If only two participants (including the muted party) remain in the conference, SIP Server drops the conference and establishes a dialog between these two parties, thus unmuting the muted party.

Starting with release 8.1.102.20, you can enable muting in two-way calls by setting the `sip-enable-two-party-mute` configuration option to `true`. That way, when a party in a two-way call issues a `TSetMuteOn` or `TSetMuteOff` request, the two-way call will be converted to a conference and a Media Server `Mute` or `Unmute` command will be issued for the requestor's leg.

Muting one of the conference's participants can be used in parallel with services, such as supervision, listen disconnect, and recording, except when the "Customer-on-Hold Privacy" is enabled. See also "Feature Limitations" on page 170.

This functionality is provided through TPrivateService requests. The Call Participant Info functionality must be activated, enabling SIP Server to maintain an LCTParty list containing DNs and their locations for all parties present in the call. The LCTParty list is distributed to a T-Library client in EventUserEvent. The OtherDN attribute of the TPrivateService request must contain the party ID received in the LCTParty list.

For all internal conference participants, SIP Server sends EventUserEvent indicating which party was muted. For a disconnected (muted) party, LCTParty[n]_mute is set to on. After a party is unmuted, LCTParty[n]_mute is not present to indicate that the party was unmuted. For the muted/unmuted party, SIP Server generates EventMuteOn/EventMuteOff, respectively.

The T-Library client must include mute/unmute-related parameters in the TPrivateService request that it sends to SIP Server, as described in Table 38.

Table 38: Mute/Unmute Parameters for TPrivateServer Request

Attribute	Value
PrivateMsgID	Specifies the type of operation to be performed: • SIPTS_PRIVATE_SERVICE_MUTE (3027)— Mutes or Unmutes a party in a conference.
ThisDN	Specifies the DN on behalf of which the mute/unmute operation is requested. This DN must be registered by the T-Library client.
ConnectionID	References the ID for the call that is currently being muted/unmuted.
Extensions	Specifies key-value pairs used to control the mute/unmute operation: OtherDN—Specifies a DN to be muted or unmuted. Mute—"on" to mute the OtherDN, "off" to unmute it.

SIP Server generates EventPrivateInfo (PrivateMsgID 4029) with the same ReferenceID as the one in the request to indicate that a Mute/Unmute request is accepted. The desktop should rely on the LCTParty of EventUserEvent to display the current party state.

Note: This feature depends on support from specific versions of Workspace Desktop or a T-Library client. Consult corresponding documentation for the availability of this new feature in those components.

Mute State Duration

In SIP Server, TSetMuteOn is applied per-call basis. The Mute state is preserved for the duration of the call or until TSetMuteOff is applied. When a new call is

created on the same DN, its Mute state is off. When a muted call is released, `EventMuteOff` is not needed and is not generated.

Examples:

- A main call can be muted, but when a consultation call is created, the consultation call starts in an unmuted state.
- When a two-step transfer or two-step conference is completed, the DN's Mute state will correspond to the Mute state of the main call. The consultation call is released and no `EventMuteOff` is generated.
- When a call is muted, then parked via the "Call Park/Retrieve" feature, and then retrieved, that call is reported as a new one and will be unmuted.
- When a Shared Call Appearance (SCA) call is muted, then parked, and then retrieved, that call is reported as a new one and will be unmuted.
- Contrary to the park scenarios, when a call is connected via the "Call Divert Destination" feature to the new divert destination, SIP Server considers and reports the diversion as part of the same call. Accordingly, no `EventReleased` is generated and the Mute state is preserved.

Situations When a Mute Operation is Prohibited

SIP Server prohibits Mute operations in the following scenarios:

- When a call is on hold, Mute or UnMute operations are not allowed.
- When a greeting is being played to a party in the call, the Mute operation is not allowed (relates to the case when the `sip-two-party-mute-enabled` option must be set to `true`).

Feature Configuration

To configure Muting/Unmuting a Party in a Conference, complete these steps:

- In the `TServer` section of the SIP Server Application, configure the following options:
  - `msml-mute-type`—Set this option to `1`.
  - `sip-enable-call-info`—Set this option to `true`.
  - `msml-support`—Set this option to `true`.
  - (Optional) `sip-enable-two-party-mute`—Set this option to `true` if required.
- Verify that the `sip-enable-call-info-extended` is set to `true`.
- In the `TServer` section of `Trunk` DNs (for all trunks between SIP Servers participating in the call flow), set the `sip-server-inter-trunk` option to `true`.
- In the `extrouter` section of the SIP Server Application, set the `use-data-from` option to `current` or `original`.

Feature Limitations

The following limitations apply to Muting/Unmuting a Party in a Conference:

- If recording is activated on the inbound (customer) trunk, the customer will be recorded even when muted. If recording is activated on the agent leg, this agent will be recorded while muted.
- If DNs with the same names configured on different switches participate in the conference, SIP Server might choose the incorrect party to mute.

Consultation Transfers and Conferences

SIP Server provides the ability for parties participating in a consultation call to initiate 3pcc (third-party call control) transfers or conferences. A typical supported scenario would be:

1. An inbound call is routed to Agent A.
2. Agent A originates a consultation call with Agent B.
3. Agent B originates a consultation call with Agent C.

Consultation Transfers for Calls on a Routing Point

SIP Server allows an agent to complete a consultation transfer of a call that is located on a Routing Point. This transfer operation is supported in single-site and multi-site environments.

In a single-site environment, the call transfer can be completed both when a treatment is playing for the call on a Routing Point, or when the call is just parked on a Routing Point.

In a multi-site environment, when a consultation call is made to a Routing Point located on another site, the call transfer can be completed only when a treatment is playing for the call on the Routing Point. If a call is just parked on a Routing Point, the complete transfer operation will not be successful and SIP Server will generate an `EventError` (`Call in invalid state`) message.

Alternating Between Main and Consultation Calls

SIP Server enables agents to handle up to three 3pcc calls on their SIP endpoint. This functionality supports alternate call operation between the main call and an answered consultation call, as well as the main call and a consultation call that is queued on a Routing Point, as described in the following scenario:

1. A call is routed to Agent A.
2. Agent A places the call on hold and initiates a consultation call by dialing to a Routing Point.
3. Agent A is placed in a queue at the Routing Point, waiting for another agent to become available (a treatment is played).

4. Agent A places the consultation call on hold and retrieves the main call from hold.

For the alternate call operation to work transparently in a multi-site environment, a treatment must be applied to a call on a Routing Point at the earliest possible time. If a treatment is not applied, the alternate call operation will not be successful and SIP Server will generate an `EventError (Call in invalid state)` message.

TCompleteTransfer using REFER or REFER with Replaces

SIP Server supports the `TCompleteTransfer` operation—which completes a previously initiated two-step transfer by merging the held call with the active consultation call—by using either the SIP `REFER` or SIP `REFER` with the `Replaces` header based on conditions described below.

- SIP Server supports `TCompleteTransfer` using the SIP `REFER` method if:
  - The transferred party and/or the transfer-destination party are internal.
  - The external transferred party and/or the external transfer-destination party do not support the `Replaces` header in the `REFER` method.
- SIP Server supports `TCompleteTransfer` using the SIP `REFER` method with `Replaces` if both the external transferred party and the external-transfer destination party support the `Replaces` header in the `REFER` method.

If `TCompleteTransfer` is performed using the `REFER` method, SIP Server stays in the signaling path. If it is performed using the `REFER` method with `Replaces`, SIP Server is taken out of the signaling path.

Feature Configuration

Table 39 describes how to enable `TCompleteTransfer` operations using `REFER`.

Table 39: Enabling TCompleteTransfer by REFER

Objective	Related Procedures and Actions
Configure the `Trunk` DN.	In the `Trunk` DN > `TServer` section: 1. Set `transfer-complete-by-refer` to `true`. 2. Set `sip-replaces-mode` to one of the following: • `0`—SIP Server only uses `REFER` if `transfer-complete-by-refer` is enabled. • `1`—SIP Server only uses `REFER` if `Allow` header contains `REFER`, and `Supported` header contains `Replaces`. • `2`—SIP Server always uses `REFER`.

Referred-By Header Support

SIP Server provides the ability to pass the identity of the party, which has originated the transfer, in the SIP URI of the outgoing `REFER` request's `Referred-By` header. In addition, SIP Server provides the ability to control the "hostport" component of the SIP URIs in `Refer-To` and `Referred-By` headers of the outgoing `REFER` requests through the configuration options.

Feature Configuration

Table 40 describes how to enable the `Referred-By` header.

Table 40: Enabling the Referred-By header

Objective	Related Procedures and Actions
Configure the SIP Server Application.	Set `sip-referred-by-support` to `true`.
(Optional) Configure a DN associated with the transferred/routed party.	If it is required to override the "hostport" component of the SIP URI in `Refer-To` and `Referred-By` headers, configure the following options on a DN where an outgoing `REFER` request is sent to: • Set `override-domain-refer-to` to the "hostport" component to be used as a "hostport" component of the SIP URI in the `Refer-To` header of the outgoing `REFER` messages. • Set `override-domain-referred-by` to the "hostport" component to be used as a "hostport" component of the SIP URI in the `Referred-By` header of the outgoing `REFER` messages.

Override of the "hostport" component of the SIP URI in the `Refer-To` header can be configured by the following options on a DN where an outgoing `REFER` request is sent to, in order of priority:

1. `override-domain-oosp`, in case of OOSP transfer
2. `override-domain-refer-to`
3. `override-domain`

Override of the "hostport" component of the SIP URI in the `Referred-By` header can be configured by the following options on a DN where an outgoing `REFER` request is sent to, in order of priority:

1. `override-domain-referred-by`
2. `override-domain`

Feature Limitations

The following known limitations currently apply to call transfer and conference scenarios:

- Blind conference calls are not supported.

- Three-way conference on the telephone is not reported properly. Call participants can talk to each other, but such a call is not reported as conference.
- SIP Server does not support the use of `REFER` for the `TCompleteTransfer` operation for calls in which a Multipoint Conference Unit (MCU) is involved. For example, a call is monitored or emergency recording is applied to a call. If regular call recording is applied to the original call, the `REFER` method can be used for the `TCompleteTransfer` operation.
- The `sip-replaces-mode` option is not supported on `Trunk` DNs that are configured between different SIP Server instances, and is ignored on `Trunk` DNs where the `sip-server-inter-trunk` option is set to `true`.
- 1pcc transfer by `REFER` with `Replaces` may fail if SIP Server receives a SIP `REFER` request with the `Replaces` parameter in the `Refer-To` header pointing to a dialog whose SIP `Call-ID` contains a % (percentage) character. This character may appear as part of the IPv6 address scope ID, and sometimes IP addresses are used as part of SIP `Call-ID` header.
- In multi-site deployments, the single-step conference operation to another site via ISCC is not supported.

Class of Service

Class of Service (COS) is functionality that defines telephony capabilities for a device or an agent. In SIP Server, COS telephony capabilities are defined by configuring Ring-through rules.

Class of Service can be assigned to the device (a `DN` object in SIP Server `Switch` configuration) or to the agent (an `Agent Login` object in SIP Server `Switch` configuration).

The COS assigned to the agent takes precedence over the COS assigned to the device. That is, when different COSs are assigned to the device and to the agent, SIP Server will use the COS assigned to the agent.

Ring-Through Rules

The ring-through rules define whether a call is sent to an agent or a device. The following ring-through rules are supported by SIP Server:

- Reject call when a device is already in a call

  This rule is enforced by the `Switch` object-level configuration option `reject-call-incall` within COS.
- Reject call when an agent is not ready on a device

  This rule is enforced by the `Switch` object-level configuration option `reject-call-notready` within COS.

Call Rejection by COS Ring-Through Rules

A call attempt can be rejected by the COS ring-through rules. To indicate this condition, SIP Server generates an `EventError` message to the corresponding request, with the reason code `Destination Invalid State (93)`. When rejecting 1pcc calls, SIP Server generates a SIP `603 Decline` error response.

Feature Configuration

Table 41 describes how to enable Class of Service.

Table 41: Configuring Class of Service

Objective	Related Procedures and Actions
1. Configure a COS DN of type `Voice over IP Service` to represent the COS entity itself.	1. Create a `Voice over IP Service` DN with the name, for example, `COS_SupportAgent`. 2. In the `Options` tab > `TServer` section, configure the following option: • `service-type = cos` 3. In the same `TServer` section, specify the ring-through rules—for example: • `reject-call-incall = true` • `reject-call-notready = true`
2. Assign the COS DN to one or multiple DNs of type `Extension` or `ACD Position` within the same `Switch` configuration object.	In the `Options` tab > `TServer` section of the DN, add the `cos` configuration option, with the value set to the name of the COS DN. For example: contact "sip:3010@172.21.9.2:5060" cos "COS_SupportAgent" dual-dialog-enabled "false" record "true" refer-enabled "true" ring-tone-on-make-call "false" sip-cti-control "talk,hold"
3. Assign the COS DN to one or multiple `Agent Login` objects.	In the `Options` tab > `TServer` section of the `Agent Login` object, add the `cos` configuration option, with the value set to the name of the COS DN. For example: `cos=COS_SupportAgent`

Checking the Destination Availability

SIP Server uses COS to analyze the availability of the destination in the following order:

1. SIP Server checks if an agent is logged in on the extension.
2. If the agent is logged in, the Agent Login COS is applied.

3. If the agent is not logged in on the device, or COS is not configured for the Agent Login, SIP Server checks if COS is configured for the device.
4. If COS is not configured for the Agent Login, SIP Server uses the options configured for the device.
5. If COS is configured for the device, SIP Server applies the COS.
6. If COS is not configured for the device, SIP Server checks if options `reject-call-incall` and `reject-call-notready` are specified for the device directly (without using COS).
7. If the `reject-call-incall` and `reject-call-notready` options are specified, SIP Server uses these options.
8. If none of the preceding apply, SIP Server considers the destination available.

Consolidated Error Response

SIP Server supports mapping a range of error messages from multiple sources to a single consistent error message that it sends to the network client. For example, in environments with several Genesys Voice Platform instances, where SIP Server sits in front of GVP (GVP Resource Manager and Genesys Media Server), SIP Server can translate its own error responses, as well as any error messages that it receives from these GVP instances, into a common error response that it sends to the network client—typically a `503 Service Unavailable` response. The network client (proxy or UAC) receiving the `503 Service Unavailable` message can then forward the original request to an alternate server—for example, to contact an alternate GVP instance to service the customer.

How It Works

Depending on which method of error detection is enabled, SIP Server consolidates error responses differently.

Passive Out-Of-Service Detection

1. SIP Server forwards a SIP `INVITE` request from the network client to GVP.
2. If GVP fails to respond, the `INVITE` request will timeout.
3. The `Trunk` DN representing the network client is configured to pass the error response to the client (`sip-busy-type` is set to `2`).
4. SIP Server suppresses the busy tone and marks the DN as out-of-service. With consolidated error response configured on either the DN or Application-level (`sip-error-conversion` is enabled), SIP Server translates its own error response to the configurable error response. For example, if

SIP Server does not receive a response to an INVITE request that it sends to GVP, SIP Server can be configured to translate the 603 Decline message that it generates into a 503 Service Unavailable message that it sends to the client.

Active Out-of-Service Detection

1. SIP Server is configured for active out-of-service detection; it periodically sends OPTIONS messages to GVP, testing its availability. If any particular GVP instance is unavailable, SIP Server will mark that DN (Trunk DN) as out-of-service.
2. On receiving an INVITE request from the network client, SIP Server checks if there are any active GVP DNs that can be used to service the call.
3. If no active DN is found, by default SIP Server normally sends a 404 Not Found error response. However, with consolidated error conversion enabled (sip-error-conversion is set to the desired error response), SIP Server translates the 404 Not Found message to the configured error response—typically 503 Service Unavailable—and sends that to the client network.

Feature Configuration

Table 42 describes how to configure a consolidated error response.

Table 42: Configuring Consolidated Error Response

Objective	Related Procedures and Actions
Configure error conversion for Active OOS Detection.	In the SIP Server Application object > Application Options tab > TServer section, configure the following option: • sip-error-conversion—Enter a comma-separated list of error-in/error-out pairs. For example, 404=503;603=503

Table 42: Configuring Consolidated Error Response (Continued)

Objective	Related Procedures and Actions
Configure error conversion for Passive OOS Detection.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following option: • `sip-error-conversion`—Enter the value `0=503`.
Configure for individual DNs.	• Suppress the busy tone for network client. In the `Trunk` DN for the network client, in the `TServer` section, configure the following option: `sip-busy-type`—Set this option to `2`. • Configure the consolidated error response. In either the DN that issues the error (for example, GVP `Trunk` DN) or in the SIP Server `Application` object, configure the following option: `sip-error-conversion`—Enter a comma-separated list of error-in/error-out pairs.

Control of SIP Response Code from within Routing Strategy

SIP Server supports the ability to configure how SIP response codes are sent from SIP Server to a routing strategy, and from SIP Server to the original caller:

- From SIP Server to routing strategy—The option `map-sip-errors` controls whether SIP Server sends a SIP Response code instead of a T-Library error code, in cases of an `EventError`.
- From SIP Server to original caller—When rejecting calls from a routing strategy, you can add the key-value pair `sip-status-code` to specify which SIP Response (in the `>=400` to `<700` range) that SIP Server will send back to the caller.

Feature Configuration

Table 43 describes how to enable this feature.

Table 43: Configuring SIP Error Response Codes

Objective	Key Procedures and Actions
Send SIP Response code to a routing strategy.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following option: • `map-sip-errors`—Set this option to `false`.
Specify SIP Response code to be sent back to caller.	1. Create a routing strategy that adds the `sip-status-code` to the `Extensions` attribute of rejected calls. 2. Configure the SIP Server Application with the following options: ◆ `ringing-on-route-point`—Set this option to `false`. ◆ `map-sip-errors`—Set this option to `true`. For details, see Procedure: Controlling SIP Response Codes from a Routing Strategy.

Procedure: Controlling SIP Response Codes from a Routing Strategy

Start of procedure

1. In Interaction Routing Designer, create an inbound routing strategy that determines under what criteria you want to accept or reject a call. For example, if no agents are currently available, the call is rejected.
2. In the rejection path of the routing strategy, use a function block to set the `Extensions` attribute with the key-value pair `sip-status-code`, where the value equals the SIP Response code you want to send. For example, to configure the Extension so that SIP Server sends a `486 Busy Here` message on call rejection, configure the `ExtensionAttach` function as follows:
   ```
   ExtensionAttach['{d}sip-status-code','486']
   ```
3. Connect the `ExtensionAttach` function to another function block that rejects the call. For example, use the `TRoute` function block with the route-type set to `reject`:
   ```
   TRoute["",RouteTypeReject,""]
   ```

4. When a call is initially placed, you may receive a `180 ringing` message in response. To suppress ringing, in the `TServer` section of the SIP Server `Application` object, set `ringing-on-route-point` to `false`.

   Note: Setting `ringing-on-route-point` to `false` suppresses automatic ringing sent by SIP Server, allowing you to specify a different response—for example, a busy signal—from the routing strategy

5. To configure SIP Server to use the specified response code (configured in Step 2) instead of the standard error code, in the `TServer` section of the SIP Server application, set `map-sip-errors` to `true`.

End of procedure

Customizing Music on Hold and in Queue

This section covers the following topics:

Playing Music to Calls on Hold

Genesys Media Server can play different media files for various contact center music on hold treatments. For example, Genesys Media Server plays a file that is associated with an agent DN if the agent places the call on hold.

Enabling Music on Hold

You can enable music on hold—as well as define the file to be played—using any of several, prioritized methods:

- From an agent DN—When the agent places the call on hold, the defined file will be played to the caller or the conference.
- From a client request—Key-value pairs in the `Extensions` attribute of the client request can specify the file that is to be played for this and subsequent calls. These extensions can be included in the following requests:
  - `THoldCall`
  - `TAlternateCall`
  - `TInitiateTransfer`
  - `TInitiateConference`

  Typically, the agent logged in to an agent desktop manually selects the music file that the agent wants to play to the caller when the caller is placed on hold.

- From the SIP Server Application—The default file that is configured on the application is played if no other configured filename is found.

Customizing Music on Hold

Starting with release 8.1.102.31, SIP Server lets you customize music for music-on-hold treatments. When the music-on-hold feature is activated, it applies to scenarios when the hold action is performed by an agent within the duration of the call explicitly (by `THoldCall`), or implicitly (by `TAlternateCall`, `TInitiateTransfer`, or `TInitiateConference`).

When custom music-on-hold is enabled on the Routing Point with the `music-on-hold` configuration option, or with the `music-on-hold` key in `AttributeExtensions` of `TRouteCall`, it remains attached (sticks) to the call until the call is released. If a `TRouteCall` request arrives with an empty value of the `music-on-hold` key in `AttributeExtensions`, the custom music-on-hold stickiness is removed from the call. If call routing fails, the custom music-on-hold setting is rolled back to the previous value.

The value of the `music-on-hold` option is attached to calls distributed via this Routing Point and used for playing the music-on-hold later.

When the `default-music` option is set for an Agent Login object, the setting applies only to a call established by the agent who activated the Hold operation.

Custom Music-on-hold in Conferences and Transfers

The custom music-on-hold setting is not applied to conferences and not shared when a consultation call is merged with the main call. However, the custom music-on-hold setting remains associated with the call, and if only two participants are left on the call, the custom music-on-hold setting will be applied if the caller is placed on hold. When a new party joins the conference, the custom music-on-hold setting is not applied.

For multi-site conferences support, SIP Servers must propagate full information about call parties. See “Providing Call Participant Info” on page 336 for information on how to enable it.

The custom music-on-hold setting is transferred with the call, which includes call routing, single-step transfers, two-step transfers, and call forwarding. In multi-site transfers, the ISCC connection is used.

If a call is transferred through a Routing Point that has a custom music-on-hold setting, the new music-on-hold setting will be applied to the next Hold scenario.

Media File Priority

The following settings determine the order of priority—from highest to lowest—in which a music file is played for a call on hold:

- The `music` key of `AttributeExtensions` in `THoldCall`, `TAlternateCall`, `TInitiateTransfer`, `TInitiateConference` requests, which initiate the Hold operation for a call.
- The `music-on-hold` key of `AttributeExtensions` in `TRouteCall` (if there are several `TRouteCall` requests for this call containing this key, the value from the last one is applied).
- The `music-on-hold` option on a Routing Point DN (if a call is passed through several Routing Points containing this option, the value from the last one is applied).
- The `default-music` option on an Agent Login level.
- The `default-music` option on an agent's Extension DN level.
- The `default-music` option on a SIP Server Application level.

Configuring Music on Hold

Table 44 describes the different configuration methods for playing music-on-hold media files.

Table 44: Configuring Music on Hold

Objective	Related Procedures or Actions
Configure client requests. Applies to: • `THoldCall` • `TAlternateCall` • `TInitiateTransfer` • `TInitiateConference`	Configure the T-Server client to include the following key-value pairs in the `Extensions` attribute of the client request, as required: • `music`—Specify the file name for music on hold for the call.
Configure an agent DN.	On the agent's Extension DN, in the `TServer` section, you can configure the following options as required: • `sip-enable-moh`—Set this option to `true` to enable music on hold for this DN. • `default-music`—Specify the music file that is to be played when this DN places the call on hold.
Configure an Agent Login.	On the Agent Login, in the `TServer` section, you can configure the following option as required: • `default-music`—Specify the music file that is to be played when this agent places the call on hold.

Table 44: Configuring Music on Hold (Continued)

Objective	Related Procedures or Actions
Configure a TRouteCall request.	You can specify the music-on-hold key in TRouteCall. If there are several TRouteCall requests for this call containing this key, the value from the last one is applied.)
Configure a Routing Point DN.	On the Routing Point DN, in the TServer section, you can configure the following option as required: • music-on-hold—Specify the music file that is to be played s attached to calls distributed via this Routing Point and used for playing the music-on-hold later. If a call is passed through several Routing Points containing this option, the value from the last one is applied.
Configure the SIP Server Application.	In the SIP Server Application, the TServer section, configure the following options: • sip-enable-moh—Set this option to true to enable default music on hold from the application. • default-music—Enter the directory and filename of the default music-on-hold file to be played in case no other settings are found. • music-in-conference-file—Enter the directory and filename of the application-wide default file that is to be played to the other participants (typically a silent audio file) when an agent places the conference on hold. • music-in-queue-file—Enter the directory and filename of the application-wide default file that is to be played when a call is waiting in an ACD Queue.

Feature Limitations

The following limitations apply to music-on-hold treatments:

- In multi-site deployments with the music-on-hold setting enabled in AttributeExtensions, the iscc-pass-extensions key in AttributeExtensions must not be set to a value of local, because it prevents extensions being passed through ISCC to a remote site.
- In Business Continuity (BC) deployments, the custom music-on-hold setting is propagated with a call transfer in DR-forward scenarios only if the Call Overflow feature is enabled. That is, the following SIP Server Application options must be set in the extrouter section:
  - cof-feature=true
  - default-network-call-id-matching=sip

Playing Music to Calls in Queue

Genesys Media Server can provide each queue on a single SIP Server with its own particular media file. Genesys Media Server plays a file that is associated with an `ACD Queue` DN while the call is in the queue.

Table 45: Configuring to Play Music to Calls in Queue

Objective	Related Procedures or Actions
Configure an ACD Queue.	In the `ACD Queue` DN > `Options` tab > `TServer` section, you can configure the following options as required: • `sip-enable-moh`—Set this option to `true` to enable music on hold for calls that are queued on this DN. • `default-music`—Specify the music file that is to be played when a call is queued on this DN. **Note:** In previous releases, the option `music-in-queue-file` was used to specify the file on the `ACD Queue` DN. This option has been deprecated at the DN-level.
Configure the SIP Server Application.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following option: • `music-in-queue-file`—Enter the directory and filename of the application-wide default file that is to be played when a call is waiting in an ACD Queue.

Customizing SIP Header Formats

SIP Server provides some flexibility for how different SIP headers are formed, depending on the needs of specific deployments.

Enabling Additional Parameters in Request-URI

SIP Server can be configured to include additional parameters in the `Request-URI`, in cases where the deployment requires it. For example, it can add the `user=phone` in the `Request-URI` of `INVITE` requests to a particular DN.

How It Works

In scenarios that require SIP Server to start a new INVITE dialog, SIP Server checks the configuration of the destination DN. If the option sip-uri-params is configured, SIP Server adds the additional parameters, as specified by this option, to the Request-URI of the INVITE request. No other SIP requests are affected. This feature is available for any DN type and affects any initial INVITE sent to that particular DN.

SIP Server supports this feature for both 1pcc and 3pcc calls.

SIP Server can apply this functionality for any of the following 3pcc requests:

- TMakeCall
- TMakePredictiveCall
- TRouteCall
- TRedirectCall
- TInitiateConference
- TInitiateTransfer
- TSingleStepTransfer
- TSingleStepConference

Feature Configuration

Table 46 describes how to enable additional parameters in the Request-URI, as required by your deployment.

Table 46: Enabling Additional Request-URI parameters

Objective	Related Procedures and Actions
Configure the Trunk DN.	In the outbound Trunk DN > Options tab > TServer section, set the sip-uri-params option to the value of the URI parameters you want to add. For example, user=phone SIP Server will include the new parameters in any outbound INVITE it sends through this Trunk.
Configure Voice over IP Service DNs.	In the Voice over IP Service DN (for example, a softswitch or music-on-hold DN) > Options tab > TServer section, set the sip-uri-params option to the value of the URI parameters you want to add. SIP Server will include the new parameters in any initial INVITE it sends to this service DN.

Table 46: Enabling Additional Request-URI parameters (Continued)

Objective	Related Procedures and Actions
Configure Extension DNs.	In the Extension DN > Options tab > TServer section, set the sip-uri-params option to the value of the URI parameters you want to add. SIP Server will include the new parameters in any initial INVITE it sends to this Extension DN. ***Note for IMS deployments:*** SIP Server ignores this option in IMS deployments. Use option ims-sip-params instead.

Enabling Server and User-Agent Headers

SIP Server supports inserting the Server header into all replies that it sends and the User-Agent header into all requests. For the Server header, you can configure this functionality at the Application level only. For the User-Agent header, you can configure this functionality on either the Application or the DN level. For non-INVITE dialogs, only the Application-level setting applies. You can also specify a User-Agent Extensions attribute by using the following T-Library requests:

- TMakeCall
- TMakePredictiveCall
- TSingleStepTransfer
- TSingleStepConference
- TInitiateTransfer (applies to consultation calls only)
- TInitiateConference (applies to consultation calls only)

Setting the User-Agent by using the Extensions Attribute overrides any values that you set in the configuration options.

Feature Configuration

Table 47 describes how to enable Server or User-Agent headers.

Table 47: Enabling Server and User-Agent Headers

Objective	Related Procedures and Actions
Enable the `Server` header.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following option: • `sip-server-info`—Enter a string or the special character `*`. The string can contain the placeholders `$VERSION$, $APP-NAME$` OR The special value `*` is equivalent to `Genesys SIP Server $VERSION$ ($APP-NAME$)`
Enable the `User-Agent` header.	You can enable the `User-Agent` header by using any of the following configurations, listed in order of priority: **1. AttributeExtensions:** Add the `User-Agent` key-value pair to the `Extensions` attribute of the supported T-Library request. **2. DN level:** In the `TServer` section of the individual DN, configure the following option: `sip-user-agent`—Enter a string or the special character `*`. The string can contain the placeholders `$VERSION$, $APP-NAME$` OR The special value `*` is equivalent to `Genesys SIP Server $VERSION$ ($APP-NAME$)` **Note**: The DN-level setting applies only to `INVITE` dialogs. **3. Application level:** Configure `sip-user-agent` in the SIP Server Application. **Note**: This setting applies to all dialogs, including `INVITE` dialogs.

Contact Header Handling Options

SIP Server supports two methods for handling the `Contact` header in SIP `REGISTER` requests:

- SIP Server formats the `Request-URI` in the `INVITE` request that it sends to an endpoint by using an exact match to the value of the URI that is obtained from the `Contact` header of the SIP `REGISTER` request. To use this method, set the `sip-preserve-contact` option to `true`.

- SIP Server disregards the user-name part of the URI obtained from the `Contact` header of the SIP `REGISTER` request. It replaces the user-name part with the DN name when formatting the `Request-URI` for the `INVITE` that it sends to an endpoint. To use this method, set the `sip-preserve-contact` option to the default value of `false`.

When multiple instances of SIP Server are deployed behind a session border control (SBC) device, it is possible for two SIP endpoints at different locations to have the same DN number. When it sends `REGISTER` requests to SIP Server on behalf of these endpoints, the SBC might add session information in the `Contact` header. If you set `sip-preserve-contact` to `true`, SIP Server will extract the cookie from the `REGISTER` message, then include it in the `INVITE` request that it sends to the endpoint through the SBC. The SBC then uses the session information to determine the correct endpoint.

Sample Call Flow Scenario

Figure 19 shows a multi-site scenario that involves an SBC, and in which agents and callers at different locations have the same DN number.

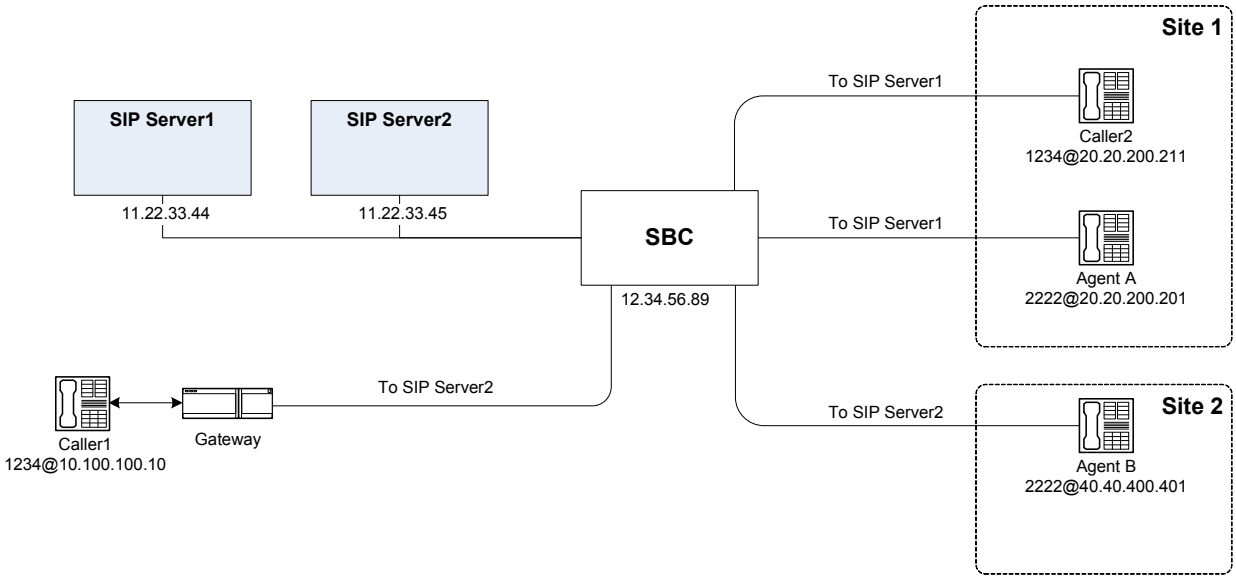

Figure 19: SIP Server with Cookie Persistence Enabled

In this case, when the SBC forwards a `REGISTER` request from an agent DN to SIP Server, the SBC translates the contact information from the original DN into contact information that represents the SBC.

For example, a `REGISTER` request that is sent by Agent A from Site—identified in the request to the SBC as `2222@20.20.200.201`—arrives at the SIP Server from the SBC with the contact info `2222-q3uq53j2hht32@12.34.56.89`. SBC overrides the `CONTACT` header and adds session information (a cookie) that uniquely identifies the agent (`-q3uq53j2hht32` for Agent A in the preceding example). If `sip-preserve-contact` is set to `true`, SIP Server preserves the

information in this cookie, then passes it back to the SBC when it is time to send an INVITE to the agent endpoint (INVITE sip:2222-q3uq53j2hht32@12.34.56.89:5060 SIP/2.0).

Feature Configuration

Table 48 describes how to enable the Contact header handling.

Table 48: Configuring Contact Header Handling

Objective	Related Procedures and Actions
Configure for all DNs.	In the SIP Server Application object > Application Options tab > TServer section: • To strip the Contact header—Set sip-preserve-contact to the default false. • To preserve the Contact header—Set sip-preserve-contact to true.
Configure for individual DNs.	In the SIP Server Switch > DNs > individual DN > Options tab > TServer section: • To strip the Contact header—Set sip-preserve-contact to the default false. • To preserve the Contact header—Set sip-preserve-contact to true. The DN-level setting takes precedence over the Application-level setting.

Diversion Header

For redirected calls, SIP Server supports the Diversion header—a SIP extension that provides the ability for the called party to identify where a particular call was diverted and why. SIP Server can process the Diversion header as follows:

- Forward the header to an inbound destination.
- Add a new header in cases of internal call redirection.

Forwarding the Diversion Header

SIP Server can forward a received Diversion header directly in the INVITE to a destination DN, or by mapping the header to a T-Library message that SIP Server sends to URS, making this information available to the routing strategy.

Forwarding to a Destination DN

If an incoming INVITE request includes the Diversion header, SIP Server can include this header in the subsequent outgoing INVITE that it sends to the destination DN. This behavior is specified by the sip-proxy-headers-enabled

option. This option takes precedence over the `sip-enable-diversion` option. If `sip-proxy-headers-enabled` is disabled (set to the non-default `false`), all SIP headers will not be forwarded (it does not matter what the setting is for `sip-enable-diversion`).

Mapping to T-Library Messages

If the incoming `INVITE` request to a `Routing Point` DN contains the `Diversion` header, SIP Server can map this header to UserData in the T-Library events `EventQueued` or `EventRouteRequest`. This makes the information available to the routing strategy for intelligent use. To enable this mapping, you must create a new `userdata-n` option in the `INVITE` section of the SIP Server application.

Sample mapping of `Diversion` header from incoming `INVITE` to subsequent `TEvent` is as follows:

INVITE to Routing Point

```
INVITE sip:5000@172.24.129.75:5060 SIP/2.0
From: <sip:21001@172.24.129.75:21001>;tag=0396021E-DB9C-488E-9EC6-
68E6C69263CF-1
To: sip:5000@172.24.129.75:5060
Call-ID: 7D64E88C-345D-478E-B0CB-FC628D4D9BB1-1@172.24.129.75
CSeq: 1 INVITE
Content-Length: 147
Content-Type: application/sdp
Via: SIP/2.0/UDP 127.0.0.1:21001;branch=z9hG4bKB67C60B1-6106-4DD2-
904C-AD8F8D70D2A6-1
Contact: <sip:172.24.129.75:21001>
Diversion: <sip:7103@172.24.129.75:5060>;reason=unconditional
```

EventQueued Message

```
MessageEventQueued
  AttributeEventSequenceNumber000000000000008e
  ...
  AttributeDNIS'5000'
  AttributeUserData[74]00 01 00 00..
  'Diversion''<sip:7103@172.24.129.75:5060>;reason=unconditional'
  AttributeCallUUID'U4BIG099V94A72Q2MIK0SUFLK4000000'
```

For more detailed information about mapping headers to T-Library messages, see “Mapping SIP Headers and SDP Messages” on page 261.

Adding the Diversion Header

SIP Server can also add a new `Diversion` header in the case of certain internal call diversions—like alternate routing to a default DN—or if asked to by the URS routing strategy.

Some situations where SIP Server can add a new `Diversion` header to an `INVITE` are:

- Dial plan—When unattended calls are diverted to alternate DNs specified using additional parameters in the dial plan.
- Alternate routing—Calls can be diverted to alternate DNs in response to a number of different call scenarios.
- No-Answer Supervision—Based on the availability of supervised agents or `Extension` DNs, SIP Server can divert the call to a sequence of overflow destinations.
- Call redirection—When the agent redirects a call to other DNs. Redirection can be either 1pcc or 3pcc.
- Call forwarding—When the call is forwarded to another DN (1pcc and 3pcc).
- Mapping from T-Library Request—The routing strategy can be designed to include Diversion parameters in a T-Library request, which SIP Server then maps to the resulting `INVITE` request or `302 Moved Temporarily` message.

Mapping from T-Library Request

You can design the routing strategy to include Diversion-related parameters in the `Extensions` attribute in T-Library requests. SIP Server can then map these parameters to a `Diversion` header in the resulting `INVITE` or `302 Moved Temporarily`. In this case, the routing strategy must be designed to include the Diversion key-value pair in the `SIP_Headers` extension of the T-Library request.

The routing strategy can either create the Diversion header based on other headers mapped from the incoming `INVITE` request, or it can be configured directly.

Note: The Diversion header should follow the syntax described in RFC 5806 “Diversion Indication in SIP”.

Sample mapping from `TRouteCall` to SIP `INVITE` is as follows:

TRouteCall

```
MessageRequestRouteCall
AttributeThisDN'5000'
AttributeConnID007001e08c992001
AttributeOtherDN'7102'
AttributeLocation''
AttributeExtensions[125]00 03 00 00..
   'SIP_HEADERS''Diversion'
   'Diversion''<sip:7103@172.24.129.75:5060>;reason=unconditional'
AttributeDNIS''
AttributeRouteType1(RouteTypeDefault)
```

INVITE

```
INVITE sip:7102@172.24.129.75:7102 SIP/2.0
From: sip:21001@172.24.129.75:21001;tag=9A8776B3-0A32-4089-83AE-
7CE09D79F7C9-2
To: <sip:5000@172.24.129.75:5060>
...
Contact: <sip:21001@172.24.129.75:5060>
Diversion: <sip:7103@172.24.129.75:5060>;reason=unconditional
```

For more information about mapping from T-Library to SIP requests, see "Using SIP_HEADERS and SIP_REQUEST_PARAMETERS" on page 273.

Feature Configuration

Table 49 describes how to enable processing of the `Diversion` header.

Table 49: Configuring Diversion Header Support

Objective	Related Procedures and Actions
Forwarding the Diversion Header	
Forward all Diversion headers.	In the SIP Server `Application` object, make sure that `sip-proxy-headers-enabled` is set to the default value of `true`. By default, SIP Server forwards all headers, including `Diversion`, in the subsequent `INVITE`.
Map to T-Library request.	In the SIP Server `Application` object, configure the following: • In the `Application Options` tab, create a new section called `INVITE`. • In this section, create a new option `userdata-n`. • For the value of this option, enter `Diversion`. SIP Server will include the `Diversion` parameter in the `UserData` Attribute of the `EventRouteRequest` or `EventQueued`.

Table 49: Configuring Diversion Header Support (Continued)

Objective	Related Procedures and Actions
Adding the Diversion Header	
Enable for dial plan.	Calls can be diverted based on the routing outcome of a dialing rule. To define where these calls will be diverted, the dialing rule uses the following parameters: • `ontimeout` • `onbusy` • `ondnd` To include the `Diversion` header, configure the destination DN defined in these parameters as follows: • In the `TServer` section, set `sip-enable-diversion` to `true`. The header will include a reason that matches the dial-plan parameter used: • `no-answer` • `user-busy` • `do-not-disturb` For a full description of how to configure dial plans, see "Dial Plan" on page 195.
Enable for No-Answer Supervision.	With this feature, calls are diverted to a sequence of overflow destinations based on agent timeout. To add the `Diversion` header in `INVITE`s sent to the overflow destination: 1. Check that the overflow destination DN is configured. See "No-Answer Supervision" on page 302. 2. In the overflow DN > `Options` tab > `TServer` section, set `sip-enable-diversion` to `true`.
Enable for Alternate Routing.	Alternate Routing uses a number of different alternate DNs to handle internally redirected calls. Set `sip-enable-diversion` to `true` in any of the alternate DNs as defined in the following option: • `default-dn` For more information about configuring these alternate DNs, see "Alternate Routing" on page 106.

Table 49: Configuring Diversion Header Support (Continued)

Objective	Related Procedures and Actions
Map from T-Library request.	In the T-Library client or URS routing strategy, configure the request to include the following: • SIP_HEADERS—Add 'Diversion' to the list of custom SIP headers to be added to the INVITE. • Define the header as follows: 'diversion' '<use syntax as described in RFC 5806>'
Enable for Call Redirection.	Calls can be redirected or forwarded to another DN. Set sip-enable-diversion to true in the destination target DN.

Early Media Private Header

SIP Server supports passing the P-Early-Media header for inbound calls, as described in RFC 5009 "Private Header (P-Header) Extension to the Session Initiation Protocol (SIP) for Authorization of Early Media". This header can be used to control the flow of media in the early dialog state. SIP Server supports the P-Early-Media header for inbound calls only, and only for the following messages: 18X, 200, INVITE, PRACK, and UPDATE.

The P-Early-Media header is passed only when both the calling and destination domains are configured with enforce-trusted set to true.

This functionality is only applicable if the calling side supports early media dialogs. Early media must be configured on the Trunk DN (sip-early-dialog-mode is set to 1).

Feature Configuration

Table 50 describes how to enable the passing of the P-Early-Media header.

Table 50: Enabling P-Early-Media Header

Objective	Related Procedures and Actions
1. Verify calling side prerequisites.	The calling side must support the option tag 100rel and early media.
2. Configure the inbound Trunk.	In the inbound Trunk DN on which the calls requiring P-Early-Media header will arrive, configure the following options in the TServer section: • sip-enable-100rel—Set this option to true. • enforce-trusted—Ensure this option is set to the default true.

Private Headers

SIP Server may include private, Genesys-proprietary custom headers in the SIP messages that it sends in certain call scenarios—for example, in multi-site scenarios where the call passes through several instances of SIP Server. These headers are identified by the prefix `X-` and include the following:

- `X-Genesys-PartyInfo`—Required for communication between instances of SIP Server in multi-site deployments.
- `X-ISCC-Id`—Required for communication between instances of SIP Server in multi-site deployments.
- `X-ISCC-CofId`—Required for ISCC/COF call matching between instances of SIP Server in multi-site deployments.
- `X-Genesys-CallUUID`—Required for communication between instances of SIP Server in multi-site deployments.
- `X-Genesys-<user_data>`—When you are integrating with GVP, Genesys recommends that you configure the `userdata-map-trans-prefix` option to use the prefix `X-Genesys-` in the custom headers that are used to map user data. In this case, you can expect to find `X-Genesys-` headers in the `INVITE` messages that SIP Server sends to GVP.

Forwarding Custom Headers

SIP Server can pass custom SIP headers from a `REFER` request to an outgoing `INVITE` or `REFER` request. The Application-level configuration option, `sip-pass-refer-headers`, must be configured to enable this functionality.

Table 51 describes how to configure custom header forwarding.

Table 51: Enabling Custom Header Forwarding

Objective	Related Procedures and Actions
Configure the SIP Server Application.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following: • `sip-pass-refer-headers`—Enter the exact name of the SIP headers to be forwarded in a comma-separated list. SIP Server will forward these headers from the `REFER` (if included) to the outgoing `INVITE` or `REFER`. You can use the asterisk (*) as a wildcard for multiple headers with the same prefix.

Filtering Custom Headers

SIP Server filters out Genesys internal SIP headers from `TRouteCall` and `TMakePredictiveCall` requests when generating an outgoing `INVITE` or `REFER` request to a media gateway, unless otherwise specified. The DN-level configuration option, `enable-extension-headers`, is used to define which call request will include the custom headers from the `Extensions` attribute. For call requests not in this list, custom headers will be filtered out from the outgoing `INVITE` or `REFER`. This functionality applies to the following types of call requests:

- `TRouteCall`
- `TMakePredictiveCall`

Table 52 describes how to configure custom header filtering.

Table 52: Enabling Custom Header Filtering

Objective	Related Procedures and Actions
Configure the `Trunk` DN.	In the outbound `Trunk` DN > `Options` tab >`TServer` section, configure the following option: • `enable-extension-headers`—Enter the type of call request that will include custom headers: `predictive`, `routing` (one, both, or `none`). If the request is not in this list, custom headers will be filtered.

Dial Plan

The dial plan feature allows you to define the rules that SIP Server applies to the dialed digits that it receives from an endpoint or T-Library request. These rules enable SIP Server to transform the received digits into the actual digits that it uses to make the call.

SIP Server also supports the Dial Plan feature implemented in Genesys SIP Feature Server. The dial plan can be configured either on the SIP Server side or on the SIP Feature Server side. Refer to the SIP Feature Server 8.1.2 documentation for details.

Notes:
- `TMakePredictiveCall` is not processed by Dial Plans provided by SIP Feature Server.
- The SIP Feature Server Dial Plan is not applied to a scenario when an inbound call arrives at a Routing Point.

Introduced in SIP Server 8.1.102.22

SIP Server offers the option to use SIP Feature Server as an "external dial plan" as an alternative to the internal SIP Server dial plan. Each choice offers distinct advantages to consider when choosing which dial plan to use. (Note that dial plans may not be combined.)

SIP Feature Server Dial Plan Highlights

- User-based calling preferences for Call Waiting and Call Forwarding (including Find-Me-Follow-Me)
- Flexible rules with pattern matching logic for choosing a trunk for outgoing calls
- Enhanced support for deployments where voicemail mailboxes are assigned to users (but not to DNs)

See "Using SIP Feature Server Dial Plan" on page 212 for configuration details.

SIP Server Dial Plan Highlights

- Many supported parameters for advanced dial-plan rules, such as `onbusy`, `type`, `calltype`, `clir`, and more
- Native support by SIP Server (smaller footprint, less complexity if Feature Server is not required for the deployment)

See "Using SIP Server Dial Plan" on page 206 for configuration details.

Dial Plan Configuration Overview

Dial plans are configured as a set of rules on a `Voice over IP Service` DN with a `service-type` of `dial-plan`. You then assign the dial plan to any of the following objects, listed in order of priority:

1. Agent Login—Applies to calls made by a caller logged in under this Agent Login ID.
2. DN-level—Applies to calls made from a DN (where Agent Login dial-plan is undefined) or for inbound calls if the dial-plan is assigned to the Trunk DN.
3. Application-level—Applies to all calls (where no Agent Login or DN dial-plan is defined).

You can also create group dial-plan DNs, where a single `Voice over IP Service` DN integrates several underlying dial plans within a single assignable DN.

Note: A dial-plan rule configured on a Routing Point is applied only for calls that are initiated through a `TMakePredictiveCall` request on behalf of that Routing Point.

Introduced in SIP Server 8.1.102.22

SIP Server offers additional control over how a dial plan is applied to the destination of `TRouteCall` and/or to multi-site (ISCC) calls that are routed through an External Routing Point with two configuration options:

- The `rp-use-dial-plan` configuration option changes the default behavior of the dial plan to any one of the following:
  - SIP Server does not apply any dial plan.
  - SIP Server applies only the digit translation to a dial plan target.
  - SIP Server applies the digit translation and forwarding rules to a dial plan target.

  The `rp-use-dial-plan` option applies to both SIP Server and SIP Feature Server dial plans. If the `UseDialPlan` key-value pair is present in `AttributeExtensions` of `TRouteCall`, then it takes priority over the `rp-use-dial-plan` option.
- The `enable-iscc-dial-plan` option enables SIP Server to apply the dial plan to the target destination when a call is routed from an External Routing Point (`cast-type=route-notoken`) to a DN at the destination site.

Note: If out-rule functionality (deprecated) is configured on the Class of Service (COS) DN, then the output of the out-rule will be used as the "dialed digits" input to the dial-plan rule. For more information about out-rule functionality on COS DNs, see "Class of Service" on page 173.

Dial Plan Call Flow

When SIP Server receives an `INVITE` message for a 1pcc call or a T-Library request for 3pcc operation, SIP Server checks to see if a dial-plan is assigned to the DN that initiated the call. If it finds that a dial-plan is assigned to the DN, SIP Server tries to match the dialed digits provided in the request to any of the patterns configured in the dial-plan. If a match is found, SIP Server can perform digit translation as specified in the dial-plan rule. Depending on what additional parameters are included in the dial-plan rule, SIP Server can perform other actions for the call—for example, to provide alternate routing if the destination is unavailable. If no pattern match is found, then the call proceeds as dialed (no modifications are made to the dialed digits).

When SIP Server applies a dial plan to a call, it includes the `original-dialplan-digits` extension key containing the original destination number (the dial plan input) before the dial plan is applied. If a call scenario contains multiple consecutive steps (for example, an inbound call to a Routing Point, routing to an agent, and a single-step transfer to the other agent), then an original dialed number is defined for each call step. For example, one dialed number is defined for an inbound call, another for routing, and a third one for a single-step transfer.

If a destination DN is a Routing Point, then the `original-dialplan-digits` extension key is passed in `EventQueued` and `EventRouteRequest` messages. If a call is made to the `ACD Position` DN, then a new extension key is added to `EventQueued`. If a call is made to the `Extension` DN, then a new extension key is added to `EventRinging`.

If the dial plan is not applied to the call, `original-dialplan-digits` will not be added.

If the initiating DN is assigned a dial-plan, SIP Server can apply dial-plan functionality to any of the following 3pcc requests:

- `TMakeCall`
- `TMakePredictiveCall`
- `TInitiateTransfer`
- `TInitiateConference`
- `TSingleStepTransfer`
- `TSingleStepConference`
- `TRedirectCall`
- `TCompleteTransfer`
- `TCompleteConference`
- `TRouteCall` (only if the `UseDialPlan` key extension is used)

SIP Server includes the resulting digits when the dial plan is applied as the username part of the `From` header in the `INVITE` message sent to the origination device. This behavior can be changed by setting the `sip-3pcc-from-pass-through` option to `true`.

SIP Server will also apply dial-plan logic to 1pcc `INVITE`, `REFER`, and `302 (Moved Temporarily)` operations.

Removal Overdialed Digits From DNIS

SIP Server provides the ability for internal and inbound calls coming to a Routing Point to remove overdialed digits from `DNIS` when the `dnis-max-length` dial-plan rule parameter is specified. Overdialed digits are added to the `DNIS_OVER` key of `AttributeExtensions` in T-Library events `EventQueued` and `EventRouteRequest`.

Outbound and transfer call flows are not supported for this feature.

Example 1

```
dial-plan-rule: 0800XXXXXXX!=>1000; dnis-max-length=11
```

Called number: 080012345670123

Then `EventQueued` and `EventRouteRequest` will contain the following attribute values:

```
AttibuteThisDN: 1000
```

```
AttibuteDNIS:   08001234567
AttibuteExtensions 'DNIS_OVER': 0123
```

Example 2

The `dial-plan-rule` parameter does not modify the DNIS, except when the `dnis-max-length` is set.

```
dial-plan-rule: 5566=>1111
```

Called number: 5566

Then attributes `ThisDN` and `DNIS` in T-Library events will contain the following values:

```
AttributeThisDN: 1111
AttributeDNIS: 5566
```

`EventQueued` and `EventRouteRequest` will not contain the `DNIS_OVER` in AttributeExtensions.

Example 3

```
dial-plan-rule: 5566=>1111;dnis-max-length=2
```

Called number: 5566

Then attributes `ThisDN` and `DNIS` in T-Library events will contain the following values:

```
AttributeThisDN: 1111
AttributeDNIS:55
```

`EventQueued` and `EventRouteRequest` will contain the following attribute value:

```
AttibuteExtensions 'DNIS_OVER':66
```

The Dial-Plan Rule

When configuring a dial-plan rule (`dial-plan-rule-<n>`) in the dial-plan, you must use the following format:

```
pattern => digits;param1=value1;param2=value2 # comment
```

Pattern Matching

SIP Server tries to match the `pattern` in this string to the actual digits dialed, using the Asterisk format with the syntax described in Table 53.

Table 53: Asterisk Dial Plan Syntax for Pattern Matching

Special Character	Pattern Matching
X	Matches any single digit from 0-9.
Z	Matches any single digit from 1-9.
N	Matches any single digit from 2-9.
[ ]	Matches any of the digits found inside the square brackets. For example, using the special characters [12345], SIP Server can match any of the digits 1, 2, 3, 4, or 5.
[X-Y]	(hyphen inside square brackets) Matches a range of digits. For example, `[125-8]` matches any of the digits 1, 2, 5, 6, 7, 8.
.	(period) Wildcard match. Matches one or more characters.
!	(exclamation point) Wildcard match. Matches 0 or more characters.

Pattern Examples

Some examples of how these special characters can be used to match the dialed digits are as follows:

- `9NXXXXXXXXX`—Matches any 11-digit number beginning with 9, where the second digit is between 2 and 9.
- `9[54]10XXXXXX`—Matches any 11-digit number beginning with either 9510 or 9410.
- `[45]XXX`—Matches any 4-digit number beginning with either 4 or 5.

Multiple Patterns

If multiple patterns match a dialed-number, SIP Server selects the pattern with the most specific match (in other words, the match with the least wildcard uses) from left to right. For example, if `5111` is dialed then the pattern `5XXX` would make the match instead of `XXXX`.

Digit Translation

After matching the number dialed to the `pattern` defined in the dial-plan rule, the `digits` parameter tells SIP Server what number to use when it makes the call. These digits can be any alphanumeric string, terminated with a semicolon.

This parameter can also use the {DIGITS} variable for flexibility in defining the digits to be dialed.

{DIGITS} Variable

The digits variable in the dial-plan rule must take one of the following formats:

${DIGITS}, ${DIGITS:x}, ${DIGITS:x:y}

where,

DIGITS	Defines the actual digits dialed from the endpoint.
X	Defines the starting position of the variable, identified by the character position in the digit string. In this case, 0 represents the first character in the string (starting from the left). This value can be negative, which indicates a character position starting from the right instead of left. For example, -1 indicates the right-most character. Default for this parameter is 0.
Y	Specifies the number of characters to be included, starting from the position defined by X. By default, all characters in the string are included.

Translation Examples

If the number 96501235678 is dialed, some examples of how the {DIGITS} variable can translate these digits are as follows:

- ${DIGITS}—Translates to 96501235678.
- ${DIGITS:1}—Translates to 6501235678.
- ${DIGITS:-4:4}—Translates to 5678.
- ${DIGITS:0:4}—Translates to 9650.

Sample Dial Plan Rules

Some sample values for the dial-plan-rule-<n> option, configured in the dial-plan DN, are as follows:

```
5XXX=>4351707${DIGITS} # This rule matches any 4-digit number
starting with 5 and translates it to the number 43517075XXX
5002=>43517075002 # This rule matches the dialed number 5002 and
translates it to the number 43517075002
```

Dial Plan Parameters

Table 54 describes additional parameters you can use to define the behavior of the dial-plan rule.

Table 54: Dial-Plan Rule Parameters

Parameter	Value	Description
type	digits, agent, reject	Defines the meaning of the digits in the dialing rule. Set the value for this parameter to any of the following: • `digits` (default)—SIP Server interprets the digits as regular dialed digits. • `agent`—SIP Server interprets the digits as an agent Extension DN, an ACD Position DN, or an Agent Login ID. If calling an Agent Login ID, SIP Server directs the call to the DN on which the agent is logged in. **Notes:** ◦ Genesys recommends using different identifiers for Agent Login IDs and the agent `Extension` DNs when using this option, otherwise SIP Server will direct the call to the extension even if the agent is logged out. ◦ 3pcc calls to an `Agent Login` ID will be converted to the Extension that the agent is logged into *before* being processed by the dial-plan rules. • `reject`—SIP Server will reject this call, sending a SIP error code (`400` to `699`) to the caller. This code is provided in the `[digits]` part of the dial-plan rule. For example, in the dial-plan rule `[pattern]=>486;type=reject`, `486` is the error code. If `0` or any other non-compliant code is provided, SIP Server will use the default code `403 (forbidden)`.
calltype	internal, inbound, outbound	Defines the `AttributeCallType` to be used in T-Library events for the new call. Changes take effect for new non-consultation calls. Set the value for this parameter to any of the following: • `internal`—SIP Server will use `CallTypeInternal` as the attribute. • `inbound`—SIP Server will use `CallTypeInbound` as the attribute. • `outbound`—SIP Server will use `CallTypeOutbound` as the attribute.
clir	on, off	Enables or disables Calling Line Identification Restriction. If set to `on`, SIP Server does not display caller ID. If set to `off` (default), SIP Server displays calling party ID, if it is available.

Table 54: Dial-Plan Rule Parameters (Continued)

Parameter	Value	Description
timeout	integer	Specifies the length of time, in seconds, that SIP Server waits for the agent to answer a ringing call. After this timeout period, SIP Server forwards the call to the destination specified in the `ontimeout` parameter. The options `timeout` and `ontimeout` must be defined together. **Note**: If call forwarding is initiated from the endpoint, or by using the `no-answer-timeout` option, SIP Server will use whichever forwarding has the shortest timeout setting.
ontimeout	string	Specifies the destination where SIP Server will forward the call after the timeout period elapses. The timeout parameter must be present for this option to have an effect. You can use the `${DIGITS}` variable to configure `ontimeout`.
onbusy	string	Specifies the alternate destination where SIP Server will forward calls made to a busy destination. You can use the `${DIGITS}` variable to configure `onbusy`.
ondnd	string	Specifies the alternate destination where SIP Server will forward calls made to a destination that is currently set to `do-not-disturb`, or that returns a DND SIP error code (603). You can use the `${DIGITS}` variable to configure `ondnd`.
onunreach	string	Specifies the destination where SIP Server will forward the call after the unreachable period (`unreach-timeout` parameter) elapses. If the `unreach-timeout` parameter is not present, then the unreachable period is set by the option `sip-invite-timeout`. The `onunreach` parameter takes precedence over the option `no-response-dn`. You can use the `${DIGITS}` variable to configure `onunreach`.
onnotreg	string	Specifies the alternate destination where SIP Server will forward calls made to a destination that has no SIP registration. You can use the `${DIGITS}` variable to configure `onnotreg`.

Table 54: Dial-Plan Rule Parameters (Continued)

Parameter	Value	Description
unreach-timeout	integer	Specifies the length of time, in seconds, that SIP Server waits for a response (provisional or `200 OK`) from an endpoint to an `INVITE` request. After this timeout period elapses, SIP Server forwards the call to the destination specified in the `onunreach` parameter. If `unreach-timeout` is not specified, then SIP Server will use the value in the option `sip-invite-timeout` to define this timeout. If `onunreach` is not specified, then on timeout SIP Server will either forward the call to the DN specified in the option `no-response-dn`, or if this option is not defined, SIP Server will terminate the call.
privilege	integer	Specifies the privilege value for this dial-plan rule. Minimum integer is `0` (default), maximum is `10`. Additional configuration as follows: • To allow the calling DN to use this rule—add the value to the `privilege-level` list in the `Class of Service` DN assigned to the calling device. • To bar the calling DN from using this rule—do not add the value to the `privilege-level` list. • To allow all calling DNs to use this rule—do not configure the privilege parameter in the dial-plan rule. **Note**: The default setting of 0 allows any class of service to use this rule. For more information, see "About Privilege Levels".
[space]#	string	Indicates the start of the comments section. Any data after the `#` is ignored. You must include a space before the `#`, otherwise SIP Server interprets it as a regular character.
dnis-max-length	integer	Valid values: 1-22. Defines the maximum length of DNIS in the dial-plan rule. The digits that are in position past the specified length are considered overdialed and removed from DNIS. Overdialed digits are included as a value of the `DNIS_OVER` key of `AttributeExtensions` in `EventQueued` and `EventRouteRequest`. Any invalid value disables this feature. See "Removal Overdialed Digits From DNIS" on page 198.

About Privilege Levels

Privilege levels are used to define which dial-plan rules are available for calls made by the caller associated with a particular COS.

For example, SIP Server can block a lobby phone from making national or international calls.

To configure privilege levels, you first define the privilege level in the dial-plan rule, then add the definition to a list of accepted privilege levels in the class of service DN. If the dialed digits of a new call match a dial-plan rule with a privilege defined, then that privilege must be configured in the COS assigned to the caller. If it is not, the call is rejected.

You can define two different privilege-level options in COS:

- `fwd-privilege-level`—This option applies to all operations that forward an existing call to a new party (transfer, conference, redirect, `302` response).
- `privilege-level`—This option applies to operations that initiate a new call (1pcc and 3pcc `TMakeCall`, `TMakePredictiveCall`, `TInitiateConference`, `TInitiateTransfer`), or for all operations if the COS does not also define the `fwd-privilege-level` option.

You then assign the COS to any of the following objects, listed in order of priority:

1. Agent Login—Applies to calls made by a caller logged in under this Agent Login ID.
2. DN-level—Applies to calls made from this DN (where Agent Login COS is undefined).
3. Application-level—Applies to all calls (where no Agent Login or DN COS is defined).

Note: The default dial-plan rule privilege setting (`0`) is allowed by all COS DNs and does not need to be explicitly configured.

Feature Configuration

This section describes how to configure a Dial Plan feature depending on whether you use SIP Server or SIP Feature Server.

- See “Using SIP Server Dial Plan” on page 206.
- See “Using SIP Feature Server Dial Plan” on page 212.

Using SIP Server Dial Plan

Table 55 describes how to configure a SIP Server dial plan.

Table 55: Configuring a SIP Server Dial Plan

Objective	Related Procedure and Key Actions
1. Configure a dial-plan DN.	1. Create a `Voice over IP Service` DN. 2. In the `Options` tab > `TServer` section, configure the following options: • `service-type = dial-plan` • `dial-plan-rule-<n>` For details, see Procedure: Configuring a Dial-Plan DN, on page 207.
2. Configure a class of service DN.	1. Create a `Voice over IP Service` DN. 2. In the `Options` tab > `TServer` section, configure the following options: • `service-type = cos` • `privilege-level` • `fwd-privilege-level` For details, see Procedure: Configuring Class of Service for a Dial Plan, on page 208. **Note:** COS DN is only required if privilege parameters are used in the applicable dial-plan rules. For more information about configuring the COS DN (for example, the inbound ring-through rules), see "Class of Service" on page 173.
3. Assign the service DNs.	**Global Configuration** • In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following options: • `cos` • `dial-plan` **Individual Configuration** • In the DN or Agent Login object > `Options` tab > `TServer` section, configure the following options: • `cos` • `dial-plan` For detailed procedures, see the following: • Procedure: Assigning the dial plan to a device, on page 210 • Procedure: Assigning COS to a device, on page 211 • Procedure: Assigning the dial plan and COS globally, on page 212

Table 55: Configuring a SIP Server Dial Plan (Continued)

Objective	Related Procedure and Key Actions
Additional Special Configuration	
Include other dial plans in a single dial-plan DN.	You can include multiple other dial plans under a single dial-plan DN. • In the dial-plan `Voice over IP Service` DN > `Options` tab > `TServer` section, configure the following option: ◆ `include-dial-plan-<n>` See Procedure: Including additional dial plans, on page 209.
To apply a dial plan to the destination of `TRouteCall`.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following option: • `rp-use-dial-plan`
To apply a dial plan to multi-site (ISCC) calls that are routed through an External Routing Point.	In the SIP Server `Application` object > `TServer` section, set this option to `true`: • `enable-iscc-dial-plan`

Procedure: Configuring a Dial-Plan DN

Start of procedure

1. Under the SIP Server `Switch` object, create a `Voice over IP Service` DN.
2. In the dial-plan `DN` object > `Options` tab > the `TServer` section, set the configuration option `service-type` to `dial-plan`.
3. Specify the dial plan rules.

   For each rule, add a new configuration option `dial-plan-rule-<n>`, using the following format:

Name `dial-plan-rule-<n>`

Value
```
pattern => digits;param1=value1;param2=value2;... # comment
```
For example:
```
dial-plan-rule-1 = 9NXXXXXX=>9${DIGITS:2} # local out-dial
```

Note: For detailed information about the syntax used in the different parts of the dial plan rule, see the following:
- `pattern`—See "Pattern Matching" on page 200.
- `digits`—See "Digit Translation" on page 200.
- `param1=value1`, and so on—See "Dial Plan Parameters" on page 201.

Figure 20 illustrates a sample configuration for a dial-plan DN with dial-plan rules specified.

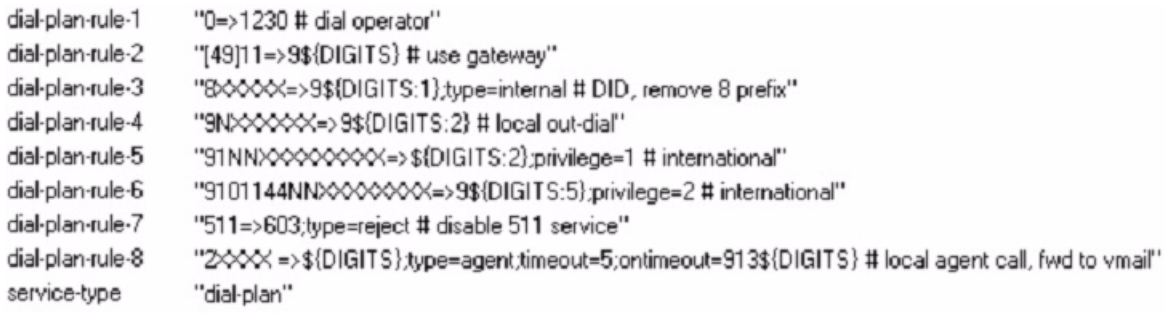

Figure 20: Configuring the Dial-Plan DN: Sample Configuration

End of procedure

Next Steps

- If you are including privilege-level Class of Service configuration, continue to Procedure: Configuring Class of Service for a Dial Plan.
- To include additional dial plans in this DN, see Procedure: Including additional dial plans, on page 209.
- To assign the dial plan to a particular device, see Procedure: Assigning the dial plan to a device, on page 210.
- To assign the dial plan globally for all calls (unless otherwise specified on the DN or Agent Login-level), see Procedure: Assigning the dial plan and COS globally, on page 212.

Procedure: Configuring Class of Service for a Dial Plan

Start of procedure

1. Under the SIP Server `Switch` object, create a COS DN with a type of `Voice Over IP Service`.
2. On the COS DN object, in the `TServer` section, set the configuration option `service-type` to `cos`.
3. In the `TServer` section, configure the privilege levels that will be accessible by this class of service:
   - `privilege-level`—Set this option to a list of integers that define which dial-plan rules are allowed for outgoing calls made by the caller associated with this COS DN.

- `fwd-privilege-level`—Set this option to a list of integers that define which dial-plan rules are allowed for transfer/redirect/conference operations.

  Tip: The privilege levels defined here must match the `privilege` parameter in the `dial-plan-rule-<n>` option, otherwise the call will be blocked.

End of procedure

Next Steps

- To assign the Class of Service to a particular device, see Procedure: Assigning COS to a device, on page 211.
- To assign the Class of Service globally for all calls (unless otherwise specified at the DN or Agent Login-level), see Procedure: Assigning the dial plan and COS globally, on page 212.

Procedure: Including additional dial plans

Purpose: To include additional dial plans in a defined dial-plan DN. This lets you to group several dial plans into a single DN, which you can then assign as required.

Prerequisites

- A configured dial-plan DN with dial-plan rules (`Voice over IP Service` DN with `service-type` set to `dial-plan`). See Procedure: Configuring a Dial-Plan DN, on page 207.

  Note: The dial-plan-rule will be selected purely on the number of specific digits matched—no preference is given to any rules in this dial-plan or the included dial-plan.

Start of procedure

1. Under the SIP Server `Switch` object, open a configured dial-plan `Voice Over IP Service` DN.
2. In the `TServer` section, set the `include-dial-plan-<n>` configuration option to the name of the underlying dial plan DN that you want added to this dial-plan DN.
3. Add new instances of the `include-dial-plan-<n>` option for every dial-plan DN to be included in this DN.

   Figure 21 shows a sample dial-plan DN that includes other dial plans.

```
dial-plan-rule-1          "0=>1230 # dial operator"
dial-plan-rule-2          "[49]11=>9${DIGITS} # use gateway"
dial-plan-rule-3          "8XXXXX=>9${DIGITS:1};type=internal # DID, remove 8 prefix"
dial-plan-rule-4          "9NXXXXXX=>9${DIGITS:2} # local out-dial"
include-dial-plan-3       "barred-calls"
include-dial-plan-2       "long-distance-calls"
include-dial-plan-1       "local-calls"
```

Figure 21: Sample Dial-Plan DN That Includes Other Dial Plans

End of procedure

Next Steps

- If this dial-plan DN is not already assigned to the required devices, continue to Procedure: Assigning the dial plan to a device, on page 210.

Procedure: Assigning the dial plan to a device

Purpose: To associate the dial-plan DN with the device DN or Agent Login that will use the dial-plan when making a call.

Prerequisites

- A configured dial-plan DN with dial-plan rules (`Voice over IP Service` DN with `service-type` set to `dial-plan`). See Procedure: Configuring a Dial-Plan DN, on page 207.

Start of procedure

1. Under the SIP Server `Switch`, open the configuration object to which you want to apply the dial-plan. Supported objects include:
   - `Agent Login`
   - `Extension` DN
   - `ACD Position` DN
   - `Trunk` DN
   - `Trunk Group` DN
   - `Routing Point` DN
   - `Voice over IP Service` DN with `service-type=softswitch`
2. In the `TServer` section, add the `dial-plan` configuration option, with the value set to the name of the dial-plan DN.

   Figure 22 illustrates a sample configuration of the device DN with an assigned dial-plan and COS.

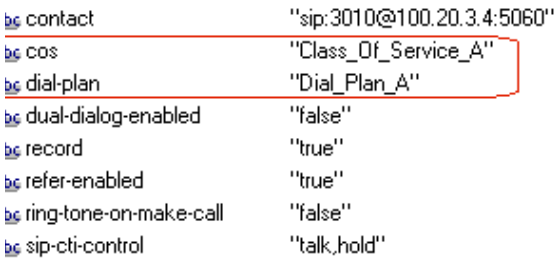

Figure 22: Assigning the Dial-Plan to a Device: Sample Configuration

Tip: Both COS for the dial-plan and the dial-plan itself can be assigned to multiple DNs by using the `Manage Options` function in GAX.

End of procedure

Next Steps

- If you are including Class of Service to control the allowed privileges for this DN or agent, continue to Procedure: Assigning COS to a device, on page 211.
- If you want to configure the dial plan and COS globally for all calls (unless otherwise configured at DN or Agent Login-level), continue to Procedure: Assigning the dial plan and COS globally, on page 212.

Procedure: Assigning COS to a device

Purpose: To associate the COS DN with the device DN or Agent Login that will use the defined privilege-levels (Class of Service) when making a call.

- A configured COS DN with defined privilege-levels (`Voice over IP Service` DN with `service-type` set to `cos`). See Table 41, "Configuring Class of Service," on page 174.

Start of procedure

1. Under the SIP Server `Switch`, open the configuration object you want to apply the COS DN. Supported objects include:
   - `Agent Login`
   - `Extension` DN
   - `ACD Position` DN
   - `Trunk Group` DN
   - `Routing Point` DN

2. In the TServer section, add the configuration option cos, with the value set to the name of the COS DN that you configured in Table 41, "Configuring Class of Service," on page 174.

End of procedure

Procedure: Assigning the dial plan and COS globally

Purpose: To assign the dial plan to the SIP Server application, where it applies globally to calls made from any DN on the switch (unless otherwise defined on the DN or Agent Login-level).

Prerequisites

- A configured dial-plan DN with dial-plan rules (Voice over IP Service DN with service-type set to dial-plan). See Procedure: Configuring a Dial-Plan DN, on page 207.
- A configured COS DN with defined privilege-levels (Voice over IP Service DN with service-type set to cos). See Table 41, "Configuring Class of Service," on page 174.

Start of procedure

- In the SIP Server Application object > Application Options tab > TServer section, configure the following options:
  - dial-plan—Set this option to the name of the dial-plan DN.
  - cos—Set this option to the name of the COS DN.

End of procedure

Procedure: Using SIP Feature Server Dial Plan

Purpose: To configure a SIP Feature Server Dial Plan.

Start of procedure

1. Administer the SIP Feature Server dial plan as described in the SIP Feature Server Administration Guide.
2. Configure the SIP Server that is associated with the Feature Server by setting the following option in the TServer section of the SIP Server Application:
   - dial-plan—Set this option to fs-dialplan, as described in the SIP Feature Server Deployment Guide.

3. Under a SIP Server Switch object that is associated with the SIP Server, create a VOIP Service DN named `fs-dialplan` and configure these options:
   - `service-type`—Set this option to `extended`.

     Important: Ensure that you add the final slash character (/) to the end of each of the following URLs.
   - `url`—Set this option to `http://<FS Node>:<port>/`

     For n+1 High Availability (HA), add the following parameters:
     - `url-1 = http://<FS Node2>:<port>/`
     - `url-2 = http://<FS Node3>:<port>/`
     - `url-n = http://<FS Node_N>:<port>/`

     Important: A Feature Server's dial plan URL must be configured only on a VOIP Service DN that was created on the Switch controlled by the SIP Server that is connected to that particular Feature Server.
   - (Optional) `enable-oosp-alarm`—Set this option to `true` to enable SIP Server to generate alarms 52035 and 52056. See "SIP Feature Server Log Messages" on page 213.

- If required, configure the following options in the SIP Server Application object, the `TServer` section:
  - `rp-use-dial-plan`—Set this option to a value suitable for your environment.
  - `enable-iscc-dial-plan`—Set this option to `true` to enable SIP Server to apply the dial plan to multi-site (ISCC) calls that are routed through an External Routing Point (`cast-type=route-notoken`).
- (Optional) In a routing strategy, set the `UseDialPlan` key extension in `TRouteCall`. The key extension setting takes priority over configuration options.

End of procedure

SIP Feature Server Log Messages

When the `enable-oosp-alarm` option is set to `true`, SIP Server generates the following alarms:

- `52056|STANDARD|GCTI_FEATURE_SERVER_URL_TIMEOUT|Feature Server URL %s missed response timeout`

  ...when SIP Feature Server does not reply within a specified timeout.
- `52035|STANDARD|GCTI_FEATURE_SERVER_URL_OFFLINE|Feature Server URL %s now offline`

  ...when SIP Feature Server does not respond on time a pre-set amount of times.
- The `52035` alarm will be cleared by the following message: `52036|STANDARD|GCTI_FEATURE_SERVER_URL_ONLINE|Feature Server URL %s now online`

Enhanced Handling of XS Requests

Introduced in 8.1.103.80

SIP Server can handle different HTTP error responses from SIP Feature Server for Dial Plan extended service (XS) requests in an enhanced way to address connection instabilities and provide a quality response to the origination side.

SIP Server sends an XS request to one of the SIP Feature Server URLs, starts the timer configured by the `xs-post-timeout` option, and waits for a Feature Server response. When the timeout expires, SIP Server sends an XS request to an alternative Feature Server URL. If SIP Server receives an error response within the timer period, it sends an XS request to an alternative Feature Server URL. In both cases, SIP Server sends an XS request to an alternative Feature Server URL only once.

When a Feature Server URL becomes out of service, SIP Server does not send subsequent requests to it until the Feature Server URL becomes in service. The Feature Server URL remains out of service, if the number of failed heartbeat requests exceeds the configured threshold (set in the `xs-missed-heartbeat-threshold` option), and that URL will not be selected for request processing, until it responds with a `200 OK` message for a heartbeat request.

Table 56 summarizes SIP Server actions for handling certain error responses received from Feature Server.

Table 56: SIP Server Actions for Handling XS Requests

Error Code	Description	Action
400 Bad Request	Invalid Request Format	SIP Server responds to a caller with the `503` message. It doesn't resend a request and doesn't mark the Feature Server URL as out of service.
404 Not Found	Invalid API	SIP Server responds to a caller with the `503` message. It doesn't resend a request and doesn't mark the Feature Server URL as out of service.
501 Not Implemented	Unsupported operation type	SIP Server responds to a caller with the `503` message. It doesn't resend a request and doesn't mark the Feature Server URL as out of service.

Table 56: SIP Server Actions for Handling XS Requests (Continued)

Error Code	Description	Action
503 Service Unavailable	Feature Server is unable to provide a response	SIP Server resends a request with an alternative Feature Server URL and marks the Feature Server URL that responded with `503` as out of service.
Any error response or Request Timeout	Feature Server internal error or unable to process a request	SIP Server resends a request with an alternative Feature Server URL and marks the Feature Server URL that responded with `503` as out of service, and responds to a caller with `503` on receiving an error or a timeout for retry.

When none of the Feature Server URLs are available and, as a result, the Feature Server VOIP Service DN is placed out of service, SIP Server starts rejecting call requests with a `503 Service Unavailable` message.

SIP Server running in primary mode switches over to backup mode if there is no active connection to any of the configured Feature Server URLs. If the `switchover-on-xs-oos` option is set to true, SIP Server reports the `SERVICE_UNAVAILABLE` status to LCA/SCS to switch over to backup mode instead of rejecting requests. This behavior ensures availability of the dial plan resolution in case of network instabilities.

SIP Server starts the switchover process after the timeout defined by the `time-before-switchover-on-xs-oos` option expires.

To control how long an XS request is considered active, use the `xs-request-timeout` option. If no response is received within this timeout, SIP Server rejects the request immediately with a `503 Service Unavailable` message.

All the above features can be enabled by setting the `enable-enhanced-dialplan-handling` option to `true` in the SIP Feature Server VOIP Service DN (`service-type=extended`).

Configuring Enhanced Handling of XS Requests

The following configuration options can be used to configure this feature:

- `enable-enhanced-dialplan-handling`
- `xs-request-timeout`
- `xs-post-timeout`
- `xs-heartbeat-timeout`
- `xs-missed-heartbeat-threshold`
- `switchover-on-xs-oos`
- `time-before-switchover-on-xs-oos`
- `xs-pool-size`
- `xs-heartbeat-interval`

Feature Limitations

- SIP Server rejects the Dial Plan XS requests with a `503 Service Unavailable` message instead of a `603 Decline` message, when:
  - A retry limit for a request is exceeded.
  - None of the Feature Server URLs are available to provide a service.
- This feature depends on support from a specific version of SIP Feature Server. Consult corresponding documentation for the availability of this new feature in that component.

Dial Plan For Multi-Site Calls

For multi-site calls, the Dial Plan is used on both the origination and destination sites. To avoid conflicts, Genesys recommends the following configuration:

Origination Site

- Dial Plan Selection: The dial plan assigned to the Agent Login or DN that initiated a call to the destination site is used (or the Application-level dial plan, if none is assigned to the Agent Login or DN object).
- Dial Plan Rule Selection: The dialed digits are used to select the dial-plan rule in the dial plan.
- Dial Plan Rule Application: Genesys recommends to place digit translation and call-barring rules (privilege and type=reject) at the origination site selected dial plan. Place destination forwarding rules, such as `ontimeout` and `ondnd`, at the destination site dial plan only, to avoid race conditions.

Destination Site Calls Using External Routing Points (ISCC type=route)

- Dial Plan Selection: The dial plan assigned to an ISCC trunk is used (or the Application-level dial plan is used if none is assigned to the ISCC trunk).
- Dial Plan Rule Selection: If a dial-plan rule matches the external Routing Point destination, then this rule is used. If there is no match for the external Routing Point, then a rule is selected that matches the agent destination DN instead.
- Dial Plan Rule Application: Genesys recommends to place destination forwarding rules only at the destination site selected dial plan (such as `ontimeout` and `ondnd`), to avoid race conditions with rules at the origination site.

AI Other Destination Site Calls

- Dial Plan Selection: The dial plan assigned to an ISCC trunk is used (or the Application-level dial plan will be used if none is assigned to the ISCC trunk).

- Dial Plan Rule Selection: Dialed digits without the stripped trunk prefix are used to select the dial-plan rule.
- Dial Plan Rule Application: Genesys recommends to place destination forwarding rules only at the destination site selected dial plan (such as `ontimeout` and `ondnd`), to avoid race conditions with rules at the origination site.

Note: Business Continuity configuration must follow the same multi-site recommendations, with the following limitations:

- For the destination site, Genesys recommends to use only an Application-level dial plan for destinations to provide consistent behavior for forwarding rules in all scenarios.
- Feature Server Dial Plan does *not* support user-based calling profiles.

Feature Limitations

The following limitations apply to a dial plan:

- Dial plan is not applied to the agent DN destination for calls that are distributed to the agent through an ACD Queue.
- Dial plan is not be applied to the destination of the `TCallForwardSet` request.

DNS Name Resolution

Using an internal DNS client, SIP Server is able to use DNS procedures for resolving a SIP URI to a corresponding IP address, port, and transport protocol. If enabled, SIP Server complies with the DNS procedures described in RFC 3263 “Session Initiation Protocol (SIP): Locating SIP Servers”, which includes support for multiple resolved destinations.

How It Works

DNS name resolution works in conjunction with the Active Out-of-Service Detection feature. The DNS server returns information about the resolved destinations from either of the following record types:

- DNS/A—address records (does not include priority and weight information)
- DNS/SRV—service records (includes priority and weight information, allowing these fields to be taken into account when resolving the SIP URI to multiple destinations)

Note: SIP Server performs all DNS name resolution procedures as described in RFC 3263, except for NAPTR records lookup.

If the SIP URI from the contact for a particular DN resolves to multiple destinations through a DNS/SRV lookup, SIP Server uses the "Active Out-of-Service Detection" feature to send `OPTIONS` requests to each resolved destination to determine which destination is available.

When sending SIP messages through `Trunk` or `Voice over IP Service` DNs that resolve to multiple destinations, by default SIP Server selects the destination without taking into account the "priority" or "weight" field from the DNS/SRV record. If you want SIP Server to take these fields into account, you must set `sip-enable-rfc3263` to `true`.

The Device Selection Algorithm

When deciding which device to select, SIP Server takes the following factors into consideration:

- If the URI contains both hostname and port, SIP Server resolves to IP address using a DNS/A record request.
- If the URI contains a hostname with no defined port, SIP Server resolves to IP address using a DNS/SRV record request. To use this method, you must enable "Active Out-of-Service Detection" on the DN.

Example

The following sample call flow demonstrates how SIP Server uses a DNS server to resolve a domain for an external Trunk with multiple destinations:

1. SIP Server sends a request to the DNS server to resolve the FQDN from the `contact` option of the `Trunk` DN.
2. The DNS server returns the SRV/A records with a list of the resolved destination IP addresses with the priority and weight for each.
3. SIP Server sends `OPTIONS` requests to each resolved destination to determine availability.
4. The agent initiates an outbound call—to route the call, SIP Server chooses from the list of active destinations according to the priority and weight information included in the SRV/A record. If there are two active destinations with the same priority, SIP Server selects the destination with the higher weight.

Feature Configuration

Table 57 describes how to enable DNS resolution through A/SRV records.

Table 57: Enabling DNS Name Resolution

Objective	Related Procedures and Actions
1. Configure the SIP Server Application.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following options: • `sip-enable-gdns`—Set this to `true`. This enables the internal DNS client. • (optional) `sip-enable-rfc3263`—Set this to `true`. This enables priority and weight to be factored in to the destination selection algorithm. • (optional) `sip-address-srv`—Enter the FQDN for the SIP Server host machine.
2. Configure the Trunk DN.	In the destination DN > `Options` tab > `TServer` section, configure the following options: • `contact`—Enter the URI for this destination DN. • `oos-check`—Set to option to a valid timeout, in seconds (`1` to `300`). For a more tuned oos-check performance, you can also enable the following related `oos-check` options: • `oos-force`—Set this to the length of time that SIP Server waits before setting an unresponsive device to out-of-service. • `recovery-timeout`—Set this to the length of time that a device is set to out-of-service in case of an error. • `sip-request-oos-timeout`—See this to the length of time that SIP Server waits before it considers a transport as failed (SIP Server abandons the dialog and sets transport out-of-service).

Asynchronous DNS Resolution

Introduced in SIP Server 8.1.103.87

SIP Server can resolve a Fully Qualified Domain Name (FQDN) specified in the contact option of a DN using the asynchronous DNS resolution method and place the DN out of service if the FQDN is unresolvable. The feature applies to DNs of type Extension, ACD Position, and Voice Treatment Port. The DN-level `enable-async-fqdn-resolve` configuration option enables this feature.

When performing DNS resolution asynchronously using the DNS library service, SIP Server does the following based on the result:

- If the DNS result is successful, SIP Server places the DN in service.
- If there is a DNS client error, SIP Server does nothing and considers this feature disabled.
- If there is a server-side error, SIP Server sends a retry attempt and, if the result is unsuccessful, places the DN out of service.

Feature Limitations

- This feature is enabled only after SIP Server restart. If the initial resolution fails after four retry attempts, the DN is placed out of service and no more retries are performed until SIP Server is restarted.
- If a DNS server is down or FQDN records are missing in the DNS server, SIP Server waits for 1 minute to recover. If the issue is not recovered, SIP Server does not send a retry attempt again until the next restart.

DTMF Clamping in a Conference

Introduced in SIP Server 8.1.101.68

Multi-site support added in 8.1.101.95

This feature guards a customer's sensitive credit card information from an agent and from call recording. DTMF clamping is supported in single-site and multi-site deployments. Here is how it works when activated and enabled:

1. The customer needs to enter a credit card number.
2. The agent adds IVR to the call, which bridges the customer, agent, and IVR.
3. The customer enters the requested credit card digits, but they are not recorded and the agent hears only silence.
4. The credit card number is passed to the IVR, securely.

This behavior is called DTMF clamping, and SIP Server supports it to comply with the Payment Card Industry Data Security Standard (PCI DSS).

MCP performs DTMF clamping for selected parties in a conference, for the following DTMF transmission modes:

- RTP packets with a Named Telephone Event (NTE) payload as specified by RFC 2833
- In-band audio tones (encoded using a regular audio codec, such as G.711)
- SIP `INFO` packets with the content-type `application/dtmf-relay`

SIP Server uses MSML messages to inform MCP about which legs of the conference should reveal DTMF tones and which legs should suppress DTMF tones. Each leg is controlled individually. SIP Server defines the DTMF mode for each leg based on the DN type or DN-level configuration option.

In multi-site deployments, SIP Server uses the same mechanism as for Call Participant Info notifications (`NOTIFY` requests) to provide information about multi-site call participants. Routing Point parties are now included in these `NOTIFY` requests when DTMF clamping is enabled.

Activating DTMF Clamping

1. Activate DTMF clamping by setting the Application-level option `clamp-dtmf-allowed` to `true`.

2. When activated, you can enable the feature on a DN object that is configured as IVR. For this purpose, IVR can be configured as DNs of type `Voice Treatment Port`, `Trunk`, or `Trunk Group`:
   - If IVR is configured as a DN of the type `Voice Treatment Port` is added to the conference, then DTMF tones are clamped for all parties in the conference except for the `Voice Treatment Port` DN. No DN-level configuration is required.
   - If IVR is configured as a `Trunk` or `Trunk Group` DN, then activate DTMF clamping by setting the `clamp-dtmf-enabled` option to `true` on the corresponding `Trunk` or `Trunk Group` DN.
3. In multi-site deployments, set the Application-level option `sip-enable-call-info` option to `true`.

On Routing Points

SIP Server automatically activates DTMF clamping in any conference where a Routing Point is invited. No DN-level configuration is required, and only a party represented by the Routing Point is allowed to receive DTMF digits. DTMF clamping is activated regardless of the type of treatment applied at the Routing Point, and it remains active as long as the Routing Point stays in the conference.

DTMF Clamping in Recordings

PCI compliance requires that DTMF tones are not recorded when clamping is enabled. To satisfy this requirement, recording must be disabled on the caller's leg. Otherwise, DTMF digits dialed by a caller could be recorded.

Genesys recommends that you enable recording on the agent's leg as shown in Figure 23.

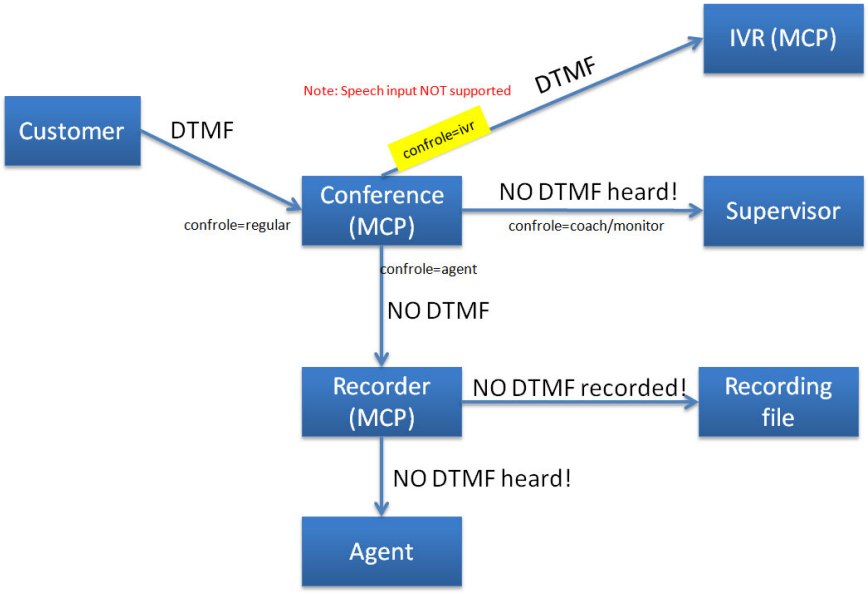

Figure 23: DTMF Clamping

Feature Limitations

DTMF Clamping requires the Application-level option `ringing-on-route-point` to be set to `true` (the default value) when DTMF digits are collected via a treatment applied at the Routing Point.

DTMF Tones Generation on Media Server

SIP Server supports two methods for initiating DTMF tone generation on the Media Server:

- The agent can request DTMF tone generation by sending a `TSendDTMF` request from Workspace Desktop. SIP Server pulls the Media Server into the call to generate DTMF tones. (Set the `sip-dtmf-send-rtp` option to `true`).
- A routing strategy can instruct SIP Server to send a request to the media server using the `TApplyTreatment` request with the `PlayApplication` treatment.

The DTMF tones generation method by Media Server can be used as an alternative method of DTMF generation using 3pcc. The most efficient method of DTMF generation by a SIP endpoint is using 3pcc where the agent inputs the digits to the Workspace Desktop. Workspace Desktop instructs SIP Server which digits to play (by sending `TSendDTMF`), and then SIP Server instructs the agent's SIP endpoint to generate the DTMF tones using a SIP `NOTIFY` message (set the `sip-cti-control` option to `dtmf`). Currently, only the Genesys SIP Endpoint/Softphone supports DTMF generation by a SIP endpoint using 3pcc. For others SIP endpoints, use the alternative DTMF generation method.

DTMF Parameters for PlayApplication Treatments

The following key-value pairs (see Table 58) are used in the attribute `parameters` for the `PlayApplication` treatment:

Table 58: Key-Value Pairs for TreatmentPlayApplication

Key	Type	Value
GSIP_APP_ID	Integer	The number of the specific application to be executed. Set this parameter to the value of `502` to trigger the play application.

Table 58: Key-Value Pairs for TreatmentPlayApplication (Continued)

Key	Type	Value
GSIP_DTMF_TO_DIAL **Note**: When designing the routing strategy in IRD, you must add the prefix `{s}` so that URS interprets the value as a string. By default, URS interprets the value as an integer (0).	String	The DTMF string to be generated.
GSIP_DTMF_DURATION	Integer	The duration of the DTMF tone in msec. This parameter is optional with the default value of `100 msec`.

Feature Configuration

Table 59 describes how to configure DTMF tones generation support.

Table 59: Configuring DTMF Tones Generation Support

Objective	Related Procedures and Actions
1. Configure a SIP Server `Application` object.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, set the configuration option `sip-dtmf-send-rtp` to `true`. **Note**: This option setting is required for `TSendDTMF`.
2. Configure an application service (NETANN).	Configure a DN of type `Voice over IP Service` with the following configuration options set in the `TServer` section: • `contact`—Set to the device's IP address used for sending DTMF tones. • `service-type`—Set to `application`. See Table 10, "Configuring an Application Service," on page 89 for details.
3. Configure an application service (MSML).	Configure a DN of type `Voice over IP Service` with the following configuration options set in the `TServer` section: • `contact`—Set to the Resource Manager's IP address/FQDN. • `service-type`—Set to `msml`. See Table 13, "Configuring an MSML Service," on page 91 for details.

Dummy SDP

In some cases, SIP Server may need to send "dummy" media session parameters (SDP) in the initial `INVITE` message for certain call routing scenarios. For example, to ensure that a continuous treatment is played to a caller in an established dialog until an available agent answers the call.

How It Works

The dummy SDP is formed according to the parameters specified in the `TRouteCall` request. These parameters are configured in the routing strategy as key-value pairs in the `AttributeExtensions` of the `TRouteCall` request:

- `sdp-c-host`—Any string. The value will be propagated as the connection address in the c= line of Dummy SDP. Typically, this would be the IP address of the media server host.
- `sdp-m-port-low`—Any integer that represents a valid UDP port. This value represents the low part in the range of ports to be included in the m= line of the SDP. Key rules:
  - The actual port used can equal this low value.
  - Only even-numbered ports are used.
- `sdp-m-port-high`—Any integer that represents a valid UDP port. This value represents the high part in the range of ports to be included in the m= line of the SDP. Key rules:
  - The actual port used can equal the low value, but cannot equal this high value.
  - Only even-numbered ports are used. For example, if the range is 4000 to 4010, the port in the INVITE can be 4000, 4002, 4004, 4006, and 4008 only.
  - If you leave this parameter unspecified (empty), then the value of the `sdp-m-port-low` will be used.
- `after-routing-timeout` —An integer that overrides the value of the `after-routing-timeout` configuration option, which specifies the length of time (in seconds) that SIP Server waits before diverting the call from the Routing Point DN to the destination DN after RequestRouteCall was processed. When the call is not diverted before the timeout expires, SIP Server generates an EventError message.

> **Note:** The `after-routing-timeout` parameter does not apply to SDP formation only, but can be used for other reasons.

Example of TRouteCall and Corresponding SDP

For example, a `TRouteCall` with the following `Extensions` Attributes (as applied from the routing strategy):

```
AttributeExtensions[107] 00 04 01 00..
      'after-routing-timeout'25
      'sdp-c-host''192.168.10.10'
      'sdp-m-port-low'44000
      'sdp-m-port-high'44500
```

results in a SIP `INVITE` request with the following SDP (**bold** indicates affected values):

```
v=0
o=Genesys 1287768950 1 IN IP4 192.168.10.10
s=3pcc Make Call
c=IN IP4 192.168.10.10
t=0 0
m=audio 44000 RTP/AVP 101 0 8 4 18 3
a=rtpmap:101 telephone-event/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:4 G723/8000
a=rtpmap:18 G729/8000
a=rtpmap:3 GSM/8000
```

Feature Configuration

Table 60 describes how to configure a dummy SDP.

Table 60: Configuring Dummy SDP

Objective	Related Procedure and Key Actions
1. Configure the Application.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following options: • `sip-treatments-continuous`—Set this option to `true`. • `sip-enable-100rel`—Set this option to `true`. • `divert-on-ringing`—Set this option to `false`.

Table 60: Configuring Dummy SDP (Continued)

Objective	Related Procedure and Key Actions
2. Configure the routing strategy.	In Interaction Routing Designer (IRD), configure the strategy to attach the following `Extensions` to the `TRouteCall`: • `sdp-c-host`—Enter the connection address provided in the `c=` line of the Dummy SDP. • `sdp-m-port-low`—Enter the low part in the range of valid UDP ports. • `sdp-m-port-high`—Enter the high part in the range of valid UDP ports. • `after-routing-timeout`—Enter the length of time that SIP Server waits for the call to arrive at the destination after processing the `TRouteCall`. Overrides the `after-routing-timeout` configuration option. For more information about configuring routing objects in IRD, consult the *Universal Routing 8.1 Reference Manual*.

E911 Emergency Gateway

SIP Server supports 911 emergency calling using the integration with the E911 EGW and service. When properly configured, 911 calls made from devices registered to SIP Server are sent through the EGW. Emergency calls can be made directly from the phone (1pcc call) or by request from a T-Library client (3pcc call); if both methods are available, 1pcc rather than 3pcc is recommended. As well, the EGW integration allows the Public Safety Answering Point (PSAP, or 911 dispatch) to discover the location of the dialing device and to provide a Call Back Number (CBN) in case the call is prematurely disconnected.

Feature Configuration

Table 61 describes how to configure SIP Server for E911 support.

Table 61: Configuring SIP Server for E911 Emergency Gateway

Objective	Related Procedures and Actions
1. Create the EGW DN.	SIP Server uses this DN to place the outgoing 911 call as initiated by the SIP endpoint. 1. Create a `Trunk Group` DN to represent the EGW, giving it the name of the emergency dialing number; for example, 911. 2. In the `TServer` section, configure the following options: • `contact`—Enter the IP address for the primary EGW. • `emergency-backup`—Enter a comma-separated list of IP addresses for the alternate route. The first address in the list must represent the backup EGW. The following addresses can represent any other PSTN gateways that deliver emergency calls to the 911 Enable service. • `emergency-device`—Set this option to true to enable this device to conduct emergency calls. **Note:** You can also enable Active Out-of-Service Detection for this DN (using the options `oos-check` and `oos-force`). If SIP Server detects this DN is unresponsive, it will not use the address from the contact but will instead try the addresses configured in the `emergency-backup` option.
2. Configure the dial plan (DID allowed).	For environments that allow DID calls, do the following: 1. Procedure: Creating a dial-plan DN for the ANI to CBN conversion, on page 229 2. Procedure: Assigning the EGW DN to the ANI-to-CBN dial plan, on page 229 3. Procedure: Creating a dial-plan DN for the CBN to ANI conversion, on page 230 4. Set the `dial-plan` option to the name of the dial-plan DN you created in Step 3. This option can be set at an Application level or at a DN level on `Trunk` DNs for inbound calls.
3. Configure the call path (DID not allowed).	For environments that do not allow DID calls, do the following: • Procedure: Configuring the call path (DID not allowed), on page 230

Table 61: Configuring SIP Server for E911 Emergency Gateway (Continued)

Objective	Related Procedures and Actions
4. Configure the endpoints.	1. Ensure that G711 is supported and enabled on all endpoint devices. Note that on the Genesys side, even if codec filtering is enabled (`sip-enable-sdp-codec-filter` is set to `true`) and G711 is not included in the filter exemption list (audio-codecs option), SIP Server will not filter the G711 codec, overriding the filter settings. 2. (Recommended) Where possible, configure the endpoint to use the `REFER` method for 3pcc calls (set `refer-enabled` to `true`). 3. If `MakeCall` using the `REFER` method is not possible, Genesys recommends that you configure the option `sip-invite-treatment-timeout` value to `5` seconds. This guarantees that in case all media services fail, emergency calls will not be stuck for any longer than this 5-second interval.
5. Run the Genesys Emergency Gateway Utility.	Genesys provides a utility that extracts a list of Extension DNs (and contact IP addresses) from Configuration Server (version 8.1.x only) and uploads it through FTP to the Emergency Gateway. 1. Procedure: Installing and Configuring the Utility, on page 231 2. When the script is started, it takes the following actions: a. It reads and processes the configuration file. b. It connects to Configuration Server via SOAP interface. c. From Configuration Server, it fetches a list of Extensions that belong to a particular switch. If the Extension is configured with an IP address for the option `contact`, the IP address will be extracted as well. d. It creates a `.txt` file in accordance with the EGW requirement. Note that 911 enable can only use a semicolon as a separator, not a comma. e. As the final step, the `.txt` file is uploaded to the EGW via FTP. 3. If these above operations are successful, three files are created: • Batch file which was uploaded to EGW. It has the same name as configuration file, but with the suffix `.txt` in place of `.cfg`. • Batch file which was uploaded during previous run of this script with the same configuration file. The name of this file includes the symbol "_" placed before the .txt suffix. • Log file which reflects the steps that the script performed. If all steps were successful, the last line should show as "`JOB: completed.`"

Procedure:
Creating a dial-plan DN for the ANI to CBN conversion

Purpose: This dial plan contains rules for transforming the calling DN (ANI) to the 10-digit Call Back Number (CBN) included as the `P-Asserted-Identity` header in the INVITE to the EGW. This CBN is required by the Public-safety Answering Point (PSAP) to identify the location of the 911 caller.

Start of procedure

1. Configure a `Voice over IP Service` DN with `service-type` set to `dial-plan`.
2. In the `TServer` section and create dialing rules to transform the ANI to the CBN.

   For example, if configuring dialing rules for extensions at two different locations (two groups, each with a different country and area code), you can create a separate dialing for each group:

   ```
   dial-plan-rule-1=1XXX=>650466${DIGITS}
   dial-plan-rule-2=3XXX=>925266${DIGITS}
   ```

   In this case, the CBN for extension `1334` will be `6504661334`. For extension `3592` the CBN will be `9252663592`. In this example, for extensions that do not start with either 1 or 3 rules will not be applied and DID will not be allowed.

End of procedure

Next Steps

- Procedure: Assigning the EGW DN to the ANI-to-CBN dial plan

Procedure:
Assigning the EGW DN to the ANI-to-CBN dial plan

Start of procedure

1. Open the EGW `Trunk Group` DN.
2. In the `TServer` section of the EGW `Trunk Group` DN, configure the option `emergency-callback-plan` with the name of the dial-plan DN.

End of procedure

Next Steps

- Procedure: Creating a dial-plan DN for the CBN to ANI conversion

Procedure:
Creating a dial-plan DN for the CBN to ANI conversion

Start of procedure

1. Create a `Voice over IP Service` DN with `service-type` set to `dial-plan`.
2. In the `TServer` section, create a dialing rule that will translate the 10-digit CBN back to the original internal DN.

   For example, to translate the CBN to the four-digit original extension, configure the dialing rule as follows:

   ```
   dial-plan-rule-1=XXXXXXXXXX=>${DIGITS:-4:4}
   ```

End of procedure

Next Steps

1. Configure the endpoints. See the Step 4 in the Table 61: Configuring SIP Server for E911 Emergency Gateway.
2. After the endpoints are configured, continue at Procedure: Installing and Configuring the Utility, on page 231.

Procedure:
Configuring the call path (DID not allowed)

Purpose: For environments that do not allow DID calls, you must create a `Trunk` DN for the EGW.

Start of procedure

1. Create a `Trunk` DN to represent the EGW. The name does not matter (SIP Server uses the prefix for Trunk selection).
2. In the `TServer` section, configure the following options:
   - `contact`—Enter the IP address for the primary EGW.
   - `emergency-backup`—Enter the IP address for the alternate route. This must be the address for the backup EGW.
   - `emergency-device`—Set this option to true to enable this device to conduct emergency calls.
   - `prefix`—Enter a prefix that SIP Server will use to find this DN. This prefix should match numbers from the pool chosen by the administrator of the EGW for dynamic callback.

End of procedure

Next Steps

1. Configure the endpoints. See the Step 4 in the Table 61: Configuring SIP Server for E911 Emergency Gateway.
2. After the endpoints are configured, continue at Procedure: Installing and Configuring the Utility.

Procedure: Installing and Configuring the Utility

Purpose: To install and configure the Perl utility on Ubuntu Linux. The Perl utility `911_Enable.pl` and an example of the configuration file `911_Enable.cfg` can be found in the folder `tools` located in the SIP Server installation folder, together with the SIP Server executable file.

As for June 2017, there is no straight-forward procedure to install the script dependency on Windows. It is recommended to use Ubuntu 16.04 Virtual Machine installed on Oracle Virtual Box or VMware Player.

Note: Configuration Server version 8.5.x does not support the SOAP interface.

Start of procedure

1. Install the following SOAP::Lite and NET::Telnet packages:
   - `sudo apt-get install libsoap-lite-perl`
   - `sudo apt-get install libnet-telnet-perl`
2. Create the `.cfg` file required by the utility.
   - `[cfgserver]`—Specifies the regular Configuration Server attributes—for example, tenant name, switch name—as well as which Extensions should be uploaded to the Emergency Gateway. The port parameter in this file does NOT correspond to the `port` option as set in the `confserv` section of Configuration Server. It instead corresponds to the `port` option as defined in the `soap` section.
   - `[egw]`—Specifies the Emergency Gateway FTP server attributes.

   For example, the following shows sample content for a `.cfg` file.

```
[cfgserver]
#host=<config server hostname or IP>
host = host12345
#port=<config server port>
port=3034
#username = <config server username>
username = default
#password = <config server password>
password = password
tenant = tenant12345
```

```
switch = sip_server_switch
[egw]
host=172.21.83.197
username=batchendpoint
password=911batch
```

3. Name the `.cfg` file.

   If you are going to supply the name for this `.cfg` file as a parameter, then you can use any name with the suffix `.cfg`. For example, your utility and configuration files could be named as follows:

   ```
   Perl 911_Enable.pl   911_Enable_sip1.cfg
   ```

   However, if you will be starting the utility without any arguments, then you must use the same filename for both configuration and utility files. For example,

   ```
   Perl 911_Enable.pl   911_Enable.cfg.
   ```

4. Place this `.cfg` file in the same working directly as the `.pl` utility file.
5. For multiple SIP Server instances, create a separate `.cfg` file for each SIP Server instance. You must start the utility once for each instance.

End of procedure

Early Media for Inbound Calls

SIP Server supports the exchange of early media before a particular session is accepted—for example, to provide an audio treatment before the call is answered, thereby avoiding toll charges for the caller.

SIP Server provides support for early media through the offer/answer exchange of provisional responses (`183 Session Progress`) and `UPDATE` requests, to manage the session parameters (SDP) that are required to deliver the early media. All early media dialog activity takes place before the `200 Ok` response. Once the call is established (`200 Ok` is sent), no further toll-free services are possible.

NETANN Sample Call Flow

The following NETANN sample call flow demonstrates the signaling that is used to provide early media for an inbound call in case of early media support (note that all dialog activity takes place before the `200 Ok` response):

1. An inbound call arrives at a Routing Point from a gateway that is enabled for SIP early dialog (`sip-early-dialog-mode` set to `1`, or `sip-server-inter-trunk` set to `true`).
2. A treatment is applied. SI P Server sends a reliable `183` response with an answer to the calling-party offer.

3. The next treatment is applied. SIP Server sends an `UPDATE` with a new offer to the calling party.
4. The call is routed to the agent. The connection with the agent is initiated and completed using `UPDATE`. The dialog is then switched to the accepted state by sending a `200 OK` to the initial `INVITE` request, at which point no additional toll-free services are possible.

Controlling Early Media with a Routing Strategy

Introduced in SIP Server 8.1.102.25

With the Early Media for Inbound Calls feature enabled, this enhancement (a new `charge-type` extension key) enables you to create a routing strategy that does the following for an inbound call:

- Switch audio treatments from cost-free early media to an established state (charged) in a SIP dialog, which can be made at the initial `TApplyTreatment` or at any sequential `TApplyTreatment`. All consecutive audio treatments in this dialog will be charged.
- Play initial audio treatments in cost-free early media in deployments that are configured to play audio treatments at a cost, until a `TApplyTreatment` request containing the `charge-type` key set to 2 (charged) arrives.

The transition from early media to an established state can be made only once within a SIP dialog and only when changing from cost-free audio treatments to charged audio treatments.

This functionality is supported for MSML deployments and is not supported for NETANN deployments.

To configure controlling early media with a routing strategy, see Step 4 in Table 62.

Feature Configuration

Support for the `UPDATE` method for cost-free early media is configured by using the options `sip-early-dialog-mode` and `charge-type`. Table 62 describes how to configure this feature.

Table 62: Enabling Cost-Free Early Media

Objective	Related Procedures and Actions
1. Configure the gateway `Trunk` DN.	To support the `UPDATE` method for early media exchange, configure the `Trunk` DN (in the `TServer` section) for the incoming gateway as follows: • Set the `sip-early-dialog-mode` option to `1`. OR • Set the `sip-server-inter-trunk` option to `true`.
2. Configure the SIP device.	Enable cost-free early media service on applicable DNs as follows: • `Voice over IP Service` DNs—Leave the `charge-type` option at the default setting of `0`. • `Trunk Group` or `Extension` DNs—Set `charge-type` to `1`.
3. Configure the SIP Server Application.	Genesys recommends that you also configure the SIP Server Application as follows: • In the `TServer` section, set `ringing-on-route-point` to `false`.
4. (Optional) To control early media with a routing strategy	In the routing strategy, specify the `charge-type` key in `AttributeExtensions` of the `TApplyTreatment` request.

Feature Limitations

SIP Server supports `UPDATE` requests with SDP only for early dialogs and does not support `UPDATE` requests for established dialogs.

In early dialog mode, SIP Server does not play Music On Hold when `UPDATE` containing a hold SDP is received. However, if a device that sends `UPDATE` is configured with `sip-enable-moh` set to `na` or `sip-enable-moh` is not configured, and the `sip-enable-moh` option is set to `na` at the Application level, SIP Server will forward the incoming `UPDATE` request with the hold SDP to the caller.

In all other cases, during early media, SIP Server responds to the `UPDATE` request containing a hold SDP with `500 Internal Server Error`.

Emulated Agents

SIP Server fully emulates business-call–handling functions. It also performs agent emulation for any agent who logs in using a request, when the device provided in the `ThisQueue` attribute is defined as a Routing Point or ACD Queue in the SIP Server configuration.

SIP Server provides a fully functional agent model that enables full agent support for SIP Server desktop applications as well as for other Genesys solutions.

Business-Call Handling

This section describes how SIP Server handles different types of calls.

SIP Server Call Classification

SIP Server automatically assigns every call to one of three categories—*business, work-related,* or *private*. Based on this assignment, SIP Server applies the appropriate business-call handling after the call is released.

Business Calls

SIP Server automatically categorizes any call distributed to an agent either from a Queue or from a Routing Point as a *business call.* Use the following configuration options to define what additional calls to or from an agent are classified as business calls:

- `inbound-bsns-calls`
- `outbound-bsns-calls`
- `inherit-bsns-type`
- `internal-bsns-calls`
- `unknown-bsns-calls`
- `agent-only-private-calls`
- `bsns-call-dev-types`

Work-Related Calls

SIP Server categorizes any call that an agent makes while in the busy (with a business call), After Call Work (ACW), or aux-work state as a *work-related call*. SIP Server does not apply any automatic business-call handling after a work-related call.

Agents can make or receive direct work-related calls while in wrap-up time. If an agent makes a work-related call during this wrap-up time, the call is considered part of the wrap-up activities, and SIP Server continues the emulated wrap-up timer. If the agent receives a call, however, the call is considered unsolicited—not part of the wrap-up activities—and so SIP Server pauses the timer for the duration of this call.

If an agent receives a direct work-related call during legal-guard time, SIP Server cancels the legal-guard timer and reapplies it at the end of the work-related call.

Private Calls

SIP Server categorizes any call that does not fall into the business or work-related categories as a *private call*. SIP Server does not apply any automatic business-call handling after a private call. If emulated agents receive a direct

private call while in wrap-up or legal-guard time, the emulated wrap-up or legal-guard timer is not interrupted.

Emulated Agents Support

SIP Server provides a fully functional emulated-agent model that you can use either in addition to agent features available on the PBX, or in place of them where they are not available on the PBX.

When this feature is used, SIP Server emulates the following functionality:

- Login and logout
- Agent set Ready
- Agent set Not Ready (using various work modes)
- Automatic after-call work
- After call work in idle
- Automatic legal-guard time to provide a minimum break between business-related calls

Emulated Agent Login/Logout

You can configure SIP Server to perform emulated login either always, never, or on a per-request basis. Use the following SIP Server configuration options to configure emulated agent login:

- `emulated-login-state`
- `agent-strict-id`

Agent Logout on Client Unregistering from DN

In some scenarios (such as a desktop crash or power failure/disconnection), agents may still receive calls but be unable to handle them. To prevent this problem, SIP Server can be configured to automatically log the agent out in such circumstances.

When a client desktop or application disconnects from the SIP Server while an agent is still logged in, the SIP Server receives a notification that the application is unregistering from the agent's DN. Also, the SIP Server is able to uniquely identify the client application which sends a T-Library request, including `TAgentLogin` and `TRegisterAddress`.

The SIP Server can associate the client application (the one that sends the initial `TAgentLogin` request) with the agent and automatically log that agent out when the client application unregisters the agent DN while the agent is still logged in. (The initial `TAgentLogin` request is the one which first logs the agent in).

This feature is enabled/disabled by the following configuration options:

- `agent-logout-on-unreg`

- agent-emu-login-on-call
- agent-logout-reassoc

Emulated Agent Ready/NotReady

Emulated agents can perform an emulated Ready or NotReady request regardless of whether they are on a call, subject to the rules governing work modes.

SIP Server also reports any change in agent mode requested by the agent while remaining in a NotReady state (*self-transition*).

Note: Note that the *Genesys Events and Models Reference Manual* and the *Platform SDK 8.x .NET (or Java) API Reference* define which agent state/agent mode transitions are permissible.

Emulated After-Call Work

SIP Server can apply emulated wrap-up (ACW) for agents after a business call is released, unless the agent is still involved in another business call (see "Business Calls" on page 235).

Timed and Untimed ACW

SIP Server applies emulated ACW for an agent after any business call is released from an established state. SIP Server automatically returns the agent to the Ready state at the end of a *timed* ACW period. The agent must return to the Ready state manually when the ACW period is *untimed*.

Events and Extensions

SIP Server indicates the expected amount of ACW for an agent in EventEstablished, using the extension WrapUpTime. It is not indicated in EventRinging, because the value may change between call ringing and call answer. Untimed ACW is indicated by the string value untimed; otherwise, the value indicates the expected ACW period in seconds.

SIP Server reports ACW using EventAgentNotReady with workmode = 3 (AgentAfterCallWork), and it indicates the amount of ACW it will apply using the extension WrapUpTime.

SIP Server sends EventNotReady(ACW) before EventReleased at the end of the business call.

Emulated ACW Period

The amount of emulated ACW that SIP Server applies (when required) after a business call is determined by the value in configuration option wrap-up-time.

Configuration option untimed-wrap-up-value determines which specific integer value of wrap-up-time indicates *untimed* ACW. To specify untimed ACW in request extensions or user data, you should use the string untimed instead. All positive integer values are treated as indicating timed ACW (in

seconds). For backward compatibility, the default value of `untimed-wrap-up-value` is `1000`.

Note: Changing the value of untimed ACW should be done with care, because it may affect the interpretation of all integer values of the option `wrap-up-time` in the Configuration Layer. If lowered, it may change timed ACW to untimed ACW, or disable ACW altogether. If raised it may change untimed or disabled ACW to timed ACW. The use of the option (string) value `untimed` is encouraged where possible to minimize the impact of any future changes to the value of option `untimed-wrap-up-value`.

ACW in Idle

An agent can activate wrap-up time on request when idle, by issuing a `TAgentNotReady` request with `workmode = 3 (AgentAfterCallWork)`.

You can configure this feature using the following options:

- `timed-acw-in-idle`
- `acw-in-idle-force-ready`

Extending ACW

An agent can request an extension to the amount of emulated ACW for a call while in emulated ACW or in the legal-guard state.

The agent requests an extension to ACW by sending `RequestAgentNotReady` with `workmode = 3 (AgentAfterCallWork)`. SIP Server determines the period of the extended ACW from the extension `WrapUpTime`, as follows:

- Value = `0`—There is no change to the ACW period, but SIP Server reports how much ACW time remains.
- Value greater than `0`—SIP Server adds the given number of seconds to the timed ACW period. Untimed ACW remains unaffected.
- Value = `untimed`—SIP Server applies untimed ACW.

SIP Server sends `EventAgentNotReady` with `workmode = 3 (AgentAfterCallWork)`, reporting the newly extended amount of ACW using the extension `WrapUpTime`. If the agent was in the emulated legal-guard state, SIP Server places the agent back into the emulated ACW state.

The agent may extend the period of ACW as many times as desired. At the end of the extended timed ACW period, SIP Server applies legal-guard time, if any is configured. No legal-guard time is applied if the emulated ACW was untimed.

Calls While in Emulated ACW

SIP Server's handling of an agent making or receiving a call while in emulated ACW is governed by the configuration option `backwds-compat-acw-behavior`.

Emulated Legal-Guard Time

SIP Server applies emulated legal-guard time for agents before they are about to be automatically set `Ready` after any period of timed ACW, or after the last business call is released where there is no ACW to be applied. It is a regulatory requirement in many countries that agents have a break of a few seconds before the next call can arrive. No legal-guard time is applied if the ACW period was not timed, or if the agent is not being placed into the `Ready` state.

SIP Server reports legal-guard time using `EventAgentNotReady` with `workmode = 2 (LegalGuard)`. If an agent requests to be logged out during emulated legal-guard time, SIP Server immediately logs the agent out.

If the agent requests to go to a `Not Ready` or `Ready` state during legal-guard time, SIP Server terminates legal-guard time and transitions the agent to the requested state. If the agent requests to return to the ACW state, SIP Server reapplies legal-guard time at the end of ACW, provided that the agent still requires it according to the preceding criteria.

The period of legal-guard time is determined by the configuration option `legal-guard-time`.

Endpoint Service Monitoring

When SIP Server starts up, it considers that all DNs that are configured in the Configuration Layer are in the `In Service` state.

SIP Server supports two methods for detecting whether a particular device is unavailable and needs to be placed in the out-of-service state:

- Passive Out-of-Service Detection—SIP Server considers a device to be out-of-service after the SIP endpoint fails to respond to the incoming `INVITE` message during the creation of a new call.

  Note: Where SIP Server functions as an application server behind a softswitch, a DN is considered to be in `Out of Service` state if the softswitch responds with a `408 Request Timeout` message to an `INVITE` message during creation of a new call.

- Active Out-of-Service Detection—SIP Server checks the availability of the device by regularly sending SIP `OPTIONS` requests to the DN that represents the device.

Passive Out-of-Service Detection

In this method, when the SIP endpoint fails to respond to the incoming `INVITE` message during the creation of a new call SIP Server generates an `EventDNOutOfService` message.

DNs are considered to be back in service in several scenarios:

- When a SIP REGISTER message comes from the endpoint.
- When an endpoint initiates a call by sending an INVITE message.
- When an endpoint responds to an INVITE message via a Ringing (OK) message.
- When a timeout period, as configured by the recovery-timeout option, expires. This option can be configured on Trunk, Voice over IP Service, Voice Treatment Port, Extension, or ACD Position DNs only.

> **Note:** Genesys recommends using Active Out-of-Service Detection for Trunk and Voice over IP Service DNs. If Passive Out-of-Service Detection is required for these DNs, you can enable the feature by setting recovery-timeout on these DNs to a non-zero value.

- When the device entry in the Configuration Layer is changed.

An EventDNBackInService message is generated in all scenarios.

> **Note:** Trunk Group DNs cannot be placed out of service. If a Trunk Group DN does not respond to an INVITE, the DN remains available for further call attempts.

Enabling Passive Out-of-Service Detection

To enable passive out-of-service detection, configure the following options in the TServer section of the corresponding DN:

- recovery-timeout
- sip-oos-enabled

Active Out-of-Service Detection

SIP Server supports Active Out-of-Service Detection that can be enabled for the following types of DNs:

- Voice over IP Service (msml, MCU, treatment, softswitches, and so on)
- Trunk
- Trunk Group

Enabling Active Out-of-Service Detection

To enable active out-of-service detection, configure the following options in the TServer section of the corresponding DN:

- oos-check
- oos-force

Note: Genesys recommends that you not configure the `recovery-timeout` option when using Active Out-of-Service Detection. This timeout is intended for Passive Out-of-Service Detection only.

However, if `recovery-timeout` and Active Out-of-Service Detection are enabled at the same time, when the device is detected as out of service, and the `recovery-timeout` is configured to a value less than the `oos-check` value, SIP Server will wait the amount of time specified in the `recovery-timeout` option before it checks if the device is back in service.

Pinging a Device using SIP OPTIONS message

The `oos-check` option specifies how often (in seconds) SIP Server sends `OPTIONS` messages to check the device for out-of-service status. When no response is received, and the `oos-force` option is configured, SIP Server waits until the specified `oos-force` timeout expires before it places a device that does not respond in the out-of-service state.

Forwarding OPTIONS Through a Proxy

SIP Server provides the ability to configure the value of the `Max-forwards` header used in the `OPTIONS` messages that SIP Server sends to check the availability of a particular SIP device. The DN-level option `oos-options-max-forwards` allows a proxy device to forward the `OPTIONS` message to the monitored device, when SIP Server and the monitored device do not share a direct connection. For example, when a session border controller (SBC) sits between SIP Server and the switch where the monitored DN is registered.

Log Messages

For `Voice over IP Service` DNs, SIP Server generates the following log message stating that the specified device is out of service based on Active Out-of-Service Detection:

```
52000|STANDARD|GCTI_DEVICE_OUT_OF_SERVICE|Device [the name of the device] is out of service
```

For `Voice over IP Service` DNs, SIP Server generates the following log message stating that the specified device is back in service based on Active Out-of-Service Detection:

```
52001|STANDARD|GCTI_DEVICE_BACK_IN_SERVICE|Device [the name of the device] is back in service
```

Failed Route Notifications

SIP Server supports a variety of alarm messages for unsuccessful routing scenarios.

When this feature is enabled, a failed route timer is set using the interval defined in the route-failure-alarm-period configuration option. Each routing failure reported during this period is added to a counter. If this counter exceeds a "high water mark" threshold value defined by the route-failure-alarm-high-wm configuration option, SIP Server sets a route failure alarm condition and resets the counter.

The alarm condition is cleared when fewer route failures than configured in the route-failure-alarm-low-wm configuration option are recorded and there is also no more than the number of route failures configured in route-failure-alarm-high-wm in one complete period (configured in route-failure-alarm-period).

Setting the value of the route-failure-alarm-period configuration option to 0 (zero) disables the feature.

High-Availability Considerations

Only the primary SIP Server maintains the failed routing counter. The backup SIP Server does not run the route-failure-alarm-period timer, and keeps the routing failure alarm in the canceled state.

On switchover from primary role to backup role, SIP Server stops the route-failure-alarm-period timer and clears any alarm internally, without sending any LMS message. On switchover from backup role to primary role, SIP Server starts the route-failure-alarm-period timer and starts counting route requests and routing failures.

Feature Configuration

Table 63 describes how to configure this feature.

Table 63: Configuring Failed Route Notifications

Objective	Related Procedures and Actions
Configure the SIP Server Application.	In the SIP Server Application > Application Options > TServer section, configure these options: • route-failure-alarm-period—Set the duration for the failed routing timer. Failed attempts during this period are added to the counter. • route-failure-alarm-low-wm—Set the high water mark (the number of routing attempts). If the counter reaches this number during the failure period, an alarm is sent and the counter is reset. • route-failure-alarm-low-wm—Set the low water mark after which the alarm condition is cleared.

Find Me Follow Me

Introduced in SIP Server 8.1.101.75

SIP Server supports the SIP Feature Server Find Me Follow Me functionality for any 1pcc and 3pcc calls where Feature Server dial plans are applied to destinations. The feature is supported for MSML–based environments.

For this feature, SIP Server supports:

- Sequential dialing (SIP Server dials all locations sequentially)
- Parallel dialing (all locations are dialed simultaneously)
- Early media for inbound calls

Feature Configuration

Table 64 describes how to configure Find Me Follow Me on the SIP Server side. Refer to the SIP Feature Server documentation for more information.

Table 64: Configuring Find Me Follow Me

Objective	Related Procedures and Actions
Configure the SIP Server Application.	In the SIP Server `Application` > `Application Options` > `TServer` section, configure these options: • `fmfm-prompt-file`—(Optional) Specify the filename of the confirmation prompt. • `fmfm-confirmation-digit`—(Optional) Specify the digit that a caller must enter for call confirmation. This digit could be included in the prompt to be used for human recognition. • `fmfm-confirmation-timeout`—(Optional) Specify the timeout value, in seconds, that SIP Server waits for a confirmation digit to be entered. • `fmfm-trunk-group`—Specify the `Trunk Group` DN where events are generated, when each destination leg connects to Media Server. • `msml-support`—Set this option to `true`.

Feature Limitations

- Find Me Follow Me is not compatible with the agent state monitored by Stat Server. If calls routed to agents have Find Me Follow Me rules applied, then the state of the DN where the agent is logged in might not be changed when the call is delivered to a non-monitored agent phone, and the next call could be delivered to the same agent.
- Early media is not supported for outgoing calls.
- Media service recovery is not supported with Find Me Follow Me. If an MSML dialog for call confirmation fails, SIP Server handles it as successful confirmation.

Genesys Voicemail

SIP Server supports integration with Genesys SIP Feature Server, formerly known as Genesys SIP Voicemail (versions 8.1.0 and 8.1.1). Genesys SIP Feature Server (version 8.1.2 or later) is a SIP-based voicemail and SIP feature manager for Genesys contact centers and enterprise environments. Callers leave voicemail, and users retrieve and manage that voicemail. Administrators manage users, devices, voicemail, and call disposition (the dial plan). A distributed architecture enables scalability and enhances performance.

SIP Server also supports SIP Voicemail of SIP Feature Server in a Business Continuity deployment. This feature requires SIP Feature Server version 8.1.2 or later.

Supported voicemail features:

- Voicemail deposit and retrieval
- Group voicemail deposit and retrieval for Agent Groups only
- SIP and T-Library MWI notifications

For all configuration details relating to this integration, including DN configuration, refer to the SIP Feature Server documentation.

Reliability And Voicemail Session Recovery

Starting with version 8.1.101.42, SIP Server supports reliability and voicemail session recovery if it detects a problem (such as unresponsiveness or error response) on an attempt to initiate a voicemail session on a Media Server. Depending on the type of problem, SIP Server may try to restart the session on an alternate available service or a different MSML object for an Active-Active Resource Manager deployment, or it may try a different voicemail device for an Active-Standby voicemail deployment.

HTTP Live Streaming

Introduced in SIP Server 8.1.102.28

SIP Server supports HTTP Live Streaming (HLS) in the following scenarios:

- When treatments are applied on a Routing Point (`TreatmentPlayAnnouncement/TreatmentMusic`)
- For music-on-hold treatments, when a call on the DN is placed on hold, or when a call is waiting on an ACD Queue

The feature is available through the MSML protocol.

Feature Configuration

To use this feature, SIP Server must be integrated with MCP version 8.5.161.34 or later.

1. In the SIP Server configuration, do the following as required:
   - For music-on-hold treatments, either at an Application or DN level, specify the proper URL to HTTP Live Streaming server in the `default-music` option.

     For example: `default-music=http://123.45.678.90/hls/audio`
   - For music treatments, in a routing strategy, specify the URI to the HTTP Live Streaming server in the `MUSIC_DN` treatment parameter.
   - For announcements, in a routing strategy, specify the URI to the HTTP Live Streaming server in the `TEXT` treatment parameter.
2. In the MCP configuration, specify the format of audio segments in the transcoders parameter. For example: if audio segments are encoded in the MP3 format, you must add `MP3` into the list of transcoders, as follows:

   `[mpc].transcoders=G722 GSM G726 G729 MP3`

HTTP Monitoring Interface

Introduced in SIP Server 8.1.102.13

SIP Server provides the ability to monitor various operational statistics for its internal modules and statistics relating to trunks. See the *Supplement to the SIP Server Deployment Guide* for more information.

Hunt Groups

Sequential ringing introduced in SIP Server 8.1.101.27

SIP Server supports the Hunt Groups feature as a type of call coverage to distribute incoming calls to a statically configured group of extensions. The Hunt Group call distribution strategy (sequential or parallel) controls how a call is propagated to one or to all extensions within the group.

Support for BC deployments added

Starting with version 8.1.101.49, Hunt Groups with the parallel distribution strategy (simultaneous ringing) are supported in Business Continuity deployments. See the *SIP Server 8.1 High-Availability Deployment Guide* for details.

How It Works

Hunt Group members are Extension DNs or ACD Position DNs listed in the `hg-members` option. In contrast to the typical Genesys call distribution using a routing point, URS/ORS and Stat Server, Hunt Group does not rely on or require any login. Hunt Group distribution does take into account the status of each DN, and will distribute calls only to those DNs which meet the following criteria:

- DN must be in-service
- DN must be idle (not in a call)
- DN must not have DND or Call Forwarding set on SIP Server

In a sequential call distribution, SIP Server selects one of the available Hunt Group members as a target for the call distribution. If the Hunt Group member answers the call, the call is diverted from the Hunt Group and the distribution is complete. If the call is rejected by the Hunt Group member, or not answered within a specified period of time (`hg-noanswer-timeout`), SIP Server selects the next available Hunt Group member for a call distribution. Depending on the configuration, SIP Server uses one of the following strategies for Hunt Group member selection:

- Linear hunting—SIP Server always distributes the calls to the first Hunt Group member, then to the second, to the third, and so on. Hunting stops at the last Hunt Group member.
- Circular hunting—SIP Server distributes the calls in a round-robin fashion. If a call was previously delivered to the first Hunt Group member, the next call SIP Server distributes to the second member, and so on. The succession throughout each of the Hunt Group members continues even if one of the previous members becomes available. When a list of Hunt Group members is exhausted, the hunting starts over at the first member. Hunting stops at the Hunt Group member who answered the previous call. That is, SIP Server makes only one circle through the Hunt Group member list.

In a parallel call distribution, when any Hunt Group member answers the call, the call is diverted from the Hunt Group and SIP dialogs with the non-answered Hunt Group members are dropped. This SIP Server behavior is known as divert-on-answer and works differently from the usual queue distribution enabled by the `divert-on-ringing` configuration option. For the Hunt Groups feature, you do not need to set the `divert-on-ringing` option to `false`. The `hg-type` option triggers that by default.

The call distribution is considered unsuccessful if:

- None of the Hunt Group members answers the call.
- There are no available Hunt Group members during the specified period of time (`hg-queue-timeout`).
- The number of queued calls on the Hunt Group exceeds the specified limit (`hg-queue-limit`).

The unanswered call is distributed to the default destination if it is configured; otherwise the call is released.

It is not recommended to use Extension or ACD Position as the `default-dn` destination for the Hunt Group to avoid call overflow at that DN. A Routing Point DN should be used instead.

Note: Configuration of the `default-dn` as a member of the Hunt Group is not supported.

Call forward redirection from a SIP endpoint or from an agent desktop application like Interaction Workspace will be ignored for calls distributed from a Hunt Group. Calls distributed by the Hunt Group to a member will not be diverted to the member's mailbox if there is no answer.

Feature Configuration

Table 65 describes how to enable this feature.

Table 65: Configuring Hunt Groups

Objective	Key Procedures and Actions
Configure a DN of type `ACD Queue`.	Go to `Switch` > `DN` object of type `ACD Queue`, `Options` tab > `TServer` section and configure the following options: • `hg-type`—Specify the type of Hunt Group algorithm that is used to deliver calls to Hunt Group members. • `hg-members`—Specify members of the Hunt Group by listing DNs separated by a comma. • `hg-noanswer-timeout`—Set the period of time that a call distributed to the members waits to be answered by a member. • `hg-queue-timeout`—Set the period of time that a call can remain in the Hunt Group (while all Hunt Group members are not reachable) before being sent to Hunt Group members. • `hg-queue-limit`—Set a maximum number of calls that can be queued on the Hunt Group at the same time. • `hg-busy-timeout`—Set the period of time during which SIP Server will not distribute calls to the Hunt Group member's device after it answers with an error. • `default-dn`—(Optional) Specifies the default destination where a call is distributed if one of the following conditions occurs: ◆ `hg-noanswer-timeout` expires ◆ `hg-queue-timeout` expires ◆ `hg-queue-limit` is exceeded ◆ no correct Hunt Group members are defined • If the Hunt Group does not have the `default-dn` option defined, SIP Server uses the Application-level `default-dn` instead.

Feature Limitations

- Hunt Groups are not compatible with SIP Server's Early Media feature. A call to a Hunt Group will be immediately connected, which typically results in the caller being charged before the call is answered by an agent.
- Predictive calls (initiated by the TMakePredictiveCall request) are not supported. If a predictive call arrives at a Hunt Group, it will be rejected by the Hunt Group.
- Hunt group is not supported in deployment with IMS (double-triggering).
- A DN with nailed-up connection (`line-type=1`) must not be a member of a Hunt Group.

- DNs of the Hunt Group members must be located on the same switch as the Hunt Group.
- Calls distributed from a Hunt Group will not invoke the external Feature Server dial plan.
- It is not possible to use the Call Pickup feature to answer ringing calls for members of the Hunt Group. An attempt by a Hunt Group member to answer a call using the Call Pickup feature will be rejected.
- 1pcc semi-attended, 3pcc semi-attended, and 3pcc mute transfers to a Hunt Group destination are not supported.

IMS Integration

SIP Server supports integration with IP Multimedia Subsystem (IMS) environments. In an IMS architectural framework, SIP Server is configured as the SIP Application Server (SIP AS), which provides the interface with the Serving-Call Session Control Function (S-CSCF) to deliver the suite of Genesys applications to the IMS network.

Genesys Contact Center in the IMS Network

When integrated into an IMS network, Genesys provides the following Contact Center (CC) functions:

- SIP Server provides the interface between the IMS network and the Genesys suite.
- Genesys Media Server provides the dedicated Media Resource Function (MRF, providing media services to Genesys applications (at the signaling level, it does not communicate directly with the IMS core).
- Genesys Contact Center agents assume the role of IMS users.

Routed Calls as Originating or Terminating

Introduced in SIP Server 8.1.101.20

In IMS deployments, SIP Server can route calls parked on a Media Server using a call-originating leg or terminating leg. Call-originating legs, compared to call-terminating legs, contain the `orig` parameter in the `Route` header of an `INVITE` request that SIP Server sends to the IMS.

Originating legs are subject to HSS interactions and might pass through a chain of application servers serving originating calls. Note that this processing consumes network and CPU resources.

For more information about termination and initiation INVITEs in the IMS, see the *3GPP TS 24.229 V9.0.0 (2009-06)—Technical Specification 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP)*.

Feature Configuration

Table 66 describes how to integrate SIP Server into an IMS network.

Table 66: Integrating SIP Server with IMS

Objective	Related Procedures and Actions
1. Configure the SIP Server Application.	In the SIP Server `Application` > `Application Options` > `TServer` section, configure these options: • `server-role`—Set this to `1`. • `ims-route*`—Set this to the SIP URI for the S-CSCF. • `ims-skip-ifc*`—Set this to the value configured for the IFCs in the IMS Service Profile. * These options are not used in the alternative deployment with Mediation Proxy. **Routed Calls as Originating or Terminating:** To route calls parked on a Media Server using a call-terminating leg, set `ims-use-term-legs-for-routing` to `true`.
2. Configure IMS endpoints.	Go to SIP Server `Switch` > `DNs` folder. 1. Create a DN of type `Extension`. 2. In the `TServer` section, configure the following options: ◆ `enable-ims`—Set to `true`. ◆ `refer-enabled`—Set to `false`. 3. Do this for every IMS endpoint that is to be integrated into the contact center.
3. Configure the S-CSCF Trunk.	Go to SIP Server `switch` > `DNs` folder. 1. Create a DN of type `Trunk`. 2. In the `TServer` section, configure the following options: ◆ `contact`—Set to the IP address of the S-CSCF. ◆ `override-domain`—Set to the value of the domain served by SIP Server used in this particular IMS deployment. ◆ `refer-enabled`—Set to `false`. ◆ `sip-route`—Set to a valid SIP URI for the S-CSCF. ◆ `enable-ims`—Set to `true`.

Note: Table 66 describes the minimum mandatory options only. Other options, including IMS-specific options, may be configured as well.

For a list of all IMS-related options, see "IMS-related options" in the Index.

Instant Messaging

SIP Server supports the Instant Messaging (IM) media type as follows:

- A special SDP message for the IM media type is supported within the SIP `INVITE` dialog.

  SIP Server generates IM messages with SDP in the form accepted by Microsoft Live Communication Server, Office Communicator, and MS RTC stack.[1]

  SIP Server supports standard SIP call flows for IM, and the SIP `INVITE` messages are assumed to be the same for the IM sessions. SIP Server produces the same TEvents for the IM sessions as it does for voice calls.
- A SIP `MESSAGE` request method is supported within an established `INVITE` dialog to exchange instant messages within a SIP session.
- The content of an instant message can be distributed via a `EventUserEvent` message to a DN of type `Communication DN` with the name `gcti::im`.

  Any application that need to receive the content of any instant message must register this DN. When a SIP Server `MESSAGE` request is received by the SIP Server IM session, SIP Server informs its clients by distributing an `EventUserEvent` message. The client applications can then perform additional tasks in response to the IM content.

Support for multi-site and BC deployments added

Starting with release 8.1.101.97, Instant Messaging (IM) functionality is supported in multi-site and Business Continuity deployments. See “Instant Messaging For Multi-Site Calls” on page 255.

Instant Messaging Transcript

SIP Server supports exchanging instant messages via T-Library within the established call context. This enables applications, such as Agent Desktop, to display instant messages that were sent or received during a conversation, including a chat transcript, and send an instant message to other parties on the call.

The instant message is delivered to a T-Library client via `EventPrivateInfo` messages. A T-Library client can send an instant message using a T-Library request `TPrivateService`.

SIP Server distributes the `EventPrivateInfo` message when one of the participants in the call sends an instant message. The `EventPrivateInfo` message is sent to all other participants in the call. `AttributeExtensions` of the `EventPrivateInfo` message contains information about the instant message.

1.This format is described in the document [MS-CONFIM]: Centralized Conference Control Protocol: Instant Messaging Extensions, currently located at http://msdn.microsoft.com/en-us/library/cc431500(v=office.12).

SIP Server distributes `EventPrivateInfo` messages with the Instant Messages Transcript when a new participant is added to the call as a result of a transfer or conference operation. The Instant Messages Transcript contains all instant messages that were previously exchanged between the participants in the call. `EventPrivateInfo` with the transcript is sent to all call participants after the transfer/conference completion.

The following keys are supported:

- `im`—Contains the text of the instant message.
- `im-content-type`—Contains the value provided in the `Content-Type` header of the SIP message that delivered IM.
- `im-transcript`—Contains the transcript of the IM call. The value of this extension is a string containing an XML document that complies with the Genesys Multimedia Chat Transcript Schema.

Note: Transcript data delivery is only supported via T-Library.

Supported Call Operations

This section describes call operations within an IM session.

Preview Interaction

SIP Server supports sending a preview interaction to an IM-enabled DN, allowing the IM user to accept or reject the request before SIP Server establishes the actual IM session. With preview interaction for IM enabled, SIP Server processes the IM call as follows:

1. The IM client initiates an IM to a `Routing Point` DN on SIP Server.
2. The strategy routes the call to the IM-enabled DN. The `RequestRouteCall` includes the `Preview` key-value pair in the `Extensions` attribute.

   For example,

   ```
   AttributeExtensions [75] 00 02 00 00..
        'Preview'(list) 'Accept call?'  'respond 'yes' or 'no''
   ```

3. SIP Server initiates a preview IM to the selected DN. The preview includes the following message:

   ```
   gsipudata@<IP_address>
   Preview
   Accept call?: respond 'yes' or 'no'
   ```

4. If the DN responds with a `'yes'`, SIP Server initiates the main IM dialog.

   If the DN responds with a `'no'`, SIP Server returns the call to the routing point.

To enable this preview mechanism, set the `preview-interaction` option on the IM DN to `chat`.

Direct Calls

SIP Server supports direct 1pcc and 3pcc calls as follows:

- Direct 1pcc calls—SIP Server processes direct 1pcc calls with IM media the same way as voice calls. SIP Server will specify `AttributeMediaType=5` (`TMediaChat`) in TEvents for such calls.

  Note: Only customers outside the contact center can initiate 1pcc IM sessions. In this case, the chat window does not open until after the call is routed to an agent.

- Direct 3pcc calls—SIP Server processes a `TMakeCall` request for a call with IM media, when it is specified in the `Extensions` attribute with the `chat` key containing a value of `true`. If it is not specified, SIP Server will process the `TMakeCall` request in the normal fashion.

  Note: For 3pcc IM calls from an Agent to a Knowledge Worker (as part of the UC Connect solution), the chat window can open while the call is still being handled by the routing strategy, and before the session is accepted and delivered to the Knowledge Worker. In this case, Genesys recommends that agents do not start entering information in the chat window until after the Knowledge Worker successfully enters the session, otherwise this information may not be delivered to the targeted Knowledge Worker.

Hold

SIP Server supports the Hold operation in the same manner as for voice calls. The `Hold` request has no effect on the SIP signaling level for IM calls. SIP Server will not re-`INVITE` SIP endpoints with different SDP dialogs when performing the `Hold` operation, because no changes in the SDP dialog are necessary.

Note: SIP Server does not support Music On Hold (MOH) for IM calls.

Transfer

SIP Server supports transfers in the same manner as for voice calls. However there are some exceptions:

- SIP endpoints that are already part of a call are not considered for `re-INVITE` requests.
- The IM SDP dialog is used to `INVITE` the SIP endpoint that is the call transfer destination.

Single-step transfers are supported for non-conference calls. Single-step transfer of the conference is not supported.

Two-step transfers are supported for all calls. Agents can exchange IM text with each other during the consultation call. These IM text messages are not visible to the customer.

Conference

SIP Server supports conference for calls made within the IM session, and it is performed in the same manner as for voice calls. However, SIP endpoints that are already part of a call are not considered for re-`INVITE` requests. Also, unlike regular voice calls, SIP Server does not use Media Server to establish conference calls. SIP Server establishes an IM conference between the SIP endpoints itself, by issuing SIP `MESSAGE` requests to all conference participants.

Routing

SIP Server supports routing of IM calls using Universal Routing Server (URS) in the same manner as for voice calls. Multiple IM calls can be routed to the same agent. To achieve that, SIP Server distributes `AttributeMediaType` set to `5` (`TMediaChat`) in TEvents for IM calls. URS and Stat Server can distinguish IM calls by this attribute and, according to the configuration, route many calls with IM media to the same agent.

Treatments

SIP Server supports treatments for calls with IM media. However, unlike voice calls, SIP Server does not use Media Server to apply treatments. Instead, SIP Server executes treatments for IM calls itself by using the SIP `MESSAGE` request in accordance with the treatment request parameters.

SIP Server supports the following treatments for the IM calls:

- `PlayAnnouncement`
- `CollectDigits`
- `PlayAnnouncementAndDigits`

The `PlayAnnouncement` treatment is performed by SIP Server when it sends the `MESSAGE` request to the caller. The `MESSAGE` request is sent for each prompt specified in the `TreatmentPlayAnnouncement` request. The content of the `MESSAGE` request is created using the `TEXT` parameter in the prompt specified by the `TreatmentPlayAnnouncement` request. When processing the `TreatmentPlayAnnouncement` request for IM calls, SIP Server supports the `TEXT` parameter only.

The `CollectDigits` treatment is performed by SIP Server when it receives the `MESSAGE` request from the caller, and it then sends the complete content of the `MESSAGE` request as collected digits to URS. When processing the

`TreatmentCollectDigits` request for IM calls, SIP Server supports the `TOTAL_TIMEOUT` parameter only. All other parameters are not supported.

The `PlayAnnouncementAndDigits` treatment is performed by SIP Server by performing `TreatmentPlayAnnouncement` and then `TreatmentCollectDigits`.

Note: It is recommended that you start a routing strategy with `TreatmentCollectDigits` when applying treatments to an IM call because the strategy can collect the details of the initial IM and store it as `UserData`.

Supervision

SIP Server supports supervision functionality for IM calls, as described in "Call Supervision" on page 139. T-Library messaging for supervisor monitoring scenarios for IM calls is the same as for voice calls, with the only exception that, for IM calls, `AttributeMediaType` is set to `5` (`TMediaChat`).

SIP Server supports the following call supervision modes for IM calls:

- Silent monitoring
- Whisper coaching
- Open supervisor presence

SIP Server supports the following call supervision scopes for IM calls:

- Agent
- Call

SIP Server supports the following call supervision types for IM calls:

- One call
- All calls

Silent Monitoring for IM Calls

When silent monitoring is applied to the IM call, SIP Server uses the following algorithm to distribute instant messages:

- Instant messages sent by a caller or an agent are visible to all participants: the caller, the agent, and the supervisor.
- Instant messages sent by a supervisor are not visible to the caller or the agent.

Whisper Coaching for IM Calls

When whisper coaching is applied to the IM call, SIP Server uses the following algorithm to distribute instant messages:

- Instant messages sent by a caller or an agent are visible to all participants: the caller, the agent, and the supervisor.
- Instant messages sent by a supervisor are visible to the agent only.

Open Supervisor Presence for IM Calls

When open supervisor presence is applied to the IM call, SIP Server uses the following algorithm to distribute instant messages:

- Instant messages sent by caller or agent are visible to all participants: the caller, the agent, and the supervisor
- Instant messages sent by supervisor are also visible to the all participants: the caller, the agent, and the supervisor

This mode is the same as the normal IM conference mode.

Multiple Instant Messaging Sessions

SIP Server supports the handling of several simultaneous IM sessions by one agent. You can define the maximum number of simultaneous sessions that are distributed by the Universal Routing Server to a particular agent by using a capacity rule for that agent. For more information about capacity rules, see the *Universal Routing 8.1 Reference Manual*.

The agent can also handle one voice call and several IM sessions at once.

Instant Messaging For Multi-Site Calls

Starting with release 8.1.101.97, IM functionality is supported in multi-site and Business Continuity deployments. The IM functionality is performed through a T-Library client (Workspace Desktop). When an agent at the desktop makes an IM input, SIP Server receives a `TPrivateService` request. The IM is delivered to the desktop via `EventPrivateInfo` messages. A SIP `INVITE` dialog establishes the IM session between SIP Servers, and a SIP `MESSAGE` message delivers the IM sentence.

Supported Call Operations

The following call operations are supported within an IM session for multi-site calls:

- Direct calls between agents using `TMakeCall`
- Routing
- Treatments
- Supervision

See “Multi-site or Business Continuity Deployments” on page 256 for information about how to configure the IM in those deployments.

See also limitations for the IM for multi-site calls on page 257.

Feature Configuration

Processing UserData

The following DN-level options are used to configure how `UserData` is processed:

- `user-data-im-enabled`
- `sip-signaling-chat`
- `sip-chat-format`

Configuring Microsoft Office Communication Server

Microsoft Office Communicator is the client of Microsoft Office Communication Server 2007 R2 (OCS 2007 R2). There is no direct communication link between SIP Server and Office Communicator; OCS 2007 R2 is the bridge in this scenario. As such, SIP Server is configured to register with OCS.

Note: Starting in 8.0.3, SIP Server supports integration with Microsoft Office Communications Server 2007 R2. For information about configuring IM and presence with this server, see "Presence Integration with Microsoft Office Communications Server 2007" on page 331.

Configuring the Instant Messaging Solution

There are a number of ways to implement the Instant Messaging solutions in Genesys. This includes enabling DNs in your contact center to handle instant messages after they arrive at SIP Server. Deploying the Instant Messaging solution requires configuring various Genesys components. For detailed information, see the *Genesys 7.6 Instant Messaging Solution Guide,* which consolidates possible Instant Messaging solutions and configuration information for each of them.

Multi-site or Business Continuity Deployments

- On the Instant Messaging DN, in the `TServer` section, set the `sip-signaling-chat` option to `none`, so no SIP session with an agent endpoint is created for the IM call.
- For the IM solution to work, make sure the following configuration options are enabled (set to `true`) in the Workspace Desktop and Stat Server applications:
  - `multimedia`
  - `voice`

Note: If a URS/ORS application (a strategy) dedicated to serve IM calls uses `CollectDigits` or `PlayAnouncementAndDigits` treatments, the processing of these treatments should be started after the first `EventPrivateInfo` is received in the application's session. The `SuspendForEvent` URS function will suspend the strategy execution until URS receives the specified event. The `Type` parameter of the `SuspendForEvent` function must be set to the integer value `150` for `EventPrivateInfo`.

Instant Messages Encoding

SIP Server supports encoding of instant messages, received in SIP messages, from UTF-8 to a local character set and vice versa in T-Library messages. To enable instant messages encoding, in the `TServer` section of the SIP Server Application object, configure the following options:

- `encoding`—Activate Unicode support.
- `encoding-area`—Must include the `chat` area.

Feature Limitations

- The `Page` mode for Instant Messaging is not supported.
- In multi-site deployments, the `route` or `direct-uui` ISCC transaction types are required.
- Instant Messaging transfers are not supported in multi-site deployments.
- Instant Messaging conferences are not supported in multi-site deployments.
- When an IM call is routed across sites, SIP Server will pass the IM transcript to the remote site if an ISCC transaction precedes the actual routing (route or direct-uui), but it will not pass the IM transcript to the remote site if an ISCC transaction follows the actual routing (such as Call Overflow (COF)).

IPv6 Support

Genesys supports Internet Protocol version 6, commonly known as IPv6, as described in the *Framework Deployment Guide*. The implementation of IPv6 in Genesys is based on the following assumptions:

- Dual-stack requirement and backward compatibility
- Dual IPv4/IPv6 server sockets
- IPv4 preference for DNS

However, there are some peculiarities in how IPv6 is supported in SIP Server, compared to non-SIP related components:

- IP addresses are often explicitly inserted in SIP messages, so SIP Server is aware of IPv4 vs. IPv6 differences.
- Since SIP Server establishes communications between endpoints that are supposed to stream data to each other directly, the value of dual stack support is limited. IPv4 and IPv6 endpoints most likely are unable to talk to each other. So SIP Server configuration options target support of one type of endpoints, either IPv4 or IPv6.

To enable IPv6 support, SIP Server uses the transport address selection algorithm that conforms to RFC 3484, where local address selection is based on the destination address. SIP Server selects a local IPv6 address when its peer uses IPv6, and selects a local IPv4 address when its peer uses IPv4. An FQDN resolving to a local IPv4 or IPv6 address can be used instead of explicit addresses.

Feature Configuration

IPv6 must be enabled by setting the `GCTI_CONN_IPV6_ON` environment variable on the machine where SIP Server is installed. Refer to the *Framework 8.1 Deployment Guide* for details.

In the SIP Server Application and DN configuration, options with the IP address or host name values can be explicitly set to a valid IPv6 address. Note that an IPv6 address is always placed between square brackets [ ]. You can also configure the `sip-fqdn-ip-version` configuration option to specify a preferred IP version for FQDN resolution.

IPv6 Support by Genesys Components

SIP Server can work with Genesys Media Server using IPv6. DNs of type `Voice over IP Service`, for media services performed by GVP, must be configured with the `contact` option set to the Resource Manager explicit IPv6 address. For more information about GVP component configuration, refer to the *Genesys Voice Platform 8.1 Deployment Guide* for details.

SIP Server can also work with SIP Proxy using IPv6. No changes are required in SIP Proxy.

High-Availability Considerations

For IP Address Takeover on Windows and Linux operating systems, to support IPv6, you must use specific HA scripts. Refer to the latest *SIP Server 8.1 High-Availability Deployment Guide* for details.

Feature Limitations

- 1pcc transfer by `REFER` with `Replaces` may fail if SIP Server receives a SIP `REFER` request with the `Replaces` parameter in the `Refer-To` header pointing to a dialog whose SIP `Call-ID` contains a % (percentage) character. This character may appear as part of the IPv6 address scope ID, and sometimes IP addresses are used as part of the SIP `Call-ID` header.
- IPv6 is only supported on Windows and Linux operating systems.
- Support of dual stack is limited.

Keep Alive for TCP Connections

Introduced in SIP Server 8.1.101.43

SIP Server provides the ability to detect stale TCP connections between SIP Server and a SIP device using the TCP keep-alive mechanism. This functionality is recommended for those environments in which SIP endpoints are located behind a firewall that is configured to drop inactive TCP connections silently and without sending any notification to SIP Server. If SIP Server tries to use a stale connection to initiate a new call or to execute call control, the attempt would fail. As a result, the SIP endpoint is placed to out of service.

When the TCP keep-alive mechanism is enabled, SIP Server sends keep-alive packets for all existing SIP connections. If there is no response for a configured time interval, and if there is an active transaction for this connection, SIP Server attempts to reopen the connection immediately and re-sends the last SIP request. If the connection does not have an active transaction, then it will be reopened only when a new transaction is initiated. If an attempt to open a connection for an active transaction fails, SIP Server releases the call.

For this feature to work with TLS over TCP, the SIP Endpoint must be able to accept the connection when SIP Server attempts to reopen it.

The TCP keep-alive mechanism does not replace the active OOS check, which should be configured as usual even if the TCP keep-alive feature is enabled.

Feature Configuration

1. Configure TCP keep-alive timeouts for your operating system. You can use the following links for your reference:
   - For Windows, see:
     `http://technet.microsoft.com/en-us/library/cc957549.aspx`
   - For Linux, see:
     `http://tldp.org/HOWTO/TCP-Keepalive-HOWTO/usingkeepalive.html`
2. In the `TServer` section of the SIP Server Application, configure the `sip-enable-tcp-keep-alive` configuration option to enable the TCP keep-alive functionality.

Feature Limitation

For `Voice over IP Service` DNs, SIP Server does not attempt to reopen the connection within an active transaction.

Mapping Treatment Errors

When SIP Server receives a SIP error from a media server (for example, from GVP), it maps both the error code and error description into the `Extensions` Attribute of the corresponding `EventTreatmentNotApplied` message.

For example, when integrated with GVP for a URS-centric call flow (routing strategy controls the call), a failed PlayApplication treatment can result in either a SIP error message or an MSML error message. SIP Server maps these error messages, along with their description, to the resulting `TreamentNotApplied` message.

Sample SIP Error Mapping

SIP Server receives a `TApplyTreatment` request and translates that to a SIP `INVITE` request to GVP, with details to start a treatment. If the treatment fails to start, GVP responds with a SIP 4xx/5xx/6xx error code, which may also include a `Warning` header with an error description.

SIP Server sends a `TreatmentNotApplied` message in response to the original `TApplyTreatment` request, with the mapped key-value pairs included in the `Extensions` Attribute:

```
Attribute ErrorCode 50
AttributeExtensions
   SipResponseCode 4xx/5xx/6xx
   ResponseDescription "description"
```

Sample MSML Error Mapping

For MSML-based treatments, a failed treatment can result in either a SIP error or MSML error. For example, GVP may respond to a failed treatment by sending the following MSML error:

```
<result response="423"/>
<description>error description</description>
```

From these fields, SIP Server maps the `response` value in the `result` tag, as well as the error description in the `description` tag to the Extensions Attribute of the `TreatmentNotApplied` message.

For example,

```
Attribute ErrorCode 50
AttributeExtensions
   MsmlResponseCode 423
   ResponseDescription <description>
```

For SIP errors, the same mapping takes place as it does for the NETANN sample above.

Feature Configuration

This feature is enabled by default. No configuration is required.

Mapping SIP Headers and SDP Messages

SIP Server can extract data from some incoming SIP messages and map it to either an `Extensions` or `UserData` attribute in T-Library event messages. SIP Server can map T-Library request attributes (passed in the `TRouteCall` message) to SIP parameters in the outgoing `INVITE` message. SIP Server can also map the whole SDP message body, or any particular line in it as `Extensions` or `UserData` attributes.

Table 67 summarizes mapping SIP headers from SIP messages to T-Library attributes and mapping T-Library attributes to SIP messages.

Table 67: SIP Headers Mapping Summary

Configuration Parameters	Direction	Description
SIP-to-T-Library Mapping		
`userdata-map-trans-prefix` (Application-level option, `TServer` section),	SIP `INVITE` -> T-Lib `UserData`	SIP headers from the `INVITE` message that start with the configured prefix in `userdata-map-trans-prefix` are mapped to the `UserData` attribute of `EventRouteRequest` without that prefix. See "INVITE Messages" on page 265 and "Mapping Examples from INVITE Messages" on page 266.
`userdata-map-trans-prefix` (Application-level option, `TServer` section),	SIP `INFO, BYE, UPDATE` -> T-Lib `UserData`	SIP headers from the `INFO, BYE,` and `UPDATE` messages that start with the configured prefix in `userdata-map-trans-prefix` are mapped to the `UserData` attribute of `EventAttachedDataChanged` without that prefix. See "INFO, UPDATE, and BYE Messages" on page 267.
`userdata-map-trans-prefix` (Application-level option, `TServer` section),	SIP `REFER` -> T-Lib `UserData`	SIP headers from the `REFER` message that start with the configured prefix in `userdata-map-trans-prefix` are mapped to the `UserData` attribute of events involved in a single-step transfer without that prefix. See "REFER Messages" on page 269.

Table 67: SIP Headers Mapping Summary (Continued)

Configuration Parameters	Direction	Description
`userdata-<n>` (Application-level option, `INVITE` section)	SIP `INVITE` -> T-Lib `UserData`	A SIP header configured in `userdata-<n>` is mapped from the `INVITE` message to the `UserData` attribute of `EventRouteRequest` as a key of the `UserData` key-value pair. A SIP header value is mapped as a value of this key-value pair. See "INVITE Messages" on page 265 and "Mapping Examples from INVITE Messages" on page 266.
`userdata-<n>` (Application-level option, `INFO` section)	SIP `INFO` -> T-Lib `UserData`	A SIP header configured in `userdata-<n>` is mapped from the `INFO` message to the `UserData` attribute of `EventAttachedDataChanged` as a key of the `UserData` key-value pair. A SIP header value is mapped as a value of this key-value pair. See "INFO, UPDATE, and BYE Messages" on page 267.
`userdata-<n>` (Application-level option, `BYE` section)	SIP `BYE` -> T-Lib `UserData`	A SIP header configured in `userdata-<n>` is mapped from the `BYE` message to the `UserData` attribute of `EventAttachedDataChanged` as a key of the `UserData` key-value pair. A SIP header value is mapped as a value of this key-value pair. See "INFO, UPDATE, and BYE Messages" on page 267.
`userdata-<n>` (Application-level option, `REFER` section)	SIP `REFER` -> T-Lib `UserData`	A SIP header configured in `userdata-<n>` is mapped from the `REFER` message to the `UserData` attribute of events involved in a single-step transfer as a key of the `UserData` key-value pair. A SIP header value is mapped as a value of this key-value pair. See "REFER Messages" on page 269.
`extensions-<n>` (Application-level option, `INVITE` section)	SIP `INVITE` -> T-Lib `Extensions`	A SIP header configured in `extensions-<n>` is mapped from the `INVITE` message to the `Extensions` attribute as a key of the `Extensions` key-value pair. A SIP header value is mapped as a value of this key-value pair. See "INVITE Messages" on page 265.

Table 67: SIP Headers Mapping Summary (Continued)

Configuration Parameters	Direction	Description
`extensions-<n>` (Application-level option, `REFER` section)	SIP `REFER` -> T-Lib `Extensions`	A SIP header configured in `extensions-<n>` is mapped from the `REFER` message to the `Extensions` attribute as a key of the `Extensions` key-value pair. A SIP header value is mapped as a value of this key-value pair. See "REFER Messages" on page 269.
`EXTRACT_SIP_HEADERS` (extension key)	SIP `INVITE` -> T-Lib `Extensions`	This extension key specified in the `TMakeCall`, `TInitiateTransfer`, and `TInitiateConference` requests instructs SIP Server to extract specified SIP headers from an `INVITE` sent to an originating party and map them to the `Extensions` attribute of `EventDialing` generated for this party. See "Using EXTRACT_SIP_HEADERS" on page 270.
T-Library-to-SIP Mapping		
`userdata-map-trans-prefix` (Application-level option, `TServer` section), `userdata-map-filter-mode` (Application-level option, `TServer` section), `userdata-map-filter` (DN-level option)	T-Lib `UserData` -> SIP	When a prefix (or a list of prefixes) is specified in `userdata-map-filter` and it matches the initial characters of the key in the `UserData` key-value pair, then SIP Server either allows or blocks mapping of `UserData` into SIP messages, based on the setting in the `userdata-map-filter-mode` option. When `userdata-map-filter=*` set on a MSML DN, SIP Server passes or blocks all `UserData` key-value pairs to GVP. For GVP, use `userdata-map-trans-prefix` to specify the prefix `X-Genesys-` to be added to a SIP message, otherwise GVP will not process it.
`SIP_HEADERS` (extension key)	T-Lib `Extensions` -> SIP `INVITE`	The value of the `SIP_HEADERS` extension in `TRouteCall` instructs SIP Server to include the corresponding `Extensions` attribute as the SIP headers in the `INVITE` message. See "Using SIP_HEADERS and SIP_REQUEST_PARAMETERS" on page 273.

Table 67: SIP Headers Mapping Summary (Continued)

Configuration Parameters	Direction	Description
SIP_HEADERS (extension key)	T-Lib Extensions -> SIP REFER	The value of the SIP_HEADERS extension in TRouteCall instructs SIP Server to include the corresponding Extensions attribute as the SIP headers in the REFER message. This applies to the scenarios when TRouteCall is executed by the SIP REFER message. See "Using SIP_HEADERS and SIP_REQUEST_PARAMETERS" on page 273.
SIP_REQUEST_PARAMETERS (extension key)	T-Lib Extensions -> SIP INVITE	The value of the SIP_REQUEST_PARAMETERS extension in TRouteCall instructs SIP Server to include the corresponding Extensions attribute to the Request-URI parameters of the INVITE message. See "Using SIP_HEADERS and SIP_REQUEST_PARAMETERS" on page 273.
extensions-<n> (Application-level option, INVITE section)	T-Lib Extensions -> SIP INVITE	The option value of extensions-<n> indicates which key of the Extensions attribute key-value pair is mapped as a new SIP header in the INVITE message. The value of this key-value pair is mapped as a SIP header value. See "Using the extensions-<n> Option" on page 272.

General Guidelines

Consider the following recommendations if both userdata and extensions could be used for mapping to and from SIP headers in your environment:

- Sensitive/confidential information arriving in SIP headers should be rather mapped to Extensions. This would not only minimize the size of the call's T-Library messages, but also mitigate the risk of sensitive information distribution across all T-Library clients in each call-related message.
- Mapping of Extensions to SIP headers minimizes the administrative overhead (eliminates the configuration step); such mapping is fully controlled by the routing application.
- The UserData key-value pair mapped to INVITE and sent in the SIP header to an external IVR can be later updated when REFER from the IVR arrives and the SIP header of the REFER message is mapped to the same UserData.

- For UDP communications, the rule of thumb is to keep the size of the data sent via an SIP header under 300 bytes.

From SIP Messages to T-Library Messages

SIP Server processes SIP messages and can map related data to T-Library event attributes as described in this section. This information becomes further available to other Genesys Framework components. This functionality is supported for following SIP messages:

- INVITE
- INFO
- UPDATE
- REFER
- BYE

Note: For proper processing of SIP header values of incoming from GVP SIP messages and mapping them into UserData, the special characters such as commas (,) in the headers' value must be encoded.

INVITE Messages

SIP Server can extract data from an incoming INVITE message and send the data to Universal Routing Server if the call is made at a Routing Point. Data is retrieved as values from the SIP headers and parameters of the INVITE message and then populated into the Extensions or UserData attributes in the EventRouteRequest message. You can configure which SIP headers and SIP header parameters to extract data from by creating a section that corresponds to a SIP method name on the Options tab of the SIP Server Application object.

For example, the INVITE section lists the SIP headers that are extracted from the SIP INVITE message.

The option names are:

- extensions-<n>
- userdata-<n>

The option extensions-<n> instructs SIP Server to include the corresponding SIP header or SIP header and its parameter, if configured, into the Extensions attribute. The option userdata-<n> instructs SIP Server to include the corresponding SIP header into the UserData attribute.

The n value is a number and must be unique for all option names containing the same prefix (extensions or userdata).

For example:

```
[INVITE]
extensions-1=From
extensions-2=To
```

A SIP header name is mapped as a key of the attribute key-value pair, and a SIP header value is mapped as a value of this key-value pair. See "Mapping Examples from INVITE Messages" on page 266.

For `extensions-<n>` options, you can use the colon character (`:`) to include the parameter name of a SIP header.

For example: `extensions-1=From:tag`

The SIP method (`INVITE`) must be used instead of the header name if you want to populate the parameter from the SIP Request-line parameter.

If the names of the headers in the `INVITE` message start with the prefix configured in the `userdata-map-trans-prefix` option, these headers will be mapped to `UserData` without that prefix.

Mapping occurs only if both conditions are true:

- The SIP header or header parameter is contained within the incoming `INVITE` message.
- The SIP header or header parameter is configured within the `INVITE` section.

As a result of mapping, the following key-value pair will be created within the `EventRouteRequest` message:

- The key in the attribute will be equal to the value of the configuration option.
- The value of the attribute will be equal to the value of the header or header parameter in the SIP `INVITE` message.

Note: Data from an incoming SIP `INVITE` message received on a Routing Point (via ISCC routing) that is located on the destination SIP Server cannot be extracted to `UserData`. In this case, use extraction to `Extensions` to obtain data from the `INVITE` message.

Mapping Examples from INVITE Messages

Example 1: userdata-map-trans-prefix

1. The `userdata-map-trans-prefix` option is set to `X-Genesys-`.
2. The `INVITE` message arrives containing the `X-Genesys-` header.

```
INVITE sip:+14085507046@srv-sip.example.com SIP/2.0
...
X-Genesys-CallUUID: 011T7L73KCD4RD44E88362LAES00BLCM
...
```

3. The SIP header is mapped to `AttributeUserData` as follows:

```
message EventRouteRequest
    ...
    AttributeUserData       00 01 00 00..
       'CallUUID'           '011T7L73KCD4RD44E88362LAES00BLCM'
    ...
```

Example 2: userdata-<n>

1. The `userdata-1` option is set to `X-Genesys-CallUUID`.
2. The `INVITE` message arrives containing the `X-Genesys-CallUUID` header.

```
INVITE sip:+14444447000@srv-sip.domain.com SIP/2.0
...
X-Genesys-CallUUID: 011T7L73KCD4RD44E88362LAES00BLCM
...
```

3. The SIP header is mapped to `AttributeUserData` as follows:

```
message EventRouteRequest
    ...
    AttributeUserData       00 01 00 00..
        'X-Genesys-CallUUID' '011T7L73KCD4RD44E88362LAES00BLCM'
    ...
```

Example 3: extensions-<n>

1. The `extensions-1` option is set to `From:tag`.
2. The `INVITE` message arrives containing the `From` header with the `tag` parameter:

```
INVITE sip:+14444447000@srv-sip.domain.com SIP/2.0
...
From: sip:5000@external_domains:37434;tag=230BCC6E-8B5C-49D3-9708-
ED4985EE719C-29
...
```

3. The SIP header and its parameter are mapped to `AttributeExtensions` as follows:

```
message EventRouteRequest
    ...
    AttributeExtensions        [97] 00 03 00 00..
        'From:tag' '230BCC6E-8B5C-49D3-9708-ED4985EE719C-29'
    ...
```

INFO, UPDATE, and BYE Messages

SIP Server can generate an `EventAttachedDataChanged` message if it receives SIP `INFO`, `UPDATE`, or `BYE` messages. Both the `UserData` attribute in the event and the corresponding call can contain information from the SIP message.

SIP Server supports two methods for mapping user data from `INFO`, `UPDATE`, and `BYE` messages into the `UserData` attribute in the T-Library event:

- Mapping data from SIP headers
- Mapping data from the body of `INFO`, `UPDATE`, and `BYE` messages
- Mapping data from the MSML body of the `INFO` message

Mapping Data from SIP Headers in INFO, UPDATE, and BYE Messages

For this method, you must configure the `INFO` and `BYE` sections in the SIP Server `Application` object to map data from these SIP messages to the corresponding T-Library events. The rules for this configuration are the same as the rules for configuring the `INVITE` section, on page 265.

Also, if the headers from `INFO`, `UPDATE`, and `BYE` messages start with the prefix configured in the `userdata-map-trans-prefix` option, these headers will be mapped to `UserData` as well.

Note: Only mapping to the `UserData` attribute is supported for the `INFO` and `BYE` messages. Therefore, the `INFO` and `BYE` sections can contain `userdata-<n>` options, but they may not contain `extensions-<n>` options.

Mapping Data from the Body of INFO, UPDATE, and BYE Messages

For this method, you must configure the application that sends the SIP message to include the following header:

```
Content-Type: application/x-www-form-urlencoded
```

The presence of this header instructs SIP Server to parse the body of the SIP message, extracting any user data that it finds there and adding it to the `UserData` attribute in the T-Library message.

Note: The Genesys Voice Platform uses this `Content-Type` header method for sending user data in the body of `INFO`, `UPDATE`, and `BYE` messages, which SIP Server then maps to the corresponding T-Library event. When modified user data is received in the body of `INFO`, `UPDATE`, and `BYE` messages, SIP Server updates the user data in the corresponding T-Library event. If a key, which must be mapped from a SIP message to T-Library user data, is already present in the user data, then the value of this key in the user data is updated with the one received in the SIP message. In the case when `UserData` contains several KVPs with the key being mapped, all those keys are removed and the only one left has the value received in the SIP message. For integration with GVP, no special configuration on GVP or on the SIP Server switch is required.

Mapping Data from the MSML Body of INFO Message

For this method to work, the SIP `INFO` message must contain the following header:

```
Content-Type: application/vnd.radisys.msml+xml
```

In addition, you must set the `msml-support` option to `true` in the SIP Server Application object to map data from MSML messages to the corresponding

T-Library events. This parameter instructs SIP Server to parse the body of the MSML message, extracting any user data that it finds there and adding it to the `UserData` attribute in the T-Library message.

SIP Message Example

```
INFO sip:21001@UTE_HOME:11000 SIP/2.0
From: sip:SVC_Mediaserver@UTE_HOME:11000;tag=467B7835-E194-435E-
B8B4-B82764E327CF-3
To: sip:21001@172.24.128.63:49904;tag=EC11FCB4-2983-4A3B-8A93-
A15A72D878B8-8
Call-ID: 7E465A2A-F623-4ECF-AC3A-63D3FE271F25-4@UTE_HOME
CSeq: 1 INFO
Content-Length: 239
Content-Type: application/vnd.radisys.msml+xml
Via: SIP/2.0/UDP 172.24.128.63:49903;branch=z9hG4bK807562CD-F67E-
403E-9F26-A003FCCEAC87-3
Contact: <sip:172.24.128.63:49903>

<?xml version="1.0" encoding="UTF-8"?>
<msml version="1.1">
<event name="msml.dialog.exit" id="conn:__MSML-CONN-
ID__ivr_application9>
<name>AccessID</name>
<value>11447700</value>
<name>Status</name>
<value>working</value>
</msml>
```

T-Library Message Example

```
EventAttachedDataChanged
    AttributeEventSequenceNumber         000000000000015a
    AttributeTimeinuSecs  137000
    AttributeTimeinSecs    1433151612    (15:10:12)
    AttributeThirdPartyDN '5000'
    AttributeThisDNRole    2
    AttributeThisDN        '5000'
    AttributeANI           '21001'
    AttributeDNIS          '5000'
    AttributeUserData       [41]      00 02 00 00..
            'AccessID'        '11447700'
            'Status'       'working'
    AttributeCallUUID      '0MS6PSNRF120593DHFFIR0AVC0000001'
    AttributeConnID         226502665c77b001
    AttributeCallID         16778217
    AttributePropagatedCallType 2
    AttributeCallType       2
```

REFER Messages

The SIP `REFER` method provides single-step transfer functionality. SIP Server retrieves data from headers and parameters of the `REFER` message, and then populates it into the `Extensions` or `UserData` attributes in all events associated

with a transfer transaction. This feature also enables Genesys Voice Platform (GVP) to transfer a call to Genesys Framework with attached data.

When a REFER message arrives, SIP Server analyzes the user part of the REFER TO URI to determine if the destination of the call is an internal DN or an external destination. SIP Server checks if any configuration mapping is provided in the REFER section of SIP Server Application object. If such mappings exist, SIP Server extracts the values of the appropriate SIP headers into the attribute Extensions or UserData, according to the configuration. If a key, which must be mapped from a SIP message to T-Library user data, is already present in the user data, then the value of this key in the user data is updated with the one received in the SIP message. In the case when UserData contains several key-value pairs with the key being mapped, all those keys are removed and the only one left has the value received in the SIP message. If the destination is a Routing Point, after this processing, SIP Server generates an EventRouteRequest message containing the necessary attribute values.

You can configure which headers and parameters to extract data from by creating the REFER section on the Options tab of the SIP Server Application object. The rules for this configuration are the same as the rules for configuring the INVITE section (see page 265).

To map data from the REFER message into the Extensions and/or UserData attributes, configure extensions-<n> options and/or userdata-<n> options in the REFER section.

If the names of the headers in the REFER message start with the prefix configured in the userdata-map-trans-prefix option, these headers will be mapped to UserData.

It is possible to configure SIP Server to extract data from custom headers added in the REFER message, and to process the data from the custom SIP headers.

Feature Limitation

This feature is applicable only to scenarios where a call is made to a Routing Point using the REFER method. To pass attached data in other scenarios, use the mapping configuration of the INFO and UPDATE messages instead.

Using EXTRACT_SIP_HEADERS

In addition, you can use the EXTRACT_SIP_HEADERS extension key in the TMakeCall, TInitiateTransfer, and TInitiateConference requests to extract specified headers from an INVITE sent to an originating party and map them to AttributeExtensions of the EventDialing message generated for this party.

Example

The following log excerpt provides the details in which 7000 is a TMakeCall originating party, 8000 is a destination party:

1. SIP Server processes the TMakeCall request containing the EXTRACT_SIP_HEADERS key to extract headers P-Charging-Vector and From from the INVITE request:

```
message RequestMakeCall
    AttributeThisDN         '7000'
    AttributeConnID         006e01886c3d7001
    AttributeOtherDN'       8000'
    AttributeExtensions
        'EXTRACT_SIP_HEADERS' 'P-Charging-Vector,From'
    AttributeReferenceID    9
```

2. SIP Server extracts headers P-Charging-Vector and From from the INVITE request:

```
INVITE sip:7000@DestinationHost:8000
From: <sip:8000@SourceHost:8000>;tag=28B10B44
To: <sip:7000@DestinationHost>
Call-ID: 931E620E-F3F9
CSeq: 1 INVITE
Content-Length: 0
Contact: <sip: SourceHost >
Max-Forwards: 70
Session-Expires: 1800;refresher=uac
P-Charging-Vector: 1234-5678-90
Min-SE: 90
Supported: timer
```

3. SIP Server inserts extracted headers and their values into EventDialing as AttributeExtensions keys:

```
message EventDialing
    AttributeThisDN         '7000'
    AttributeConnID         006e01886c3d7001
    AttributeOtherDN        '8000'
    AttributeExtensions[371] 00 0B 00 00..
        'From' '<sip:8000@SourceHost:8000>;tag=28B10B44'
        'P-Charging-Vector' '1234-5678-90'
    AttributeReferenceID     9
```

Feature Limitation

In a single-dialog mode, no new dialogs are created for a ThisDN device while SIP Server processes TInitiateTransfer or TInitiateConference requests. So, the EXTRACT_SIP_HEADERS extension keys are not included in TInitiateTransfer

or TInitiateConference requests. Dialog parameters can be extracted using the EXTRACT_SIP_HEADERS key attached to a TMakeCall request.

From T-Library Messages to SIP Messages

SIP Server can map headers or header parameters passed in the TRouteCall message to the outgoing INVITE message that is sent as a result of the call routing process. There are two ways to specify the values of the headers or header parameters to be mapped:

- Use the extensions-<n> SIP Server configuration option. This method maps only headers.
- Use the SIP_HEADERS and SIP_REQUEST_PARAMETERS extension of the TRouteCall message.

Both methods can work simultaneously; for example—you can create a mapping list using extensions-<n> options in the SIP Server configuration, and also specify the names in the SIP_HEADERS extension of the TRouteCall request.

Note: The use of the UserData key name as a SIP message header field name implies that the UserData key name should conform to the IETF RFC 3261 requirement for allowed characters in the SIP message header field name.

User data is also mapped in scenarios in which TRouteCall is involved. Refer to the "T-Library Unstructured Data" chapter of the *Genesys Events and Models Reference Manual*.

Note: SIP Server does not pass key-value pairs from a list of key-value pairs to GVP.

Using the extensions-<n> Option

To configure T-Library request attributes mapping to the SIP messages, use extensions-<n> options in a SIP Server Application object. Those options are specified in the section named after the SIP request used to route the call, which is INVITE.

Example

This example demonstrates how to map a TRouteCall request extension called InfoToSendInInvite to the outgoing INVITE message.

Follow these steps to configure SIP Server:

1. Create the INVITE section on the Options tab of the SIP Server Application object.
2. In the INVITE section, create an option named extensions-1, and set the value to InfoToSendInInvite.

Note: This configuration example is based on the assumption that this is the first extensions-<n> option in the INVITE section.

SIP Server uses this configuration when it receives a TRouteCall request from the URS with either InfoToSendInInvite or InfoToSendInRefer extensions defined. The following log excerpt provides the details:

```
message RequestRouteCall
     AttributeThisDN         '5000'
     AttributeConnID         006e01886c3d7001
     AttributeOtherDN        '21101'
     AttributeExtensions     [371] 00 0B 00 00..
       'InfoToSendInInvite'  'INVITE from SIP Server'
     AttributeDNIS           '5000'
     AttributeRouteType      1 (RouteTypeDefault)
     AttributeReferenceID    9
```

SIP Server adds a new header InfoToSendInInvite to the outgoing INVITE message:

```
INVITE sip:21101@DestinationHost:21101 SIP/2.0
From: <sip:7102@SourceHost:7102>;tag=28B10B44
To: <sip:21101@ DestinationHost>
Call-ID: 931E620E-F3F9-4D72-A451-36B1BB259532-1
CSeq: 1 INVITE
Content-Length: 145
Content-Type: application/sdp
Contact: <sip: SourceHost:5060>
InfoToSendInInvite: INVITE from SIP Server
Max-Forwards: 70
Session-Expires: 1800;refresher=uac
Min-SE: 90
Supported: timer
```

Using SIP_HEADERS and SIP_REQUEST_PARAMETERS

This method does not require any changes to the configuration of the SIP Server Application object. In this case, all mapping information is provided in the TRouteCall request that is created by the URS routing strategy and sent to SIP Server.

The TRouteCall request should contain two specialized extensions to trigger T-Library-to-SIP mapping in SIP Server:

- SIP_HEADERS—Contains a list of extension names of TRouteCall to be mapped to the outgoing SIP message as headers.
- SIP_REQUEST_PARAMETERS—Contains a list of extension names to be mapped to the outgoing SIP message as Request URI parameters.

The values of headers and header parameters must be also specified in the TRouteCall request.

The SIP_HEADERS extension can be used to map T-Library extensions to the SIP REFER message. This applies to the scenarios when TRouteCall is executed by the SIP REFER message.

Example

This example demonstrates how mapping works in SIP Server when it receives the following TRouteCall request:

```
message RequestRouteCall
    AttributeThisDN         '5000'
    AttributeConnID         006e01886c3d7001
    AttributeOtherDN        '21101'
    AttributeExtensions     [371] 00 0B 00 00..
        'SIP_HEADERS'               'hdr-host1,hdr-host2'
        'SIP_REQUEST_PARAMETERS'    'prm-host1,prm-host2'
        'hdr-host1'                 'host1'
        'hdr-host2'                 'host2'
        'prm-host1'         'local1'
        'prm-host2'         'local2'
    AttributeDNIS           '5000'
    AttributeRouteType      1 (RouteTypeDefault)
    AttributeReferenceID    9
```

This message contains both SIP_HEADERS and SIP_REQUEST_PARAMETERS extensions, which means both new headers and new Request-URI parameters should be added to the outgoing SIP message:

```
INVITE sip:21101@ DestinationHost:21101;prm-host1=local1;
prm-host2=local2 SIP/2.0
From: <sip:7102@ SourceHost:7102>;tag=28B10B44
To: <sip:21101@ DestinationHost >
Call-ID: 931E620E-F3F9
CSeq: 1 INVITE
Content-Length: 145
Content-Type: application/sdp
Contact: <sip: SourceHost >
hdr-host1: host1
hdr-host2: host2
Max-Forwards: 70
Session-Expires: 1800;refresher=uac
Min-SE: 90
Supported: timer
```

Known Limitation

When mapping T-Library extensions to SIP headers, SIP Server may add additional extension keys as SIP headers even if those keys do not exactly match a comma-separated entry listed in the value of the `SIP_HEADERS` key pair.

SDP Message Mapping

SIP Server can map the whole SDP message body, or any particular line in it as `AttributeExtensions` or `AttributeUserData`. Configuring this type of mapping is similar to configuring mapping of SIP Server headers or parameters. The option names in the `INVITE` section must use the same rules described for SIP Server message mapping; however, the option name must be `SDP`, and it must be followed by a colon and a letter that indicates the type of mapped SDP message. For example, the `userdata-1=SDP:m` option is mapped using the following parameters from `AttributeUserData`:

```
m=audio 18234 RTP/AVP 8 101
```

Note: If the `AttributeUserData` parameter in the `EventRouteRequest` message contains a pair such as:

```
SDP:m audio 18234 RTP/AVP 8 101
```

You must use the `SDP` value to configure the mapping of the whole SDP message.

Dynamic DN Replacement

SIP Server can dynamically replace the `[DN]` parameter in T-Library to SIP header mapping. If enabled, SIP Server replaces the `[DN]` pattern in the mapped SIP message with the digits of the DN where the SIP message is being sent. This applies to both `Extensions` and `UserData` T-Library to SIP mapping.

To enable dynamic DN replacement, in the `TServer` section of the SIP Server `Application` object, set the `tlib-map-replace-dn` option to `true`.

SIP Headers Encoding

SIP Server supports encoding of SIP headers from UTF-8 to a local character set and vice versa in T-Library messages. To enable SIP headers encoding, in the `TServer` section of the SIP Server Application object, configure the following options:

- `encoding`— Specify the converter name to translate UTF-8 data to the local character set.
- `encoding-area`—Must include the `tlibsip` area.

Masking Sensitive Data in SIP Messages

Introduced in SIP Server 8.1.102.51

SIP Server can now mask sensitive data in SIP messages contained in SIP Server logs. When enabled, SIP Server replaces:

- All private SIP header values with a single asterisk
- SIP message body content with the phrase `CONTENT FILTERED`

SIP Server does not replace the content of type application/sdp, and it replaces application/vnd.radisys.msml+xml in the SIP message body only when it contains user data.

Feature Configuration

To enable masking sensitive data in SIP messages, set the `x-sip-mask-sensitive-data` configuration option to `true` in the `[log]` section of the SIP Server Application.

Starting with version 8.1.103.88, SIP Server can unmask specific SIP headers contained in SIP Server logs. This feature is enabled by the `x-sip-unmask-headers` and `x-sip-unmask-headers-default` configuration options.

Media Server Reliability—NETANN/MSML

SIP Server supports reliability for music-on-hold; treatments that are set to continuous playback, conference, or supervisor; voice call recording features provided by a media server. SIP Server can provide reliability in cases where the media server:

- Fails when the service is initially requested.
- Becomes unavailable while the service is underway.

In either case, SIP Server can restart the service on a second media server, with minimal impact on the customer experience.

Note: To provide reliability for treatment services, SIP Server supports the Active Out-of-Service Detection method only. This limitation applies to treatments that are started for calls located on a Routing Point.

How SIP Server Detects a Media Server Failure

To provide reliability for media services, SIP Server must first detect when a media server becomes unavailable. It can perform this detection either of the following methods:

- Active Out-of-Service Detection.

Active detection is the only method that allows SIP Server to detect service failure before it engages the service.

- Passive Out-of-Service Detection.

  If only passive detection is enabled, SIP Server can only detect a failure after initial service engagement. In this case, if the device fails to respond to an `INVITE` request for a media service, SIP Server considers the device to be out-of-service and tries an alternate media server.

- Error Response Handling

  The media server returns an error in response to the SIP transaction. In this case, SIP Server does not place the device in an out-of-service state. However, it does try to restart the service on an alternate available media server.

  In case of multiple `Voice over IP Service` DNs with `service type=msml`:

  - When Resource Manager returns an error response, SIP Server selects an alternate `Voice over IP Service` DN (`msml`) to recover the media service.
  - When MCP returns an error response, SIP Server selects a `Voice over IP Service` DN (`msml`) on a round-robin fashion to recover the media service.

  In case of MSML conference, when Media Server responds with a `503` response, SIP Server does not retry the conference, and drops the conference request.

- The media server application terminates the dialog due to a lost RTP stream or other error.

  Termination of a SIP dialog by a continuous service (such as MCU, recorder, or treatment that is set to continuous playback) is considered a failure of the corresponding endpoint. In this case, SIP Server does not consider this to be a media server failure. SIP Server will restore the call without the failed party. For example, in a conference call, SIP Server will release the failed party and then restore the call, minus that party. For emergency call recording, SIP Server restores the call but does not restart recording.

Reliability for Conference Calls

If the Media Server (acting as an MCU) that is supporting a conference call is detected as unavailable while the conference is in progress, SIP Server detects the failure and restarts the conference on a secondary media server with minimal impact on the customer experience.

Reliability for Supervisor Features

In case of Media Server failure, SIP Server restarts the Supervisor features on a second Media Server, with minimal impact on the customer experience. See “Call Supervision” on page 139 for more information.

Reliability for Voice Call Recording

SIP Server can provide continuity for voice call recording. In cases where the media server that is supplying the recording function is detected as unavailable, SIP Server can continue recording the call on a second alternate media server with minimal loss of content.

The recording is captured in two separate files:

- The first part of the call is recorded to a file on the original media server.
- The second part of the call, which starts after the failover, is recorded to a file on the alternate media server.

Reliability failover for voice call recording applies to both regular and emergency call recording. For more information about the call recording feature, see "Call Recording—NETANN-Based" on page 121.

Media Server Reliability—NETANN

In the case of NETANN, SIP Server is responsible for monitoring the Media Server. When SIP Server detects that a particular Media Server is out of service after a service is started, SIP Server is able to continue the service on an alternate available Media Server device.

However, Genesys recommends that you:

1. Install multiple Media Servers and configure them in such a way that an alternate Media Server of each type (for example, `moh`, `treatment`, `mcu`, `recorder`) is available in case of failure of the original server.
2. Enable the Active Out-of-Service Detection feature, which allows SIP Server to detect a failure in the media service device before attempting to engage it in a service, or replace the failed media service in the middle of a call.

Configuring Active Out-of-Service Detection

To configure the length of time between `OPTIONS` requests, use the options `oos-check` and `oos-force` options (configured on the `Options` tab of the `Voice over IP Service` DN). For more information about configuring this feature, see "Active Out-of-Service Detection" on page 240.

Media Server Reliability—MSML

SIP Server supports the SIP `SUBSCRITBE/NOTIFY` method for providing reliable MSML-based media services through Genesys Media Server. By subscribing to the DN used to represent Genesys Media Server for MSML-based services, SIP Server is able to determine when a particular instance of Media Server becomes unavailable, and if so, SIP Server can disconnect from the failed media server, then reconnect to an available instance of Media Server, providing continuity for any ongoing msml-based media services.

How It Works

With subscription enabled, SIP Server typically handles the failure of a Media Server instance as follows:

1. SIP Server sends periodic `SUBSCRIBE` messages to GVP Resource Manager for each `Voice over IP Service` DN (`service-type` is set to `msml`) that SIP Server is subscribed to.
2. If a particular instance of Media Server becomes unavailable, Resource Manager sends a `NOTIFY` message to SIP Server, which lists the status of each media server monitored by Resource Manager. A sample body for the `NOTIFY` message is as follows:
   `*msml/<mediaserver-ipaddress>:<mediaserver-port>/out-of-service*`
3. SIP Server disconnects all ongoing media services from this Media Server instance, then reconnects with an active Media Server through Resource Manager.
4. All existing treatments and conference connections are re-established on the alternate Media Server instance.

Feature Configuration

This feature requires no special configuration. The feature is enabled through the `subscription-id` option, which is part of the basic configuration of the `Voice over IP Service` DN. For more information, see Table 14: Integrating Media Server for MSML, on page 94.

Feature Limitation

SIP Server supports reliability for media services after the initial failure of a Media Server only. For any subsequent media server failure, SIP Server is unable to restart the service using another Media Server.

Modifying the From Header in SIP INVITE

Introduced in SIP Server 8.1.102.05

SIP Server provides the ability to modify the `From` header in outgoing SIP `INVITE` messages. Use the following configuration options to enable this functionality, depending on your needs. These `cpn`-controlling options are configured on an `Extension` DN or `Voice over IP Service` DN with `service-type=softswitch`, in the `TServer` section:

- `cpn-self`
- `cpn-dnis`
- `cpn-digits-to-both-legs`

Note: Genesys does not recommend using the `cpn` configuration option and the options described above together on the same device.

Multi-Threaded Logging

SIP Server is a multi-threaded application where the number of threads is controlled by the sip-link-type configuration option. Each SIP Server thread can produce a dedicated log file. If the standard T-Server log configuration is applied, then only the main thread log is generated. Logs from other threads can be enabled by applying configuration explained in this section.

How It Works

To enable each SIP Server thread to produce a dedicated log file, specify a file name for the x-sip-log option. SIP Server creates an extra file for each running thread, as specified in the sip-link-type option. Table 68 provides the recommended x-sip-log settings and the corresponding SIP Server behavior for different sip-link-type configurations.

Table 68: Multi-threaded Log Files

sip-link-type	Log File
0	If the x-sip-log option is set to an empty value, a single log file for all SIP Server activities (including T-Library and SIP messages) is created according to common log file configuration. If the x-sip-log option does not exist in the configuration, SIP Server generates Transport (768) and Operational Information (1536) log files in addition to the main thread log.
3	If the x-sip-log option is set to an empty value, SIP Server generates one log file, which contains only the T-Library–related activity. If the x-sip-log option does not exist in the configuration, SIP Server creates several log files, generated by running threads, including: • T-Server thread (T-Library messages) • Call Manager thread (SIP messages) • Transport thread • Service Checker thread • Operational Information thread The T-Server thread is defined by common logging parameters. The other threads are defined by the x-sip-log option.

Table 68: Multi-threaded Log Files (Continued)

sip-link-type	Log File
4	If the `x-sip-log` option is set to an empty value, SIP Server generates one log file, which contains only the T-Library–related activity. If the `x-sip-log` option does not exist in the configuration, SIP Server creates several log files, generated by running threads, including: • T-Server thread • 16 Call Manager threads • Transport thread • Service Checker thread • Presence Manager thread • Operational Information thread The T-Server thread is defined by common logging parameters. The other threads are defined by the `x-sip-log` option.

How the Filenames are Created

In multi-threaded logging, all log filenames, except for the T-Server thread log, are created using the following format:

`LogName-ThreadId.DateTime.log`

- `LogName` is defined in the `x-sip-log` parameter.
- `ThreadID` identifies the particular thread:
  - 001-016 is used for any of the Call Manager threads
  - 256 is used for the Presence Manager thread
  - 512 is used for the Service Checker thread
  - 768 is used for the Transport thread
  - 1536 is used for the Operational Information thread
- `DateTime` is created using the format `yyymmdd-hhmmss_iii`, where `iii` shows milliseconds.

Log Expiration

Automatic log deletion upon expiration is controlled by the `expire` option. One of the ways to control log expiration is to set a maximum number of files to store. In this case, the limit is applied to each thread separately. For example, if `sip-link-type` is set to `3`, multi-threaded logging is enabled and the limit is set to `30`, SIP Server keeps approximately 150 files, at least; the rest are considered expired. The five threads (T-Server, Call Manager, Transport, Service Checker, and Operational Information), each contribute 30 files toward the total.

Logging To Remote Location

SIP Server supports storing log files at the remote network location. This logging mode introduces a number of limitations and, as a result, is not recommended. If logs are stored on the network drive, then a `.snapshot.log` file must be disabled (see `no-memory-mapping`). This file is a key for troubleshooting the problems when SIP Server exits unexpectedly. In addition, logging to a network drive may cause SIP Server to stop responding while access to the log file is not available or delayed due to network issues.

Genesys recommends configuring the `expire` option and storing all SIP Server logs locally, and during a period of low activity periodically move the logs from the SIP Server host to network storage. This widely used approach helps to keep the size of the local hard disk drive small and at the same time to keep all required information for troubleshooting of potential issues.

Feature Configuration

Table 69 describes how to configure multi-threaded logging.

Table 69: Configuring Multi-Threaded Logging

Objective	Related Procedures and Actions
Enable multi-threaded logging.	In the SIP Server `Application` object > `Application Options` tab > `Log` section, configure the following option: • `x-sip-log`—Enter a path and filename where the additional SIP-based log files will be created.

Sample Configuration

The following is an example of how the different configuration option settings result in a particular set of log files:

- `x-sip-log` option is set to `c:\logs\extralog`
- `sip-link-type` is set to `3`

In this case, in addition to the main T-Sever log (defined by the common options found in the `log Section`), several additional log files are created, including:

- `extralog-001.20101109_093122_439.log` - Call Manager
- `extralog-768.20101109_093122_376.log` - Transport
- `extralog-1536.20101109_093122_864.log` - Operational Information

Music and Announcements

SIP Server is able to control the playing of announcements by using Genesys Media Server. Media Server provides an announcement service that plays various types of prompts, such as music or recorded files, and also provide a service that plays prompts and collects DTMF tones inputted by the caller. For information about how to configure Genesys Media Server, refer to "Configuring Genesys Media Server" on page 92.

This section describes the following topics:

Announcement Treatments on Routing Points

When creating a routing strategy in Interaction Routing Designer for announcement treatments `PlayAnnouncement` and `PlayAnnouncementAndDigits`, include the parameters listed in Table 70.

Table 70: Announcement Treatment Parameters

Parameter	Description
`LANGUAGE`	Ignored.
`MSGID`	Ignored.
`MSGTXT`	Ignored.
`PROMPT`	Contains up to 10 subprompts. Each contains a music file, and these are played in order.
`INTERRUPTABLE`	When this check box is selected, the caller can interrupt the announcement with a `DTMF` keystroke. **Note**: See "Feature Limitation" on page 286.
`ID`	Contains an integer, that refers to the media file located in the `announcement` subdirectory of the installed MCP root directory. For example, the value `1` refers to the file `announcement/1_alaw.wav`, if the G.711 A-law codec is used.
`DIGITS`	Ignored.
`USER_ID`	Supported by Media Server using the `users/<customer id>_<ann.id>` file.

Table 70: Announcement Treatment Parameters (Continued)

Parameter	Description
USER_ANN_ID	Supported by Media Server using the users/<customer id>_<ann.id> file.
TEXT	Starting with version 8.1.101.15, SIP Server supports the TEXT parameter in TApplyTreatment with the following applicable treatments: • PlayAnnouncement • PlayAnnouncementAndDigits • RecordUserAnnouncement This parameter is used to specify the URL/path of the file to be played. It is sent as an item in the PROMPT list. URL format: protocol://FQDN-or-IPAddress:port/path/filename For example: http://localhost/rMessages/dk/ATP_VMIntro.wav Or, Absolute file path: file://C:/announcement/test/WelcomeGreeting.wav;is-absolute-path=true Relative file path: • file://announcement/test/WelcomeGreeting.wav • file://announcement/test/WelcomeGreeting.wav;is-absolute-path=false When the is-absolute-path parameter is set to true, SIP Server does not remove the file:// string when passing the file path to MCP, and MCP can access the absolute file path. When this parameter is not set or set to false, SIP Server removes the file:// string from the file path.

If the treatment is terminated early because of a problem with Media Server or SIP Server, SIP Server sets the Extension data fields ERR_CODE and ERR_TEXT. To determine whether these fields and their values exist, from a routing strategy, use the function ExtensionData. Place this function on a normal completion branch (not the error branch) after the treatment.

Refer to the *Universal Routing 8.1 Reference Manual* for more information on the use and configuration of strategies.

Note: When creating a new strategy in Interaction Routing Designer, leave the Wait For Treatment End check box selected, to allow the treatment to play until completion.

Music Treatments on Routing Points

When creating a strategy in Interaction Routing Designer for music treatments, specify the parameters described in Table 71.

Table 71: Music Treatment Parameters

Parameter	Description
MUSIC_DN	Specifies the music source that Media Server plays. The format is: <directory>/<music file name> Example: music/in_queue Where music is a subdirectory in the MCP root directory and in_queue is a base file name. Refer to the *Genesys Media Server Deployment Guide* for supported media file types and archives. The default-music option is used if the value of the MUSIC_DN parameter is not specified.
DURATION	Specifies the duration of the music (in seconds). **Note:** This parameter is ignored if MUSIC_DN is blank. To continue playing music after the treatment terminates, consider creating one of the following strategies using Interaction Routing Designer: • Execute the treatment inside a route-selection treatment block. In this case, the treatment continues until a route target is selected. • Follow the treatment with the SuspendForTreatmentEnd function. In this case, the treatment plays music until terminated after the delay specified in option DURATION. • Follow the treatment with the delay function. In this case, the treatment plays music for the period specified in option delay. If DURATION is less than delay, silence is played for the time difference.

Refer to the *Universal Routing 8.1 Reference Manual* for more information on the use and configuration of strategies.

Other Treatments on Routing Points

The treatments in Table 72 continuously loop a pre-defined audio file to a call. The treatment types are as follows:

Table 72: Other Treatments on Routing Points

Treatment Type	Description
Busy	Plays a busy tone. To define the busy tone audio file, configure the SIP Server `busy-tone` option.
Fast Busy	Plays a fast busy tone. To define the fast busy tone audio file, configure the SIP Server `fast-busy-tone` option.
Silence	Plays no sound. To define the `silence` audio file, configure the SIP Server `silence-tone` option.
Ringback	Plays ringback tone. To define the `ringback` audio file, configure the SIP Server `ring-tone` option.
CollectDigits	Collects customer-entered digits.
RecordUser-Announcement	Records a user's announcement and saves into a `users` folder.

Refer to the *Universal Routing 8.1 Reference Manual* for more information about the use and configuration of strategies.

Feature Limitation

When digits collection is completed (`MAX_DIGITS` limit is reached or `ABORT/TERM_DIGITS` is entered), the treatment `PlayAnnouncementAndCollectDigits` ends, causing the interruption of announcement regardless of the `INTERRUPTABLE` flag set for this announcement.

Nailed-Up Connections for Agents

SIP Server supports a persistent "nailed-up connection" for agents, where it maintains an extended telephone call between SIP Server and the agent. During this time, the agent can handle multiple customer interactions without dropping the telephone connection to SIP Server.

Nailed-up connections offer a few key benefits, including:

- Minimal delay between the time an agent is selected and the audio path to the customer is established
- Improved overall reliability—the connection is already established when delivering a customer "call", and the agent is less likely to take non-contact center calls

One typical use of nailed-up connections is for agents who use a legacy PSTN phone. These agents could be working from their homes, or in a branch office that has simple PSTN connectivity. Another typical use of nailed-up connections is for agents behind a third-party PBX, when the PBX is connected to SIP Server through a gateway or simple SIP trunk.

SIP Server supports virtually all agent functionality in conjunction with nailed-up connections. The agent can make calls, receive calls, transfer calls, consult with other agents, use call supervision, and more. In addition, SIP Server's call recording functionality is fully compatible with nailed-up connections.

Inbound calls to an agent with a nailed-up connection are delivered by default with "auto answer"—meaning the audio connects immediately. If this "auto answer" experience is not desired, then Preview Interactions should be used to provide the agent the opportunity to see call information in their agent desktop and accept or reject the call.

Nailed-up connections can be established or disconnected either by SIP Server or by the agent.

Note: In Business Continuity deployments, any DN with a statically configured contact must use `dr-forward` set to `no-agent`. In practical terms, such a DN is commonly used for a "remote agent", often in conjunction with the nailed-up connection. See the *SIP Server 8.1 High-Availability Deployment Guide* for details.

Establishing the Nailed-Up Connection

Nailed-up connections can be established by three different methods:

- SIP Server establishes the nailed-up connection on agent login or when an agent is in Ready state.
- SIP Server establishes the nailed-up connection on the first customer call.
- An agent establishes the nailed-up connection by calling into a contact center Route Point.

SIP Server Establishes the Nailed-up Connection on Agent Login or Ready state

SIP Server can establish the persistent nailed-up connection with an agent when the agent logs in, depending on the configuration:

- When `connect-nailedup-on-login=<Routing Point number>`, SIP Server connects the agent's endpoint with the specified Routing Point and then, after processing the `TRouteCall` to the predefined `gcti::park` device, SIP Server parks the agent on the `gcti::park` device establishing the persistent nailed-up connection with the agent's endpoint.
- When `connect-nailedup-on-login=gcti::park`, SIP Server directly parks the agent on the `gcti::park` device, establishing the persistent nailed-up connection with the agent's endpoint while processing `TAgentLogin`.

If a nailed-up connection is terminated for any reason, SIP Server places the agent in the `NotReady` state. If an agent is in the `NotReady` state and a nailed-up connection is not yet established, SIP Server, while processing the `TAgentSetReady` request, initiates a SIP call to the agent's phone with further parking on the `gcti::park` device. If the call fails, SIP generates `EventError` in response to `TAgentSetReady`; the agent remains in the `NotReady` state.

If the agent logs out, the nailed-up connection is dropped.

SIP Server Establishes the Nailed-up Connection on First Customer Call

SIP Server calls the agent to start a session—SIP Server sends the call to an agent DN configured for the nailed-up feature. This applies to the first transfer to the agent, where the initial nailed-up session starts. When the caller releases the call or the agent releases the call using 3pcc, SIP Server parks the agent on Media Server, keeping the connection for the call leg to the nailed-up connection.

The basic call flow when SIP Server first calls an agent configured for the nailed-up feature is as follows:

1. SIP Server receives a customer call, which the Universal Routing Server then processes.
2. After qualification and queuing, the routing strategy selects the agent who will handle the call.
3. SIP Server contacts the agent as it would for any remote TDM extension (SIP Server does not yet consider the agent to be nailed-up).
4. At the end of the call, when the agent requests to release the call through the Agent Desktop (a 3pcc `TReleaseCall`), SIP Server does not disconnect the call leg to the nailed-up connection but, instead, parks the agent on the predefined `gcti::park` device. At this point, the agent is considered to be nailed-up. Media Server plays a silent treatment while the nailed-up connection is maintained.

In Business Continuity deployments, SIP Server applies the following "Call Delivery" logic when establishing the initial call to a DN with a statically configured contact:

1. If the first SIP Server to handle the call determines an agent is locally logged in and using the DN, this SIP Server delivers the call directly to the DN.
2. Otherwise, the first SIP Server forwards the call to the second SIP Server on the alternate site, using the inter-site Trunk DN and ISCC. The second SIP Server delivers the call to the DN, regardless of whether any agent is logged in and using the DN or not.

Note: Carefully consider this behavior. This could result in high telephone connection charges, if, for example, DNs and data centers are distributed across different countries.

Agent Establishes the Nailed-Up Connection by Calling into a Contact Center Route Point

The agent calls the contact center to start a session—The agent DN (configured for the nailed-up feature) initiates a call (1pcc) to the contact center.

The basic call flow when an agent DN is configured for the nailed-up feature is as follows:

1. A call from the remote agent arrives at the contact center on a Routing Point DN.
2. A short treatment is applied, and URS issues a `TRouteCall` to the predefined `gcti::park` device (`RouteType=Unknown`; `OtherDN='gcti::park'`).
3. SIP Server parks the agent on the `gcti::park` device, keeping the call leg to the agent connected. At this point, the agent is considered to be nailed-up. Media Server applies a silent treatment while the nailed-up connection is maintained.

In Business Continuity deployments, each data center should have a unique routing point, which allows an agent to connect to their preferred data center based on which routing point they contact.

Reporting on Nailed-Up Connection Calls

Calls that involve the nailed-up connection use T-Library events that are identical to those used for a regular agent. Each customer interaction established with an agent in a nailed-up connection is reported as a distinct Genesys call, allowing monitoring and reporting on each individual call.

Disconnecting the Nailed-Up Connection

Nailed-up connections can be disconnected for several reasons:

- The agent hangs up the phone.
- A network problem between SIP Server and the phone causes the call to be dropped.
- The agent logs out (applies when SIP Server established the connection on login, or if the `drop-nailedup-on-logout` option is set to `true`).
- The agent is inactive (no changes in agent state or incoming/outgoing calls at the DN) for the specified period of time (`disconnect-nailedup-timeout`).

Note: If you enable `drop-nailedup-on-logout`, SIP Server can only establish the nailed-up connection when the remote agent is logged in.

Feature Configuration

Table 73 describes how to configure the nailed-up connection for an agent DN.

Table 73: Configuring a Nailed-Up Connection

Objective	Related Procedures and Actions
1. Configure the agent DN.	In the `ACD Position` or `Extension` DN for the agent, in the `TServer` section, configure the following options: • `contact`—Set this option to the contact URI of the PSTN gateway/SBC or third-party PBX, depending on the agent location. • `line-type`—Set this option to `1`. • `refer-enabled`—Set this option to `false`. • `dual-dialog-enabled`—Set this to `false`. • `reject-call-notready`—Set this option to `true` (recommended, not mandatory). • `sip-cti-control`—Ensure that this option is not configured. For general information about creating agent DNs, go to: • Table 5, “Configuring Endpoints,” on page 83. For an agent DN behind the softswitch, set the options above on the softswitch object representing the DN, while the DN itself will not have any options set.

Table 73: Configuring a Nailed-Up Connection (Continued)

Objective	Related Procedures and Actions
2. Configure the SIP Server Application and/or DN objects.	**Connection on Agent Login** To enable the persistent nailed-up connection on agent login, use any of the following configurations, listed in order of priority: 1. Extensions Attribute: `connect-nailedup-on-login` key 2. In the `TServer` section of the individual DN, configure the `connect-nailedup-on-login` option. 3. In the `TServer` section of the SIP Server Application, configure the `connect-nailedup-on-login` option. **Note:** If the agent logs out, the nailed-up connection will be dropped; the same behavior as if the `drop-nailedup-on-logout` is set to `true`. **Disconnection on Inactivity** To terminate the agent's nailed-up connection because of agent's inactivity, in the `TServer` section, configure the following option: • `disconnect-nailedup-timeout` (can be set at both Application or DN levels) **Disconnection on Agent Logout** To enable automatic disconnection of the agent from the nailed-up connection on agent logout, in the `TServer` section, configure the following option: • `drop-nailedup-on-logout`—Set this to `true`. **Note**: If enabled, SIP Server can only establish the nailed-up connection if the agent is logged in.
3. (Optional) Configure Business Continuity.	For Business Continuity deployments, set `dr-forward` to `no-agent`. See the *SIP Server 8.1 High-Availability Deployment Guide*.

Feature Limitations

- Consultation calls for nailed-up DNs are supported in single dialog mode only.
- If an agent with the nailed-up connection is participating in the first call before it was ever parked, SIP Server cannot park this agent if the call is released before it is established. For example, if the agent with the nailed-up connection initiates a call and releases it while the call is ringing, or if the agent with the nailed-up connection completes a two-step transfer in ringing state. To avoid this, the agent must call the call center to get parked first.

Network Asserted Identity

SIP Server supports the Network Asserted Identity mechanism for controlling the presentation of personal information (caller details) in SIP messages within a trusted network. This mechanism uses special private headers to prevent or allow user information—the SIP URI—from being shared across network nodes, depending on whether a particular node is trusted or untrusted. It allows SIP Server to comply with caller requests to keep their identity private. For example, Calling Line Identification Restriction (CLIR).

How the Mechanism Works

If privacy is required, the special header `privacy:id` is included in the `INVITE`, and all user information in the `From` part of the `URI` is replaced with anonymous content. For example:

```
From = "Anonymous" <sip:anonymous@anonymous.invalid>;tag=1928301774
```

If the `INVITE` is sent between trusted nodes, it will also include the header `P-Asserted-Identity`, which provides the identity of the caller (a URI and an optional display name) in a controllable header. For example:

```
P-Asserted-Identity: "Bob" <sip:7000@company.com>
```

If, in the incoming `INVITE`, there is a `P-Asserted-Identity` header and no `Privacy` headers, in both trusted and non-trusted modes, the provided `P-Asserted-Identity` header will be supplied in the resulting `INVITE` to the destination.

For example, an `INVITE` between trusted or untrusted entities might look like this:

```
INVITE sip:UserB@internalresource:5067 SIP/2.0
From: "Anonymous" <sip:anonymouse@anonymous.invalid>;tag=1928301774
To: <sip:UserB@internalresource:5068
P-Asserted-Identity: "Bob" <sip:UserA@externaldevice.com>
Privacy: id..
```

Note: For complete information about either the privacy mechanism or the network asserted identity extensions, refer to the following RFCs:

- "RFC 3323: A Privacy Mechanism for the Session Initiation Protocol (SIP)"
- "RFC 3325: Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks"

How SIP Server Supports the Mechanism

SIP Server supports control of the Network Asserted Identity mechanism using the following DN-level options:

- `privacy`
- `p-asserted-identity`
- `enforce-trusted`
- `enforce-privacy`
- `enforce-p-asserted-identity`

On receiving an `INVITE` request that includes the `P-Asserted-Identity` header, SIP Server will consider this header as the ANI for the call.

For backwards compatibility for IMS integrations, where all devices inside the IMS deployment are trusted by default, the enforce-trusted option is also supported on the Application level.

Inserting the P-Asserted-Identity Header

SIP Server will make the `From` header anonymous as well as insert the `privacy` and `P-Asserted-Identity` headers in the following cases:

- Incoming `INVITE` request includes the `p-preferred-identity` and `Privacy:id` headers.
- Origination DN is configured with the `privacy` option set to `id`.
- Destination DN or outbound `Trunk` DN is configured with the `enforce-p-asserted-identity` option.

Generating the P-Asserted-Identity Header

SIP Server gets the information it needs to generate the `P-Asserted-Identity` header, if required, from the following sources listed in order of priority:

1. Value of the `enforce-p-asserted-identity` option configured on the destination DN or outbound `Trunk` DN.
2. Value of the `p-asserted-identity` option configured on the origination DN or inbound `Trunk` DN.
3. Content in the `P-Asserted-Identity` header included in the incoming `INVITE` from a trusted origination.
4. Content in the `P-Preferred-Identity` header included in the incoming `INVITE` from a trusted origination.
5. Content in the `From` header of the incoming `INVITE`.

Removing the P-Asserted-Identity Header

SIP Server will remove the `P-Asserted-Identity` header from an outbound request if the `Trunk` or `Voice over IP Service` DN is configured with the `enforce-trusted` option set to `false`.

Inbound Calls

For inbound calls arriving at SIP Server, the incoming `INVITE` may already include privacy restrictions—if, for example, CLIR was enabled externally by the caller. If SIP Server receives such an `INVITE`, it will forward it to the destination with privacy and, if both the inbound `Trunk` DN and the destination DN are trusted, with the `P-Asserted-Identity` header as well.

Inbound Call From Trusted Entity to Trusted Entity (Caller Requests Privacy)

Figure 24 shows a sample call flow for an inbound call between a trusted external domain and a trusted destination DN on SIP Server.

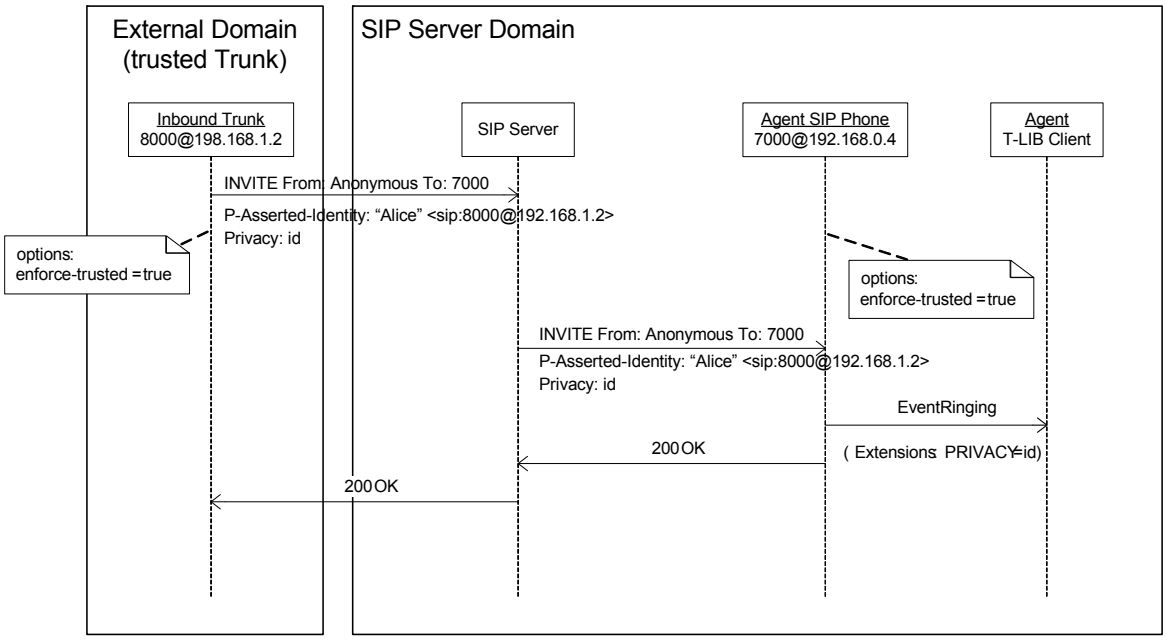

Figure 24: Trusted External Call To Trusted Destination DN (with privacy restriction)

In this case, both the inbound `Trunk` DN and the destination DN are configured with `enforce-trusted` set to `true`. SIP Server forwards the `INVITE` with all privacy elements included: `From` is anonymous and both `privacy: id` and `P-Asserted-Identity` headers are included.

Inbound Call From Trusted Entity to Non-Trusted Entity (With Privacy)

In this case, the inbound `Trunk` DN is configured as trusted, but the destination DN is not trusted. When SIP Server receives a privacy-enabled `INVITE`, it forwards only the anonymous `From` and the `privacy:id` header to the destination. It removes the `P-Asserted-Identity` header, so no personal information is presented to the destination device.

Inbound Call From Non-Trusted Entity (With Privacy)

In this case, the inbound `Trunk` DN is configured as non-trusted. It does not matter if the destination DN is trusted or not, SIP Server removes the `P-Asserted-Identity` header when it forwards the restricted `INVITE` to the destination.

Inbound Call From Trusted Entity (COLR)

In this case, the `Trunk` DN is configured as trusted and the destination DN is configured with Connected Line Identity Restriction (COLR) which restricts presentation of called line identity to the caller. To enable COLR on the destination DN, set the `privacy` option to `id`. When SIP Server sends the `200 OK` back to the caller it includes the `Privacy: id` header and the `P-Asserted-Identity` header, obtained from the `p-asserted-identity` option configured on the on destination device.

Inbound Call From Trusted Entity (COLP)

In this case, the `Trunk` DN is configured as trusted, and the destination DN is configured with Connected Line Identity Presentation (COLP), which allows presentation of called line identity to the caller. To enable COLP on the destination DN, set the `p-asserted-identity` option on the destination DN. When SIP Server sends the `200 Ok` back to the caller, it includes the `P-Asserted-Identity` header.

Inbound Call From Any Entity (CLIP)

In this case, SIP Server passes through the `P-Asserted-Identity` header as the identity of the origination DN, regardless of `enforce-trusted` option.

Inbound Calls to GVP

For inbound calls to GVP, the following functionality applies:

- When GVP receives the `P-Asserted-Identity` header in an incoming `INVITE`, it provides the value of the header as the ANI to the VoiceXML application.
- When GVP receives both the `P-Asserted-Identity` and `privacy` headers in an incoming `INVITE`, it provides the value of both headers to the VoiceXML application.

- For call transfers using the `<transfer>` tag, the VoiceXML application can set the `P-Asserted-Identity` and privacy headers to be used in the `INVITE` from GVP to SIP Server.
- If the VoiceXML application does not set the `P-Asserted-Identity` and `privacy` headers for the call transfer, GVP will propagate the header values from the inbound call leg to the outbound call leg during the transfer.

Outbound Calls

When generating an outbound `INVITE` request, SIP Server will establish privacy if the `Extension` DN that initiates the call is configured with the `privacy` option set to `id` and the `p-asserted-identity` option set to the correct URI and `enforce-trusted` is set to `true`. In this case, SIP Server will replace the `From` with an anonymous URI and add both the `privacy` and `P-Asserted-Identity` headers to the `INVITE`.

Outbound Call to Non-Trusted Entity (Restricted Identity)

Figure 25 shows a sample call flow for an outbound call from a trusted internal DN to an untrusted external destination.

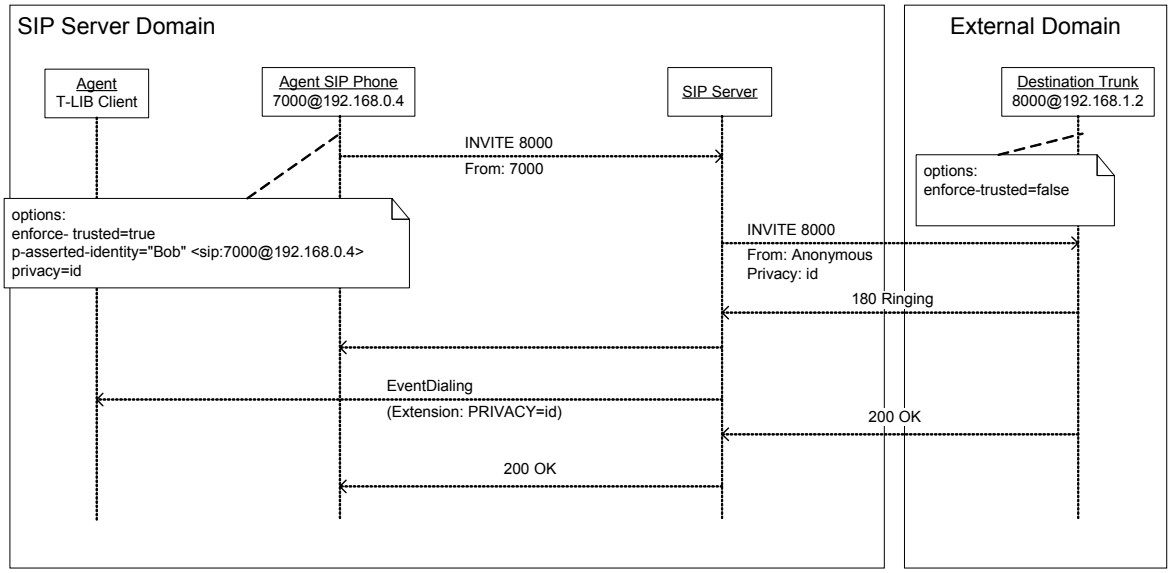

Figure 25: Outbound Call to Non-Trusted Destination (With Privacy Restriction)

In this case, the `Extension` DN 7000 is configured as both requesting privacy and as a trusted entity. On receiving the `INVITE` from DN 7000, SIP Server translates the `From` to `Anonymous` and adds the `privacy:id` header. However, because the `INVITE` is addressed to the non-trusted gateway `Trunk` DN 8000 (`enforce-trusted` option set to `false`), SIP Server does not include the `P-Asserted-Identity` header—no personal information is available for the rest of the call.

Outbound Call to Trusted Entity (Restricted Identity)

In this case, the outbound `Trunk` DN represents a trusted external domain so it is configured with `enforce-trusted` option set to `true`. The internal DN placing the call is configured for privacy. On receiving the outgoing `INVITE` from the internal DN, SIP Server translates the `From` to `Anonymous` and adds both the `privacy:id` header as well as the user identifying `P-Asserted-Identity` header.

Outbound Call to Any Entity (CLIP)

In this case, the internal DN placing the call is configured for presentation using only the `p-asserted-identify` option. On receiving the outgoing `INVITE` from the internal DN, SIP Server passes through the user identifying `P-Asserted-Identity` header, regardless of `enforce-trusted` option.

Internal Call (CLIR)

In this scenario, an internal DN sends a CLIR call to another DN on the same domain. For example, if the caller (DN1) is configured for privacy, and the called party (DN2) is configured as non-trusted, SIP Server sends an anonymous `INVITE` without the `P-Asserted-Identity` header:

DN1@SIPServer1 → DN2@SIPServer1

```
INVITE sip:DN2@SIPServer1:5060 SIP/2.0
From:"Anonymous" <sip:anonymous@anonymous.invalid>;tag=1928301774
To: <sip:DN2@SIPServer1:5060>
Privacy: id
```

Figure 26 shows a sample call flow for an internal call from a trusted Extension DN to an untrusted Extension DN within the SIP Server domain.

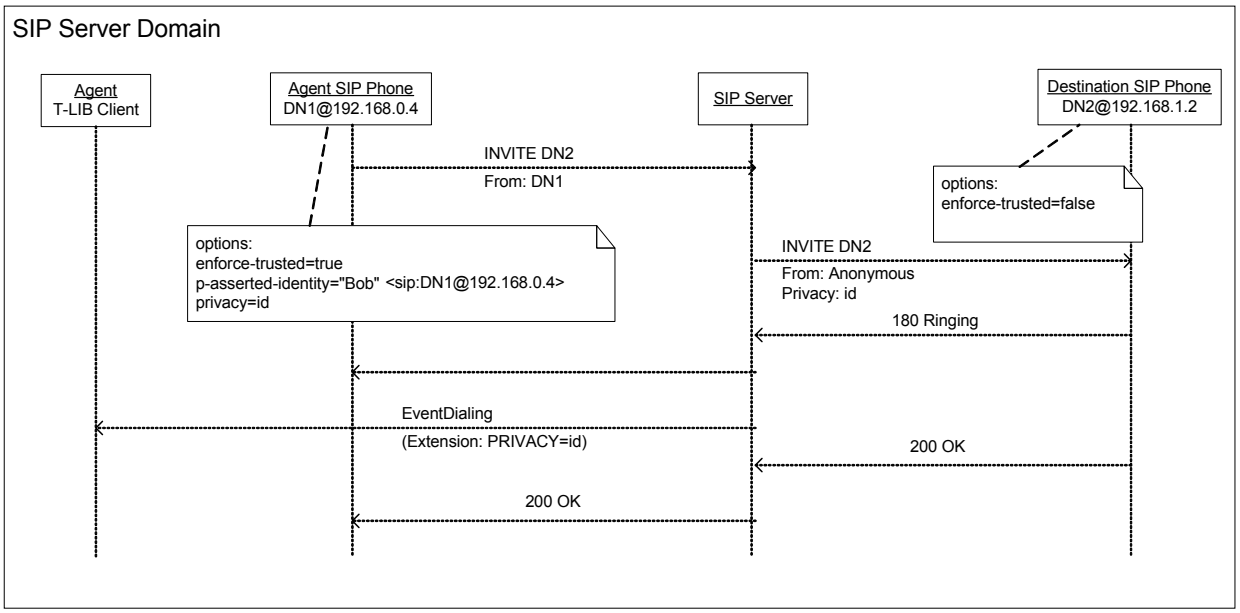

Figure 26: Internal Call From Trusted Entity to Non-Trusted Entity

In this case, the calling entity (DN1) is configured to request privacy. The destination (DN20 is configures as non-trusted. When SIP Server receives the `INVITE` to place the call from DN1 to DN2, it adds the `privacy:id` header, anonomizes the `From` header, but does not insert `P-Asserted-Identity` to the outgoing `INVITE` that is forwards to DN2.

Feature Configuration

Table 74 describes how to control privacy for SIP messaging within the network.

Table 74: Configuring Network Asserted Identity

Objective	Key Procedures and Actions
Request privacy for outbound calls.	In the `Extension` DN that places the call, in the `TServer` section, configure the following options: • `privacy`—Set to `id`. • `p-asserted-identity`—Set to correct URI. • `enforce-trusted`—Set to `true`.
Enable privacy presentation on the inbound Trunk.	**If the INVITE includes privacy...** In the `Trunk` DN that represents the external device sending the inbound `INVITE`, in the `TServer` section, configure the following option: • `enforce-trusted`—Set to `true`. **If the INVITE does not include privacy...** In the `Trunk` DN for the external device, in the `TServer` section, configure the following options: • `enforce-trusted`—Set to `true`. • `privacy`—Set to `id`. • `p-asserted-identity`—Set this to the identity you want to include in the `P-Asserted-Identity` header. If this option is not configured, then SIP Server gets the content from the `From` header instead.
Disable privacy presentation on the inbound Trunk.	In the `Trunk` DN that represents the external device, in the `TServer` section, configure the following option: • `enforce-trusted`—Set to `false`.
Enable privacy presentation on the DN.	For any DN you would like to allow presentation services, in the `TServer` section, configure the following option: • `enforce-trusted`—Set to `true`.

Table 74: Configuring Network Asserted Identity (Continued)

Objective	Key Procedures and Actions
Enable privacy presentation on the destination DN or outbound `Trunk` DN.	For the destination DN or outbound Trunk DN, in the `TServer` section, configure the following options: • `enforce-privacy`—Set to `id`. • `enforce-p-asserted-identity`—Set to the correct URI.
Request presentation for outbound calls.	In the `Extension` DN that places the call, in the `TServer` section, configure the following option: • `p-asserted-identity`—Set to the correct URI.

Feature Limitations

The Network Asserted Identity mechanism is not supported on any scenarios involving Routing Points except predictive calls.

Network Attended Transfer

SIP Server supports Network Attended Transfers (NAT)—the ability of an agent on one SIP Server to consult another agent in a multi-site environment before transferring a call. To prevent signaling loops for these types of transfers, SIP Server supports the following transfer and reconnect operations, used to communicate between SIP Server instances:

- `TNetworkConsult`
- `TNetworkAlternate`
- `TNetworkTransfer`
- `TNetworkReconnect`

For NAT, SIP Server supports both implicit and explicit transfers, including premature disconnection and blind transfers, as well as reconnect operations.

Network T-Server Replacement

NAT enables SIP Server to take over some of the functionality that was previously provided by Network T-Server—the key differences being that SIP Server does not support `NetworkMerge` and performs URS-controlled consultations through a separate call.

How It Works

Figure 27 provides an overview of the NAT process in a pure SIP environment, with one SIP Server with network capability, and two premise SIP Server instances that are used to serve agents.

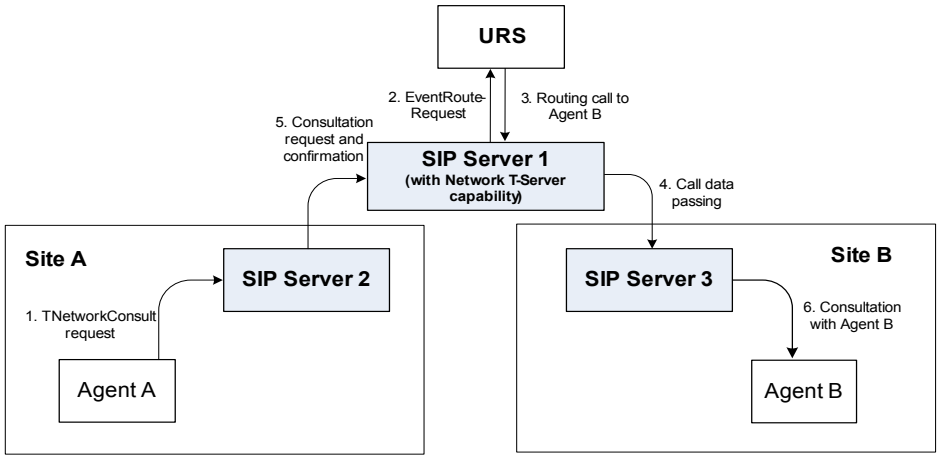

Figure 27: Steps in Network Attended Transfer in URS-Controlled Mode

Step 1

Agent A makes a request for a consultation with another agent. A `TNetworkConsult` request is relayed to SIP Server. Depending on the parameter settings of the `TNetworkConsult` request, the NAT feature will operate in either *direct* or *URS-controlled* mode. For more information, see the *Platform SDK 8.x .NET (or Java) API Reference.*

Step 2

(URS-controlled mode only.) SIP Server 1 creates the call (with new `AttributeConnID`) and sends `EventRouteRequest` to URS.

Step 3

(URS-controlled mode only.) URS locates an available agent at Site B and instructs SIP Server 1 to route the call to Agent B. SIP Server1confirms the initiation of the network transfer by sending `EventNetworkCallStatus` to SIP Server 2, which then relays it to Agent A.

Step 4

SIP Server 1 proceeds to obtain the access number from SIP Server 3, and passes the call data to SIP Server 3.

Step 5

SIP Server 1 makes the call to Agent B on SIP Server 3. Once the connection is established, SIP Server 1 distributes `EventNetworkCallStatus` to both SIP Server 2 and SIP Sever 3, which then relays it to Agent A and Agent B (in direct mode only) respectively, to indicate that the consultation call is being established.

SIP Server 1 also distributes `EventRouteUsed` to URS to confirm successful routing of the call to the selected agent (URS-controlled mode only).

Step 6

At this point, the customer is on hold, and Agent A is consulting with Agent B. Agent A can do one of the following:

- End the consultation and retrieve the original customer call.
- Alternate between Agent B and the customer.
- Transfer the customer call to Agent B.

Notes:
- SIP Server supports NAT requests with `AttributeHomeLocation` provided that this attribute identifies a network location that is capable of processing such requests.
- For network-capable SIP Server, URS-controlled mode is only supported when `RequestNetworkConsult` includes an `AttributeOtherDN` that is set to the CDN of the network-capable SIP Server, and does not include an `AttributeLocation`.

Feature Configuration

Table 75 describes how to enable Network Attended Transfers in multi-site environments that include SIP Server.

Table 75: Configuring Network Attended Transfer

Objective	Key Procedures and Actions
Configure multi-site connection.	Configure multi-site connection (ISCC) between SIP Server (network capable) and either premise SIP Server (agent) or TDM T-Server instances. You do not need to configure multi-site connection between premise SIP Sever or T-Server instances. For details, see "Configuring Multi-Site Support" on page 700.

Feature Limitations

- Because SIP Server creates a new call for URS-controlled consultations (with a new `AttributeConnID`), SIP Server will propagate `NetworkCallStatus` events to the consulting party only if, in the `extrouter` section, you set the option `use-data-from` to `original`.
- While processing a Network Transfer, SIP Server may send an incorrect `EventPartyChanged` message if the `use-data-from` option is set to `current`. Genesys does not recommend setting the `use-data-from` option to `current` while using the Network Attended Transfer feature.

No-Answer Supervision

This section describes SIP Server's No-Answer Supervision feature and its configuration.

Business and Private Calls

No-Answer Supervision can be applied to business and private calls.

Business Calls

SIP Server automatically categorizes any call distributed to an agent either from a Queue or from a Routing Point as a *business call.* Use the following configuration options to define what additional calls to or from an agent are classified as business calls:

- `inbound-bsns-calls`
- `outbound-bsns-calls`
- `inherit-bsns-type`
- `internal-bsns-calls`
- `unknown-bsns-calls`

Private Calls

SIP Server categorizes any call that does not fall into the business or work-related categories as a *private call*. SIP Server does not apply any automatic business-call handling after a private call. If an agent receives a direct private call while in wrap-up or legal-guard time, the emulated wrap-up or legal-guard timer is not interrupted.

You can apply No-Answer Supervision to private calls using the configuration option `nas-private`.

Agent No-Answer Supervision

This feature provides the following functionality:

- If an agent does not answer a call within a specified timeout, SIP Server can divert the call to a sequence of overflow destinations. Alternatively, you can configure SIP Server to return calls automatically to the last distribution device.
- If an agent fails to answer a call within a specified timeout, you can configure SIP Server to either log out the agent or set the agent to `NotReady` to prevent further calls from arriving.

Configuration Options

SIP Server provides the following configuration options for defining the behavior of the Agent No-Answer Supervision feature:

- `agent-no-answer-action`
- `agent-no-answer-overflow`
- `agent-no-answer-timeout`
- `nas-private`
- `set-notready-on-busy`

Reporting ReasonCode

Introduced in SIP Server 8.1.101.52

SIP Server can report `ReasonCode` set to `no-answer` in the `AttributeExtensions` or `AttributeReason` of the corresponding `EventAgentNotReady` message when an agent is placed in the `Not Ready` state after not answering a call.

To enable this functionality, set the `agent-no-answer-action` option to `notready` at the Application level or the option `no-answer-action` at the `Agent Login` level. For SIP Server to report the `ReasonCode` in `AttributeExtensions`, set the `reason-in-extension` option to `true`. For SIP Server to report the `ReasonCode` in `AttributeReason`, set the `reason-in-extension` option to `false`.

Defining After Routing Timeout Action

Introduced in SIP Server 8.1.102.01

You can define SIP Server's default action for setting the state of an agent who was not able to answer the routed call before the `after-routing-timeout` expired. Enable this feature with the `after-routing-timeout-action` configuration option, or the `AFTER_ROUTING_TIMEOUT_ACTION` key in `AttributeExtensions` of `TRouteCall`. The key extension setting takes priority.

Use the agent-no-answer-timeout option with the corresponding action specified by the agent-no-answer-action option to control direct calls to an agent.

Note: Using No-Answer Supervision when the divert-on-ringing configuration option is set to false does not require the value of no-answer timeout options to be smaller than the value of the after-routing-timeout option. The value of no-answer timeout options can be bigger than the value of the after-routing-timeout option.

Multi-site support added in SIP Server 8.1.102.38

Starting with SIP Server release 8.1.102.38, this feature is supported in multi-site deployments. If the original site is configured with the divert-on-ringing option set to false, but the routing destination resides at another site, this feature is supported only if SIP Server stays in the signalling path (oosp-transfer-enabled =false).

When configured, the after-routing-timeout action is performed at the SIP Server site of the call routing destination.

If after-routing-timeout is in progress and a caller ends the call, neither agent-no-answer-action nor no-answer-action is performed, and an agent state will not be changed.

The after-routing-timeout-action option configured at the site where the TRouteCall request is processed has higher priority than the agent-no-answer-action and no-answer-action parameters at the destination site.

Limitations:

- The after-routing-timeout action is not supported at destinations where there are no agents logged in.
- The after-routing-timeout action is not supported for Shared Call Appearance or Hunt Groups.
- In case of a switchover, the after-routing-timeout timer is restarted at the new primary SIP Server.

Extension No-Answer Supervision

The No-Answer Supervision feature includes devices of type Extension. If a call is not answered on an extension within a specified timeout, SIP Server can divert the call to a sequence of overflow destinations. Alternatively, you can configure SIP Server to return calls automatically to the last distribution device.

Configuration Options

SIP Server provides the following configuration options for defining the behavior of No-Answer Supervision with devices of type Extension:

- extn-no-answer-overflow
- extn-no-answer-timeout

Position No-Answer Supervision

The No-Answer Supervision feature includes devices of type `ACD Position`. If a call is not answered on a position within a specified timeout, SIP Server can divert the call to a sequence of overflow destinations. Alternatively, you can configure SIP Server to return calls automatically to the last distribution device.

Configuration Options

SIP Server provides two configuration options for defining the behavior of No-Answer Supervision with devices of type `ACD Position`:

- `posn-no-answer-overflow`
- `posn-no-answer-timeout`

Device-Specific Overrides

SIP Server provides three configuration options to configure device-specific overrides for individual devices. You set the values for these options in the `TServer` section of the individual device.

The options are:

- `no-answer-action`
- `no-answer-overflow`
- `no-answer-timeout`

Note: The `no-answer-action`, `no-answer-overflow`, and `no-answer-timeout` configuration options are not supported on `Voice over IP Service` DNs in which `service-type` set to `softswitch`.

Extensions Attributes for Overrides for Individual Calls

For all of the No-Answer Supervision options, you can specify the corresponding `Extensions` attribute in the `TRouteCall` request, to override the configured value for individual calls. This method allows the no-answer behavior to be determined in a routing strategy. The three extensions are:

- `NO_ANSWER_ACTION`
- `NO_ANSWER_OVERFLOW`
- `NO_ANSWER_TIMEOUT`

Feature Limitations

To use No-Answer Supervision when the `divert-on-ringing` configuration option is set to `false`, follow these configuration guidelines:

1. The value of no-answer timeout options (`extn-no-answer-timeout`, `agent-no-answer-timeout`, `posn-no-answer-timeout`, or `no-answer-timeout`) must be smaller than the value of the `after-routing-timeout` option (5 seconds of the time difference is recommended).
2. The value of no-answer overflow options (`extn-no-answer-overflow`, `agent-no-answer-overflow`, `posn-no-answer-overflow`, or `no-answer-overflow`) must not be set (they must be empty).

Some details to consider: An agent phone is released from the ringing state as soon as the `after-routing-timeout` expires. At the same time, the agent state will be changed as specified by the `no-answer-action` or `agent-no-answer-action` option. If a call is answered when the `no-answer timeout` expires, but before `after-routing-timeout` expires, the call will be established normally, and there will be no change in the agent state specified by options `no-answer-action` or `agent-no-answer-action`.

Outbound IP Solution Integration

SIP Server supports the Outbound IP Solution, an offering that combines SIP Server IP signaling with the media services of the Genesys Voice Platform (GVP) to provide IP-based outbound call campaign functionality. Using either GVP or the Outbound Contact Server (OCS) to initiate the outbound call, Outbound IP can perform media services as needed—for example, Call Progress Detection (CPD) to determine if the called party is a voice or not—and then connect the called party with a Voice XML application for the voice self-service portion of the call.

Outbound IP functionality can be provided through either of the following deployments:

- Voice Platform Solution (VPS) deployment—This configuration uses a third-party "trigger" application to manage the outbound call campaign. GVP initiates the outbound call, provides media services, and connects the called party to the voice self-service application.

  For more information, see the *Voice Platform Solution 8.1 Integration Guide*.
- Outbound Contact Server (OCS) in an Outbound-VoIP deployment—This configuration uses OCS to manage the outbound call campaign. In this case, OCS initiates the outbound call, while GVP still provides the media services and connects the called party to the voice self-service application.

  For more information, see the *Outbound Contact 8.1 Deployment Guide*.

SIP Server Features for Outbound IP Solution

Table 76 describes SIP Server features supported for the Outbound IP Solution.

Table 76: SIP Server Feature Support for Outbound IP Solution

Feature	Related Options
Device selection procedure SIP Server incorporates two parameters required by the Outbound IP Solution into the procedure that it uses to select the most appropriate device from a pool of compatible device. **Note**: The partition-id for a particular predictive outbound call is assigned based on the `partition-id` setting on the Routing Point or `Trunk Group` DN that the call is made from. For more information, see "Working with Multiple Devices" on page 386.	• `partition-id` • `cpd-capability`
Additional error codes for predictive call failure When SIP Server tries to engage the GVP Media Control Platform (MCP) to make a predictive call, and MCP returns a SIP error, SIP Server adds this error as an integer value in the new key-value pair, `MediaServerErrorCode`, that it reports to the T-Library clients in the `UserData` for the call.	• No configuration required.
Support for MSML To support Outbound IP, GVP can integrate with SIP Server using the Media Server Markup Language (MSML) protocol.	• `subscription-id` • `msml-support`
Configurable beep timer SIP Server supports a timer to control the duration of the beep tone used to notify agents participating in an outbound campaign when they are about to be connected to a customer (this option is only available if the outbound campaign is running in the Active Switching Matrix (ASM) mode). Beep tones are configured as Extensions in `TMakeCall` requests. To configure the duration of a beep tone, SIP Server provides the option `beep-duration`, configured on the GVP Resource Manager `Trunk Group` DN.	• `beep-duration`
Updated timeout for CPD INFO messages SIP Server supports a modified `cpd-info-timeout` option.	• `cpd-info-timeout`

Table 76: SIP Server Feature Support for Outbound IP Solution (Continued)

Feature	Related Options
Additional Extensions in TMakePredictiveCall SIP Server supports the OCS ability to add CPD results as the keys in key-value pairs included the `Extensions` attribute of `TMakePredictiveCall` requests: • `AnsMachine` • `FaxDest` • `SilenceDest` Use these extensions—configured with the value `drop`—to override the corresponding SIP Server Application-level options `am-detected`, `fax-detected`, and `silence-detected`, in cases where any of these options are set to the value `connect`. If any of these key-value pairs appear in the `AttributeExtensions` of the `TMakePredictiveCall`, SIP Server disregards its own option settings and drops the call when the corresponding CPD result (`AM`, `FAX`, `Silence`) is detected. **Limitation** Currently, support for these OCS additions to the Extensions attribute is limited. For details, see the first item in the "Feature Limitations" list on page 312.	• `am-detected` • `fax-detected` • `silence-detected`
CPD Performed by Media Gateway If the media gateway is configured to perform CPD analysis, SIP Server sends the CPD result to GVP as parameters in the MSML dialog. For example, in the `MSML-INFO` message that SIP Server sends to start a treatment on GVP. Prior to release 8.1.0, CPD results were returned in either `INFO` messages for Audiocodes gateways, or in `200 OK` messages for Paraxip. If you prefer to use `INFO` message for Audiocodes, you can specifically configure that backwards compatible behavior using the option `info-pass-through`. For more information about configuring a media gateway trunk for CPD, see the *Voice Platform Solution 8.1 Integration Guide*.	• `cpd-capability` • `info-pass-through` (for backward compatibility)
CPD Performed by Genesys Media Server SIP Server enables you to improve the reliability of silence detection in deployments where CPD is performed by the Genesys Media Server. The `timeguard-reduction` configuration option can be used to make the time interval used by the Media Server for post-connect CPD shorter than the time SIP Server is waiting to receive a CPD result from the Media Server. Proper configuration can guarantee that SIP Server will always receive CPD results from the Media Server on time and will never be forced to provide its own default CPD result when its internal timeout expires. This feature applies mostly to silence detection because a CPD result of silence takes the longest time to be detected and the highest risk of SIP Server internal timer expiration. Use the `timeguard-reduction` option to improve silence detection quality.	• `cpd-info-timeout` • `timeguard-reduction`

Table 76: SIP Server Feature Support for Outbound IP Solution (Continued)

Feature	Related Options
Routing to Alternate Destination SIP Server supports routing the call to a special destination in cases where the CPD result shows that a fax, answering machine, or silence was detected on the other end. Alternate routing is implemented the following ways: • Routing Point-based calls SIP Server sends CPD results in either the `CallState` or `UserData` of the `EventRouteRequest`. The routing strategy can use this information to send the call to an alternate extension. • Trunk Group-based calls Starting with version 8.1.101.51, SIP Server attaches CPD results (using the `AnswerClass` key with values `AM`, `FAX`, or `SILENCE`) in `UserData` along with `CallState` of `EventEstablished/EventReleased` messages on a `Trunk Group` DN. The routing strategy can use this information to send the call to an alternate extension. **CPD in the CallState** Depending on the CPD results, SIP Server returns one of the following `CallState` values in the `EventRouteRequest/ EventEstablished`: • `CallStateFaxDetected` • `CallStateAnsweringMachineDetected` • `CallStateSilenceDetected` **CPD in UserData** Depending on the CPD results, SIP Server can return CPD results in UserData, using the `AnswerClass` key with one of the following values: • `AM` • `FAX` • `SILENCE` **Limitation** • For details about how SIP Server `am-detected`, `fax-detected`, and `silence-detected` options interact with OCS functionality, see "Feature Limitations" on page 312.	This functionality is implemented either in the routing strategy or in the OCS configuration, depending on the call flow. For more information, consult the following guides: • *Universal Routing 8.1 Reference Manual* • *Outbound Contact 8.1 Deployment Guide*

Table 76: SIP Server Feature Support for Outbound IP Solution (Continued)

Feature	Related Options
Cancelling Calls to Voicemail Number SIP Server can cancel outbound Predictive Calls when the outbound gateway returns a voicemail number in the `redirectNumber` of the `181 Call Is Being Forwarded` response to the initial outbound `INVITE` request. The `redirectNumber` header includes the number to which the call is being redirected, as well as the reason for the redirection. If SIP Server matches the number in the header to the configured voicemail pattern in the outbound `Trunk` configuration, SIP Server cancels the call, mapping the reason to a Genesys call state (for example, Busy). **Option Configuration** You can configure the `voicemail-pattern-<n>` option in the `Trunk` DN representing the outbound calling gateway. **Extensions Configuration** You can also use the key-value pair `voicemail-pattern` in the `Extensions` attribute of `TMakePredictiveCall` requests to match the `redirectHeader` to a particular voicemail number. Multiple patterns can be configured in a comma-separated list. If present in the `TMakePreditiveCall`, this value takes precedence over the Trunk configuration.	• `voicemail-pattern-<n>`

Media Server alarms
SIP Server supports two alarms used to report the state of the SIP Server subscription to the GVP Resource Manager (RM), for Media Server functionality. If the `SUBSCRIBE` message sent by SIP Server is rejected or times out, SIP Server generates the following alarm message:
`52005 - Media Server <NAME> failed to accept subscription. Subscription is unavailable.`
Once the subscription is reactivated (a new `SUBSCRIBE` message for an inactive subscription results in a `200 OK` from the RM), then SIP Server generates the following alarm message:
`52006 - Subscription to Media Server <NAME> is restored.`
These alarms are useful in cases where the RM is running but is unable to accept the subscription. In this case, the `Trunk Group` DN for outbound calls remains in service, but no subscription is available. The `52005` alarm reports this condition. If the subscription is later restored, the `52006` alarm is generated.
Note: If Active Out-of-Service Detection is configured for the outbound `Trunk Group` DN (`oos-check` option), SIP Server does not generate the `52005` and `52006` alarms. With Active Out-of-Service Detection, failures are reported using `EventDNOutOfService` and corresponding alarms, as per existing Active Out-of-Service Detection functionality. Genesys recommends using the Active Out-of-Service Detection method for monitoring Resource Manager availability; the `52005` and `52006` alarms are offered as backup.
	• `oos-check`

Table 76: SIP Server Feature Support for Outbound IP Solution (Continued)

Feature	Related Options
Predictive calls on a Routing Point DN SIP Server supports a modified `predictive-call-router-timeout` option.	• `predictive-call-router-timeout`
Mapping SIP error codes to Genesys call states SIP Server maps error codes received from the media gateway (in response to predictive-call `INVITE` requests) to specific values in the `AttributeCallState` included in the `TEvent` response to the `TMakePredictiveCall` request. By default, SIP Server uses a hard-coded map of error codes and T-Library messages. The hard-coded map is as follows, in the format `ErrorCode` —> `TEvent`: • 404 —> `CallStateSitInvalidnum` • 408 —> `CallStateNoAnswer` • 503 —> `CallStateGeneralError` • 504 —> `CallStateSitNocircuit` • 603 —> `CallStateDropped` • 486 —> `CallStateBusy`	• `sip-<SIP_error_code>`
Enhanced disaster recovery solution for outbound calls After receiving a negative response, SIP Server can now select an alternative trunk for outbound calls. In addition, SIP Server can now attempt to connect to a DN via an alternative softswitch (found in the DN configuration) if the first attempt to connect to a DN via a softswitch resulted in a negative response from that softswitch.	• `sip-error-codes-overflow`

Configuring the GVP DN for Outbound IP Solution

Table 77 describes configuration options for the Outbound IP solution, which you configure in the `TServer` section of the GVP Resource Manager `Trunk Group` DN.

Table 77: Configuring the GVP DN for Outbound IP Solution

Option	Setting
`contact`	Set this option to the Resource Manager IP address and SIP port (typically `5060`), using the following format: `sip:<RM_ip_address>:<RM_sip_port>`
`make-call-rfc3725-flow`	Set this option to `1`. This instructs SIP Server to use the 3pcc call flow as defined in the RFC 3725.
`refer-enabled`	Set this option to `false`. It forces SIP Server to use the re-`INVITE` method instead of `REFER`, as required for 3pcc calls.

Table 77: Configuring the GVP DN for Outbound IP Solution (Continued)

Option	Setting
ring-tone-on-make-call	Set this option to false (no ringtone is required for scenarios that may include CPD).
request-uri	Set the user part of the URI to msml, and identify the tenant-id as the name of the tenant. Format the value of this option as follows: sip:msml@<RMHost>:<RMPort>;media-service=cpd;gvp-tenant-id=[<tenant name>] For example: sip:msml@172.24.128.38:57393;media-service=media;gvp-tenant-id=[Environment] SIP Server sends an INVITE request to Resource Manager with the Request-uri modified to tell GVP to act as media server for the call.
subscription-id	Set this option to the name of the Tenant to which this Trunk Group DN belongs. For a single-tenant deployment, set this option to Environment. **Note:** Starting in GVP 8.1.2, multi-tenancy is supported.
cpd-capability	Set this option to the following: • mediaserver—This enables CPD analysis to be performed by the MCP. If CPD is to be performed on a media gateway instead, you do not need to configure this option on this DN. Instead, configure it on the Trunk DN for the media gateway (set to paraxip or audiocodes, depending on the gateway). For more information about configuring a media gateway trunk for CPD, see the *Voice Platform Solution 8.1 Integration Guide*.

Feature Limitations

- OCS is able to assign a value of either drop or a particular destination DN (to which the outbound call with a corresponding CPD result should be connected) in the key-value pairs added to the Extensions attribute of the TMakePredictiveCall request. However, SIP Server in this integration supports only the value drop. In this case, SIP Sever will drop the call if it detects these key-value pairs in the Extensions attribute, regardless of its own internal settings (am-detected, fax-detected, or silence-detected set to connect). In addition, if any of these SIP Server options are set to drop, SIP Server is unable to connect the call with a corresponding CPD result to the alternate destination DN provided by the OCS Extensions.
- After establishing a connection, certain SIP phones can disconnect the call if no RTP packets arrive before a predefined time period runs out. In the Outbound IP Solution, this disconnection can occur when an agent is

waiting for an engaged call to be connected to the customer. To avoid this issue, disable the RTP timeout feature in the configuration of the agent's SIP phone.

- When using the Paraxip gateway for Outbound IP, SIP Server cannot disable AM detection using the `TMakePredictiveCall` request. If you set the Extension `answer_type_recognition` to `no_am_detection` in the `TMakePredictiveCall` request, SIP Server might still report AM as the CPD result in the `EventEstablished` that it generates for the call.
- Early media for outbound predictive calls is not supported.

Overload Control

SIP Server includes an overflow control mechanism to gracefully handle situations where the load on the SIP Server exceeds its configured rate capacity threshold(s). If SIP Server encounters an overload situation, it first sends a warning stating that one of its monitored capacity levels has been reached. If the load continues to increase, SIP Server will then take graceful action to gradually reduce the load for the particular exceeded level, by rejecting calls with SIP `503 (Service Unavailable)` responses. SIP Server will send a warning cancelled message once the level is reduced to an acceptable level. In cases of severe overload, SIP Server will start immediately rejecting all new requests altogether.

How Overload Control Works

SIP Server monitors the current load levels for the following load factors:

- Dialog Rate—The number of newly created SIP dialogs per second.
- Call Rate—The number of newly created calls per second.
- T-Request Rate—The number of incoming T-Requests.

If SIP Server discovers the load level for any of these factors exceeds the acceptable limit, SIP Server takes action in three phases: Warning, Reaction, and Severe.

Stages of Overload Control Actions

1. First, when the dialog rate reaches 20% or higher above the configured `overload-ctrl-dialog-rate`, SIP Server issues a warning message stating that the maximum capacity for that level has been exceeded, and by what percentage.

   For example, if the dialog rate rises too far above the maximum capacity, SIP Server will issue a Warning alarm:

   ```
   52012|TRACE|GCTI_OVRLOAD_WARN_START|Overload warning mode
   DIALOGRATE started
   ```

2. If the dialog rate continues to rise past a certain critical percentage (30% above the configured dialog-rate), SIP Server sends a Reaction alarm message, and begins to take gradual action to reduce the load.

   `52014|STANDARD|GCTI_OVRLOAD_REACT_START|Overload reaction mode '%s' started`

   Note: The variable `'%s'` in this message represents the mode of the overload condition: `CALLRATE, DIALOGRATE`.

3. Once the load is reduced to a less than critical level, SIP Server cancels the Reaction alarm.

   `52015|STANDARD|GCTI_OVRLOAD_REACT_STOP|Overload reaction mode '%s' stopped`

4. Once the load is reduced even further and reaches an acceptable level, SIP Server cancels the Warning alarm:

   `52013|TRACE|GCTI_OVRLOAD_WARN_STOP|Overload warning mode '%s' stopped.`

Note: If SIP Server experiences a severe overload (overall call rate reaches 50% or higher than the `overload-ctrl-call-rate-capacity`), it immediately begins rejecting all new requests, sending Warning and Reaction alarms, in either order.

Figure 28 shows the different stages of overload control, and how the capacity rates determine how the stages are implemented.

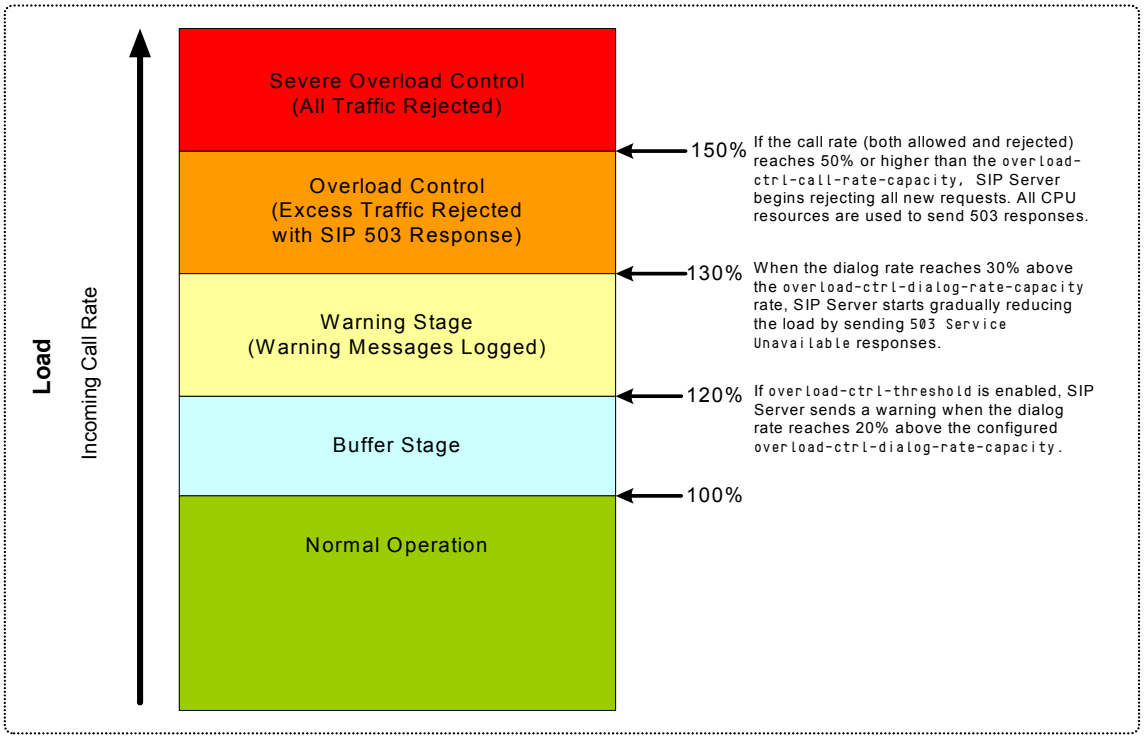

Figure 28: Stages of Overload Control Reaction

Recommended Settings

Figure 29 shows a sample historical graph representing peak periods of call volume, and how overload control settings should be set to handle the maximum demand experienced by SIP Server.

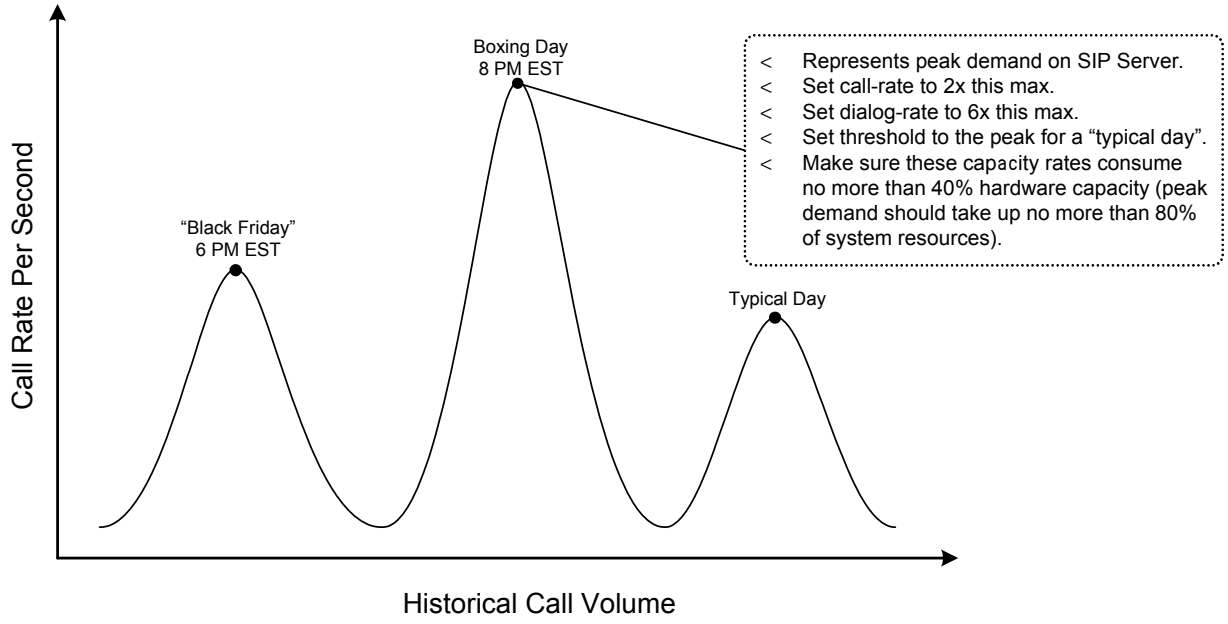

Figure 29: Sample Historical Call Volume and Recommended Overload Control Settings

T-Request Rate Control

If SIP Server encounters an overload situation with an excessive number of incoming T-Requests, it first sends a warning that states that the capacity level has been reached. If the load continues to increase, SIP Server takes a graceful action to reduce the load. If a T-Requests rate exceeds the capacity, SIP Server rejects all excessive T-Requests with `EventError` (error code `TERR_SERV_UNAVAIL118` and the following text: `Request rate exceeded threshold`). If the T-Requests rate for a given call exceeds capacity, SIP Server rejects either the corresponding or all excessive T-Requests for the given overloaded calls by sending the `EventError` (error code `TERR_SERV_UNAVAIL118`).

If a particular (`UserData` or `ApplyTreatment`) T-Requests rate for a given call exceeds capacity, SIP Server rejects excessive corresponding T-Requests for the given overloaded call with `EventError` (error code `TERR_SERV_UNAVAIL118`). SIP Server continues processing T-Requests for all other calls. It sends a warning canceled message once the T-Requests volume is reduced to an acceptable level for non-call-related thresholds.

Introduced in SIP Server 8.1.102.33

CPU Usage Overload Control

This feature provides the ability to control SIP Server's CPU usage overload by decrementing the server's log level when the CPU usage overload threshold is reached. Overload is detected by per-thread CPU usage measurement. CPU usage is checked every 10 seconds. If the CPU usage of any core SIP Server thread exceeds the value specified in the `log-reduce-cpu-threshold` configuration option, the log level is decremented to allow SIP Server to handle traffic more efficiently. Once the load drops below 40% of the `log-reduce-cpu-threshold` configuration option setting, it remains at that level for approximately 300 seconds; after that the logging level is restored to the initially configured level.

When configuring the overload threshold, keep in mind the following:

- The threshold value must not be configured too high; otherwise, the reduced logging can bring a risk not being enabled at all.
- The threshold must not be configured too low; otherwise, the lack of logging will make troubleshooting impossible in case of any failure.

Genesys recommends monitoring the SIP Server usage during a typical load spike period, detecting both the start and finish of the period, so the overload threshold is set appropriately.

In HA deployments, the primary and backup SIP Servers monitor and process overload conditions independently. For example, the primary server might be overloaded, while the backup server is not.

Feature Configuration

Table 78 describes how to configure overload control for any of the different overload factors.

Table 78: Configuring Overload Control

Objective	Related Procedures and Actions
Enable the Dialog Rate and Call Rate overload control mechanism.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section configure the following: • `overload-ctrl-threshold`—Set this threshold to the call rate (per second) at which SIP Server starts taking overload control action. **Notes:** • This option enables the overload control feature as a whole. • Value of this threshold must be at least four times smaller than `overload-ctrl-dialog-rate-capacity`, and at least two times smaller than `overload-ctrl-call-rate-capacity`.

Table 78: Configuring Overload Control (Continued)

Objective	Related Procedures and Actions
Set the capacities for individual load factors for the Dialog Rate and Call Rate control mechanism.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, set the thresholds for the individual load factors as follows: • `overload-ctrl-dialog-rate-capacity`—Set this factor to the SIP Dialog Rate (per second) capacity above which SIP Server begins taking corrective action. • `overload-ctrl-call-rate-capacity`—Set this factor to the Call Rate (per second) capacity above which SIP Server begins taking corrective action. **Notes:** • These options define the limits for the hardware on which SIP Server is running, so that SIP Server knows what action to take as these limits are approached. However, no action is taken unless the feature as a whole is enabled (using `overload-ctrl-threshold`). • SIP Server enforces a call-rate capacity at least 2x higher than the threshold, and a dialog-rate capacity at least 4x higher than the threshold. • The default values for these capacity rates assume that SIP Server is running on fairly robust hardware. For older machines, Genesys recommends setting these capacity rates to a lower value. You may also need to check that the `overload-ctrl-threshold` value is still correct.

Table 78: Configuring Overload Control (Continued)

Objective	Related Procedures and Actions
Enable T-Request Rate control.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following: • `overload-ctrl-trequests-rate`—Set this option to the T-Request rate (per second) that SIP Server is able to maintain without performance degradation. • `overload-ctrl-call-trequests-rate`—Set this option to the T-Request rate (per second) that is allowed for each call. This prevents performance degradation if particular clients issue too many requests. If the T-Request Rate for any particular call exceeds the configured value, SIP Server first sends a warning and, if T-Requests continues to increase, SIP Server rejects excessive T-Requests for that call. • `overload-ctrl-call-tupdateuserdata-requests-rate`—Set this option to the User Data update T-Request (`TAttachUserData, TUpdateUserData, TDeleteUserData, TDeletePair`) rate (per second) that is allowed for each call. If the T-Request Rate for any particular call exceeds the configured value, SIP Server first sends a warning and, if T-Requests continues to increase, SIP Server rejects excessive User Data T-Requests for a particular call. • `overload-ctrl-call-tapplytreatment-requests-rate`—Set the `TApplyTreatment` request rate (per second) that is allowed for each call. If the T-Request Rate for any particular call exceeds the configured value, SIP Server first sends a warning and, if T-Requests continues to increase, SIP Server rejects excessive `TApplyTreatment` requests for that particular call.
Disable the Dialog Rate and Call Rate overload control mechanism.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following: • `overload-ctrl-threshold`—Set this threshold to `0`. This will completely disable the Dialog Rate and Call Rate overload control feature.
Disable the T-Request Rate control mechanism.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, set the following options to `0`, as required: • `overload-ctrl-call-trequests-rate` • `overload-ctrl-call-tupdateuserdata-requests-rate` • `overload-ctrl-call-tapplytreatment-requests-rate` • `overload-ctrl-trequests-rate`
Enable CPU usage overload control.	In the SIP Server `Application` object > `Application Options` tab > `overload` section, set the `log-reduce-cpu-threshold` option as required.

P-Access-Network-Info Private Header

SIP Server supports passing the `P-Access-Network-Info` header, as described in RFC 3455 "Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)." This header can be used to provide access network, location, and emergency call information about the user agent. SIP Server supports the `P-Access-Network-Info` header only in `INVITE` and `UPDATE` messages.

The `P-Access-Network` header is passed only when both the calling and destination DNs are configured with `enforce-trusted` set to `true`.

Personal Greetings

Personal greeting functionality enables Genesys Media Server to play a media file to a customer and an agent when the agent answers the call. It is possible to play the same file or different files to the customer and agent. You can control how SIP Server handles the personal greeting feature in different scenarios.

This section describes the following greeting features and their configurations:

- VXML Support for Agent Greetings, page 321
- Disabling Media Before Greeting, page 324
- Recording an Agent Greeting, page 324

Filter Greetings By Call Type

Starting in 8.1.100.78, SIP Server lets you suppress agent greetings for different call types. You can block greetings for internal, consultation, and outbound calls, either globally at the Application level, or individually per Agent Login, by setting the `greeting-call-type-filter` option, as described below.

Enabling Personal Greetings

Table 79 describes the basic steps required to enable personal greetings.

Table 79: Enabling Personal Greetings

Objective	Key Procedures and Actions
(Mandatory) Configure the Agent Login.	In the SIP Server `Switch` object > `Agent Logins` > `Agent Login` object > `TServer` section, configure the following options: • `agent-greeting`—Set to the name of the media file that will be used as a greeting for the agent. • `customer-greeting`—Set to the name of the media file that will be used as a greeting for the customer. The customer greeting plays continuously until the agent greeting finishes playing. • (Optional) `greeting-call-type-filter`—Set to `internal`, `consult`, and/or `outbound` to block greetings for that type of call. **Note:** If, for whatever reason, one of these greetings cannot be played, SIP Server does not attempt to play the other greeting, but immediately connects the customer and agent. No greetings are played.
(Optional) Configure a routing strategy.	Enable personal greetings by specifying `agent-greeting` and `customer-greeting` keys in `AttributeExtensions` of the `TRouteCall` request. **Note:** The keys that are contained in `AttributeExtension` take precedence over the options specified in the `Agent Login` object. The customer greeting plays continuously until the agent greeting finishes playing.
(Optional) Configure the SIP Server Application.	Control how SIP Server provides greetings for different scenarios. In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following options: • `greeting-after-merge`—Set to `true` to enable greetings after transfers or conferences. • `greeting-call-type-filter`—Set to `internal`, `consult`, and/or `outbound` to block greetings for that type of call. • `greeting-delay-events`—Specifies when `EventOffHook` and `EventEstablished` events are sent. Set to `false` to send events before the greeting, set to `true` to send events after the greeting. If set to `true`, it might cause SIP Server to release the call on an agent DN in the middle of the greeting. See `greeting-stops-no-answer-timeout` for details. • `greeting-notification`—Set to `started` and/or `complete` to send notifications. • `greeting-repeat-once-party`—Set to `agent` to play agent greeting once, or set to `customer` to play customer greeting once. • `propagated-call-type`—Set to `true` in cases where SIP Server routes calls via ISCC to another SIP Server instance.

Feature Limitation

The following known limitations currently apply to personal greetings:

- The dynamically requested greeting (for example, through a `TRouteCall` request) is not supported in multi-site OOSP call routing scenarios with the following ISCC transaction types: `direct-uui` and `direct-notoken`.
- Greetings configured at an Agent Login are not supported for manual outbound calls (MakeCall).

VXML Support for Agent Greetings

Introduced in SIP Server 8.1.101.29

Support for multi-site and BC deployments added in 8.1.101.57

VoiceXML (VXML) support for agent greeting functionality allows an agent to accept, reject, transfer the call (arrived from a Routing Point), or redirect the call (using `TRedirectCall`) to a new destination.

When the agent answers the call, SIP Server informs GVP about the VXML file and Genesys Media Server starts its processing. VXML does the following:

- Might play the details about the call collected by URS to the agent.
- Prompts the agent to take action for the call—to accept, reject, or transfer the call to a new destination.
- Collects the result provided by the agent and passes it as user data to SIP Server. The VXML file can collect the result from the agent in the following ways:
  - By asking the agent to press the DTMF keys.
  - By asking the agent to say some words.

Media Server sends the user data `acceptcall` to SIP Server in the SIP `INFO` message which terminates VXML file processing. SIP Server receives the user data and based on that does the following:

- When the agent accepts the call, SIP Server adds the user data `acceptcall=true` to the call and connects the agent and the caller.
- When the agent rejects the call, SIP Server adds the user data `acceptcall=false` to the call and returns the call to the same Routing Point.
- When the agent transfers the call to other destination, SIP Server adds the user data `acceptcall=false` to the call and returns the call to the same Routing Point from which it is routed by URS to the other destination specified by the agent in user data.

Message Example

This is an example of the msml dialog.exit message sent by the MCP at the end of the VXML when an agent rejects the call:

```
INFO sip:7101@172.24.133.150:11000 SIP/2.0
```

```
From: sip:SVC_Mediaserver@UTE_HOME:11000;tag=C5EC0EA5-84A9-4611-
B864-03E9CBC10EC0-4
To: sip:7101@UTE_HOME:11000;tag=F5D150ED-A603-4532-ADF4-
8D8CB1272939-36
Call-ID: B38BDC0E-C1EB-4FB3-8071-D376DAE89C0F-31@172.24.133.150
CSeq: 1 INFO
Content-Length: 255
Content-Type: application/vnd.radisys.msml+xml
Via: SIP/2.0/UDP 172.24.133.150:53329;branch=z9hG4bK1F4DFBF2-472D-
4068-9747-12AF5BA6720E-3
Contact: <sip:172.24.133.150:53329>

<?xml version="1.0" encoding="UTF-8"?>
<msml version="1.1">
<event name="msml.dialog.exit" id="conn:__MSML-CONN-
ID__/dialog:ivr_application">
<name>acceptcall</name>
<value>false</value>
</msml>
```

Feature Configuration

Table 80 describes how to configure VXML support for agent greetings.

Table 80: Enabling VXML Support for Agent Greetings

Objective	Key Procedures and Actions
1. Configure the SIP Server Application.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following options for VXML functionality: • `greeting-after-merge`—Set this option to `false`. • `greeting-delay-events`—Set this option to `true`. • `greeting-repeat-once-party`—Set this option to `agent`. • `agent-reject-route-point`—(For multi-site deployments) Set this option to a valid Routing Point.

Table 80: Enabling VXML Support for Agent Greetings (Continued)

Objective	Key Procedures and Actions
2. Configure the MSML DN.	1. Create a DN of type `Voice over IP Service`. 2. In the `TServer` section, configure the following options: • `contact`—Set to the Resource Manager IP address and port. • `prefix`—(Optional) Set to `msml=`. Required for conference and monitoring services only. • `service-type`—Set to `msml`. • `subscription-id`—Set to the name of tenant (used for reliability). • `userdata-map-filter`—(Optional) Specify a prefix (or a list of prefixes) that must match the initial characters of the key in the `UserData` key-value pair, which SIP Server passes to GVP when agent greeting is played.
3. Configure a routing strategy.	1. Enable personal greetings by specifying `agent-greeting` and `customer-greeting` keys in `AttributeExtensions` of the `TRouteCall` request. 2. Enable VXML functionality by setting the `agent-greeting-type` key to `vxml`. Configure the URS strategy to collect some basic details about the call and to route the call to the agent with the agent greeting VXML file. The VXML file can be in a regular file directory (file://) or on a web server (http://). The `TRouteCall` request must have this VXML file along with agent greeting and customer greeting music files. 3. For multi-site deployments: The URS strategy must be able to route a call to the origination Routing Point on the origination SIP Server. URS can find this information from the `AttributeLastTransferOrigDN` in the `EventQueued` message.

Feature Limitations

VXML support for agent greeting has the following limitations:

- This feature is supported for MSML-based integration only.
- Customer greetings are only voice files. VXML files for customer greetings are not supported.
- This feature is not supported for greetings configured in the `Agent Login` object.
- The `greeting-delay-events` option does not support the `direct-uui` ISCC transaction type. Delaying `EventEstablished` until the agent accepts the call is not possible in `direct-uui` multi-site call flows.

Disabling Media Before Greeting

Introduced in SIP Server 8.1.101.50

SIP Server provides the ability to prevent establishing a preliminary audio/video connection between a caller and an agent before greetings are applied. This feature can be applied to scenarios where for a very short time a caller and an agent could hear each other before a greeting starts playing. SIP Server is able to disable the media connection between the caller and agent for that period of time before greetings are applied.

Feature Configuration

In the `TServer` section of the SIP Server Application (or in the DN object), set the `disable-media-before-greeting` configuration option to `true`.

Feature Limitations

The following known limitations currently apply to the Disabling Media Before Greeting feature:

- This feature is enabled only when a call is delivered to an agent from a Routing Point.
- This feature does not apply to a greeting after a two-step conference or transfer is completed.
- This feature does not apply when `TRedirectCall` is used by an agent to whom the call is routed.
- This feature does not work when early media is involved in a call.
- The phones must accept an initial `INVITE` with the hold SDP.
- In the case of the `INVITE` timeout from a Media Server, there is a delay in establishing a media path between a caller and an agent.
- This feature is enabled only when MSML is used for playing greetings.
- In the case of a multi-site call, this feature is enabled only for a greeting configured using `TRouteCall` extensions.

Recording an Agent Greeting

Introduced in SIP Server 8.1.102.26

You can configure SIP Server to record the agent call leg during the personal greeting. This feature works only when both recording and greeting are enabled for the call.

Feature Configuration

To enable recording of the agent call leg during the personal greeting:

1. In the `TServer` section of the SIP Server Application, configure the following options:
   - Set the `msml-support`option to `true`.

- Set the msml-record-support option to true.

2. Do one of the following:
   - Set the record-agent-greeting option to true in the TServer section of the SIP Server Application.
   - Set the record-agent-greeting key to true in AttributeExtensions of the TRouteCall request.

   If set at both places, the setting in AttributeExtensions takes precedence.
3. Do one of the following:
   - Set the record option to true on the agent's DN.
   - Set the record key to source or destination in AttributeExtensions of the TRouteCall request.
4. Enable personal greetings by specifying agent-greeting and customer-greeting keys in AttributeExtensions of the TRouteCall request.

Feature Limitations

The following known limitations currently apply to agent greeting recordings:

- This feature is supported for MSML-based integration only.
- This feature is supported only for greetings played for inbound calls.
- This feature is not supported for greetings configured in the Agent Login object.

Presence from Switches and Endpoints

Presence is an indicator of an agent's status regarding possible communication. Presence subscriptions allow SIP Server to receive notifications about the availability status for an agent endpoint and to distribute agent-state TEvents to its clients.

This functionality can be used when:

- An agent endpoint is behind a third-party softswitch, and that switch is able to provide a notification about the status change for the endpoint. In this case SIP Server is not engaged in signaling for each and every endpoint call, and the endpoint is not registered on SIP Server.
- Genesys Agent Desktop is not available to the agent, and the agent endpoint supports agent-status notification.

SIP Server can also accept subscription requests from an endpoint, in cases where the endpoint requires notifications regarding the status of a particular Extension or ACD Position DN.

SIP Server supports the following presence scenarios:

- "Subscription to SIP Server" on page 326
- "SIP Server Subscription to Endpoints Behind a Switch" on page 326

- "Endpoint Sends PUBLISH Requests to SIP Server" on page 327
- "Agent Login and State Update on SIP Phones" on page 329
- "Presence Integration with Microsoft Office Communications Server 2007" on page 331

Subscription to SIP Server

When a user subscribes to a particular `Extension` or `ACD Position` DN on the SIP Server switch, the user sends a `SUBSCRIBE` request to SIP Server, asking to receive notifications about the targeted DN. In response, SIP Server sends ongoing `NOTIFY` messages whenever the target DN registers with SIP Server, indicating whether the DN is in `open` status. If the targeted DN is not registered, or the registration has expired, SIP Server sends a `NOTIFY` message indicating a `closed` status for that DN.

No configuration on SIP Server is required to accept and process `SUBSCRIBE` requests.

SIP Server Subscription to Endpoints Behind a Switch

If SIP Server requires presence information about an endpoint that is behind a third-party switch—where the endpoint DN registers with the third-party switch, and not directly with SIP Server—you must configure a channel that SIP Server uses for sending the `SUBSCRIBE` request and receiving the subsequent `NOTIFY` messages regarding the presence state of the endpoint. In this case, a specially configured `Trunk` DN provides this channel.

Note: This presence mechanism is not compatible with other methods of modifying the agent state. For example, the "No-Answer Supervision" feature.

Notes:
- A separate `Trunk` DN or `Voice over IP Service` DN is required to make outbound calls. For more information, see "Configuring Devices and Services" on page 80.
- When you are configuring presence subscription for Microsoft Live Communication Server (LCS), you must also configure the "channel" `Trunk` DN as specified in "Remote Server Registration" on page 348.
- When Microsoft Office Communicator is integrated with Microsoft Office Outlook, and the presence state is set to `In a Meeting` or `Vacation`, LCS sends a presence notification with the `Busy` presence state to SIP Server. However, SIP Server is unable to provide `In a Meeting` or `Vacation` presence states to any subscriber. Instead, a notification with the `Busy` presence state is generated by SIP Server.

Procedure:
Enabling presence subscription

Start of procedure

1. Create a DN of type `Trunk` in the Configuration Layer. Parameters for all presence subscriptions from the SIP Server to a particular softswitch are configured in this `Trunk` DN.
2. Configure these options in the `TServer` section on the `Trunk` DN:
   - `contact`
   - `subscribe-presence-domain`
   - `subscribe-presence-from`
   - `subscribe-presence-expire`
3. Create a DN of type `Extension`.
4. Configure these options in the `TServer` section on the `Extension` DN:
   - `contact`
   - `request-uri`
   - `subscribe-presence`
   - `enable-agentlogin-presence`
5. Create an `Agent Login` object for each DN that will have subscription enabled. The `Agent Login` name must be equal to the `DN` object name. Each `Agent Login` object must be associated with an agent.

End of procedure

Note: Any internal calls that are made with this softswitch will not be monitored by SIP Server, and the agent state will be changed by SIP Server to `Not Ready`.

Endpoint Sends PUBLISH Requests to SIP Server

SIP Server changes the agent state in response to any notifications about presence state changes by using the `PUBLISH` request method. SIP Server accepts the `PUBLISH` request and provides automatic agent state updates based on any presence updates received within the `PUBLISH` request. SIP Server distributes notifications about presence updates to all subscribers based on the presence update received within the `PUBLISH` request.

SIP Server accepts `PUBLISH` requests when they are received for a DN in an internal domain. SIP Server processes the presence update from the `PUBLISH` request and distributes presence update notification to all subscribers for this DN.

The PUBLISH request functionality is enabled at the DN level in the Configuration Layer by specifying the subscribe-presence option. The value must be set to publish to indicate that presence change notifications are issued from the PUBLISH request.

Note: This functionality has been verified with the Eyebeam SIP endpoint when it is configured to work in Presence Agent mode. This mode enables PUBLISH request processing.

SIP Server updates the agent state when the agent login name matches the DN name. Agent updates are processed as follows:

- When SIP Server receives a presence notification with an open status, it performs the following steps:
  - Confirms if the agent is logged in. If the agent is not logged in, SIP Server sends an EventAgentLogin message.
  - Confirms if any activity is indicated in the presence notification.
    - If there is no activity, and if the agent is in a NotReady state, SIP Server sends an EventAgentReady message.
    - If there is activity, and if the agent is in a Ready state, SIP Server sends an EventAgentNotReady message and attaches the activity from the presence notification as the ReasonCode attribute.
- When SIP Server receives a presence notification with a closed status, it confirms that the agent is logged in. If the agent is logged in, SIP Server then sends an EventAgentLogout message.
- All notifications about the changes of an agent state are ignored when the agent is in the NotReady (AfterCallWork) state. The requested agent state is applied when the ACW time is over. For example, if an agent completes the call, SIP Server transfers the agent into the ACW state, and the PUBLISH request with an open status comes from the agent's SIP phone, then SIP Server does not change the agent state immediately. It waits for the ACW time to expire, and then places this agent into the Ready state. If in the same scenario SIP Server receives the PUBLISH request with a busy status from the agent's SIP phone, SIP Server will not change the agent state until the ACW timer is over, meaning that the agent remains in the NotReady state.

Note: An agent state cannot be modified using both SIP PUBLISH and T-Library requests. For example, if an agent is set to the NotReady state through the SIP PUBLISH (open/busy) request, that agent cannot be set to the Ready state through a T-Library request (RequestAgentReady). For this, SIP Server sends EventError with ErrorCode 506. Use the SIP PUBLISH (open/busy) request.

This also applies when AgentLogout is done using a T-Library request and AgentLogin is attempted using the SIP PUBLISH request. In this scenario, SIP Server simply ignores the SIP PUBLISH request. Use a T-Library request.

Agent Login and State Update on SIP Phones

Introduced in SIP Server 8.1.101.56

This feature enables an agent to perform agent-related operations from the phone and then synchronize the phone and agent's desktop. A typical scenario involves an agent using the phone exclusively to log in/log out and set the `Ready/Not Ready` status without using the agent desktop application. Or, if an agent prefers using the agent desktop, then with this feature, the agent login and state will be automatically updated on the phone display. SIP Server fully synchronizes agent actions that are done using the phone or the desktop.

This functionality is implemented using two subscription packages described in the *SIP Access Side Extensions Interface* document by BroadSoft:

- Application Server Feature Event Package
- Hoteling Event Package

SIP phones that support these subscriptions enable agents to perform the following operations without using the desktop:

- Log in and log out
- Change the state to `Ready`, `Not Ready`, or `AfterCallWork`
- Set/synchronize the `Reason` code for the `Not Ready` state

SIP Server distinguishes subscription requests by DN (the `From` field) and subscription type (the `Event` field).

Agent Login and Authentication

There is a difference between agent desktop and phone authentication. If an agent logs in to the phone first and enters the password, the agent still must enter the password on the desktop. If an agent logs in to the desktop first and enters the password, the agent gets logged in to the phone automatically. The agent does not re-enter the password to change the agent state or to log out.

Introduced in SIP Server 8.1.103.18

The `agent-allow-empty-password` configuration option, when set to `true`, enables an agent to log in from a SIP phone without the password. When `agent-allow-empty-password` is set to `false`, SIP Server rejects agent logging from a SIP phone without the password.

High Availability and Business Continuity Deployments

SIP Server synchronizes the agent state if a switchover occurs after the agent logs in from the phone or desktop. If the UDP transport is used, SIP Server continues sending agent state notifications to the phone through the existing subscription. If SIP is sent over TCP, it is expected that the phone should re-establish the TCP connection to SIP Server and use this connection to re-subscribe for agent state notifications from SIP Server. If the phone re-establishes the connection but does not re-subscribe, notifications are not sent. See "Feature Limitations" on page 330.

In Business Continuity deployments, phones must be configured with a single registration using the FQDN resolved in two IP addresses that correspond to

SIP Server peer 1 and SIP Server peer 2. See Business Continuity deployments in the *SIP Server High-Availability Deployment Guide* for details.

The Business Continuity recovery steps, if an agent uses both the desktop and phone, are as follows:

- The desktop remains in the logout state until it receives the registration request from a phone.
- The phone registers and subscribes.
- The desktop logs in automatically.

The Business Continuity recovery steps, if an agent uses only a phone, are as follows:

- The phone registers and re-subscribes.
- SIP Server sends a notification about the missing the agent-DN link and logout state.
- The phone can indicate this logout state or automatically re-log in.

Feature Configuration

- Enable the "ACD agent Availability" and "Hoteling Enhancement" features on the phone.
- If the ACD login operation on the phone requires agent authentication, provide the agent password in the `Agent Login` configuration object. Note that for agent authentication on the agent desktop, the desktop reads agent information from the `Person` configuration object.

Feature Limitations

- There is no synchronization of subscriptions between primary and backup SIP Servers. The phone must re-subscribe after the switchover.
- When an agent uses both the phone and desktop, the phone will not receive notifications after the switchover until the next `SUBSCRIBE` request.
- If you use phone-based agent operations, the `agent-emu-login-on-call` option must be set to `true` or not used at all.

Agent Login Control Using RFC 3863

Note: Agent login control using RFC 3863 functionality is maintained for backward compatibility with the older versions of SIP phones, which do not support Broadsoft extensions.

For IP phones that support agent-status updates initiated from the device—for example, where the user presses `Login`, `Logout`, `Unavail`, or `Avail` on the phone itself—the endpoint that represents the device can send `SUBSCRIBE` or `NOTIFY` requests to SIP Server, which SIP Server then maps into the corresponding T-Library events.

To enable this mapping, set the option `enable-agentlogin-subscribe` to `true` on the DN that represents the IP phone. In this case, when the agent presses `Login` and specifies their agent ID on the IP phone, the agent endpoint sends a `SUBSCRIBE` request to SIP Server. SIP Server generates an `EventAgentLogin` message. If the agent presses `Logout` on the phone, the endpoint sends another `SUBSCRIBE` request and an `EventAgentLogout` message is generated by SIP Server. To control agent-ready status, the agent endpoint sends `NOTIFY` requests with either an `open` or a `closed` status. SIP Server maps the `NOTIFY` request to an `EventAgentReady` or an `EventAgentNotReady` message, depending on the status.

If agent state was changed as the result of a `RequestAgentReady` or a `RequestAgentNotReady` message, SIP Server notifies the agent endpoint by using a `NOTIFY` request to update agent status on the IP phone.

Table 81 describes the required configuration for this feature.

Table 81: Mapping Agent Status

Objective	Key Actions and Procedures
Configure the DN.	In the `TServer` section of the `Extension` DN for the IP phone, configure the following: • `enable-agentlogin-subscribe`—Set this to `true`.

Presence Integration with Microsoft Office Communications Server 2007

For integrations with Microsoft Office Communications Server (OCS) 2007 R2, presence monitoring between the contact center and the Microsoft OCS environment is used to support the following:

- Ability for Microsoft OCS 2007 R2 to add a Genesys `Routing Point` DN, configured on SIP Server, as one of its contacts. This allows Microsoft OCS to push an `online` presence status for this contact.
- Ability for Microsoft Office Communicator or Yahoo Messenger Client to generate IM interactions towards this Routing Point contact.
- Ability for SIP Server to subscribe to the presence status of a Microsoft OCS 2007 R2 user—for example, an expert or knowledge worker in the Enterprise environment with a PSTN phone and Office Communicator—as well as to map this presence status to a Genesys Agent State. For example, an `online` status in Communicator maps to the Genesys Agent State `Ready`, `offline` maps to `Logged in/NotReady`, while other statuses such as `away`, `dnd`, or `busy` map to `NotReady`.
- Ability for the OCS knowledge worker to contact agents in the contact center either directly or through additional `Routing Point` DNs (other than the ocs-rp configured for basic presence and routing).

Table 82 describes how to integrate SIP Server with Microsoft OCS for presence monitoring, based on these assumptions:

- Access to Microsoft OCS is configured directly through the front-end server (not through Edge server).
- The OCS user has a PSTN phone with no Genesys client on which to login, or a SIP phone that cannot send `REGISTER`/`NOTIFY` or `PUBLISH` requests to convey presence. In these cases, the `ocs-dn` configuration is required.

Table 82: Configuring Presence for Microsoft OCS

Objective	Key Actions and Procedures
1. Configure Microsoft OCS.	In the Microsoft OCS environment, configure the following: • Add the SIP Server IP address to the list of Trusted Hosts/Server. ◦ IP address—Enter the IP address of the SIP Server host. ◦ Throttle As Server—Enable this setting. ◦ Treat As Authenticated—Enable this setting. • Create an account/user to represent the Routing Point that you will create in Step 4. For example, `ocs-rp` **Note**: The user name on OCS and DN name in the switch must match. • Create an account/user to represent the expert using Microsoft Communicator as the IM/Presence endpoint. • Add a static route through SIP Server. ◦ Domain—Enter the target domain. All requests by MOC users to parties at this domain will be sent to the IP address that you configure for SIP Server below. For example, `domain=callcenter.com` If the MOC user issues a request to `jack@callcenter.com`, OCS forwards the request to the static configured SIP Server IP address. ◦ IP address—Enter the IP address of the SIP Server host. ◦ Port—Enter the SIP port used by SIP Server. **Note**: You must restart the Front-End Server after configuring a static route. For details, consult the vendor documentation for Microsoft OCS 2007 R2.
2. Configure the SIP Server Application.	In the `TServer` section, make sure the following option is configured: • `sip-address`—Set this to the IP address of the SIP Server host machine (not the URI).

Table 82: Configuring Presence for Microsoft OCS (Continued)

Objective	Key Actions and Procedures

3. Configure the OCS Trunk.	In the SIP Server Switch, create a `Trunk` DN to represent the Microsoft OCS front-end server. In the `TServer` section, configure the following options:
	• `contact`—Set this to the IP address URI for the Microsoft OCS 2007 front-end server. You must also include the `transport` parameter, set to `tcp`. For example,
	`192.xxx.xxx.xxx;transport=tcp`
	• `force-online-state-lcs`—Set to `true`.
	This forces SIP Server to provide an `online` status for the ocs-rp to the OCS.
	• `force-register`—Set this to the Sign-in Name field (sip:username@domain) for the Route Point user you created in Step 1. Use a SIP URI format.
	For example,
	`sip:ocs-rp@your-ocs-address.com`
	• `prefix`—Set this to any non-duplicated value (no other `Trunk` with the same `prefix`).
	• Set the following options to the domain part of the Sign-in Name.
	◦ `subscribe-presence-domain`
	◦ `request-uri`
	◦ `override-domain`
	For example,
	`your-ocs-address.com`
	• `override-domain-from`—Set this to the IP address of the SIP Server host machine.
	• `subscribe-presence-expire`—Set this to the length of time between SUBSCRIBE requests.
	• `subscribe-presence-from`—Set this to the Sign-In Name field for the user you created in Step 1.
	For example,
	`sip:ocs-rp@your-ocs-address.com`
	• `ocs-dn`—Set this to a value of `register`.
	This enables SIP Server to view and map MOC user agent statuses.

4. Configure the Routing Point.	Create the `Routing Point` DN to be exposed in the Microsoft OCS environment. • The name must match the account/name configured in Step 1. For example, `ocs-rp`. • Set `subscribe-presence` to the name of the OCS Trunk used for subscription.

Table 82: Configuring Presence for Microsoft OCS (Continued)

Objective	Key Actions and Procedures

5. Configure the Extension DNs.	For each expert (using Microsoft Communicator as IM/Presence endpoint), create an `Extension` DN with the following options:
	• `contact`—Set this to the IP address or the front-end server. You must also include the `transport` parameter, set to `tcp`. For example,
	`192.xxx.xxx.xxx;transport=tcp`
	• `request-uri`—Set this to a URI consisting of the Sign-In Name field for the user (expert) you created in Step 1.
	For example,
	`bobparker@yourcompany.com`
	• `override-domain`—Set this to the domain served by the OCS.
	• `override-domain-from`—Set this to the IP address of the SIP Server host machine.
	• `sip-signaling-chat`—Set this to `session`.
	• `sip-chat-format`—Set this to `text`.
	• `ocs-dn`—Set this to the expert account/user name that you created in Step 1.

6. Add contacts in Microsoft OCS.	To view/access mutual statuses, add all users to the list of contacts for the Route Point, and add the Route Point to the list of contacts for all users. For more information, see "Feature Limitations" on page 334.

Feature Limitations

- Presence information regarding the status of the SIP Server Routing Point DN is available only to those Microsoft Communicator users authorized to view it. To enable a user to view the Routing Point status, you must do the following:
  - Run Microsoft Office Communicator once using the Routing Point user account as configured in the OCS 2007 Active Directory.
  - If additional visibility and access controls are necessary, add all agents authorized to view the status of the Routing Point as contacts.

  These steps are required to make changes to visibility. For a simpler solution, consult Microsoft documentation.
- Microsoft Office Communicator is unable to run on the same host as SIP Server.

Preview Interactions

Preview interactions allow agents to preview desktop interactions before receiving a call. SIP Server sends Preview Interaction messages to the desktop applications using the `EventPrivateInfo` message. The desktop application sends preview interaction messages using the `TPrivateService` request.

SIP Server sends a `previewInteractionRequest` message to the desktop application when it receives a `TRouteCall` request to a DN that is configured with the `preview-interaction` option set to `true`.

Starting with release 8.1.103.54, SIP Server now supports enabling the Preview Interactions feature using the `TRouteCall` request containing the `preview-interaction` key in `AttributeExtensions`. (The setting of the `preview-interaction` key in `AttributeExtensions` takes precedence over the DN-level `preview-interaction` configuration option.)

The desktop application responds with a `previewInteractionResponse` message to SIP Server. The `previewInteractionResponse` message provides SIP Server with information regarding the agent's ability to process the incoming interaction. The `status` field contains an `accepted` value or a `rejected` value that specifies if the agent will accept the interaction.

SIP Server sends a `previewInteractionAcknowledge` message to the desktop application after it receives the `previewInteractionResponse` message from it. This message informs the desktop application that the `previewInteractionResponse` message was successfully processed by SIP Server.

The `previewInteractionCancel` message is sent by SIP Server to an application in the following scenarios if there was an unsuccessful completion of a preview interaction:

- The preview timeout expired. SIP Server sends the `previewInteractionCancel` message with the `status` field set to `expired` to an application when the `previewInteractionRequest` message was issued but SIP Server did not receive a `previewInteractionResponse` message within the specified timeout value for the `preview-expired` option.
- The call was abandoned. SIP Server sends the `previewInteractionCancel` message to an application with the `status` field set to `canceled`.

Preview Interaction for IM

SIP Server supports the preview mechanism for Instant Messaging (IM) interactions as well. To enable this mechanism, set the `preview-interaction` option to `chat`. For more information, see "Preview Interaction" on page 251.

Providing a Caller ID

SIP Server supports providing caller ID information that is displayed on a destination party's phone, and replacing the caller ID with another number if necessary. This feature is supported using either of the following methods:

- The `Extensions` attribute with the `CPNDigits` key in the following messages:
  - `TMakeCall`
  - `TMakePredicitiveCall`
  - `TInitiateConference`
  - `TInitiateTransfer`
  - `TRouteCall`
  - `TSingleStepTransfer`
  - `TSingleStepConference`

  If the `CPNDigits` key is set in the `Extensions` attribute, the value of this key overrides the username provided in the URI in the `From` header of the `INVITE` message.
- The `cpn` option at the `Trunk` DN level. In this case, the caller ID information will be replaced by the SIP URI setting in this option for all outgoing calls through this `Trunk` DN.

Note: The `CPNDigits` key in `AttributeExtensions` of a T-Library request takes precedence over the `cpn` option set at the `Trunk` DN level.

Providing Call Participant Info

LCTSupervisor KVPs introduced in SIP Server 8.1.101.74

SIP Server can distribute information about all call participants—except the trunks allocated for communication between SIP Servers, and distribution devices (such as Routing Points or ACD)—to logged-in agents by using the SIP `NOTIFY` method and `EventUserEvent` messages. This information is primarily used by T-Library clients, such as Workspace Desktop, to display parties participating in the call.

The information about the call participants is reported in the `Extensions` attribute of the relevant event using the following key-value pairs:

- `LCTPartiesLength`—An integer that specifies how many parties are involved in a single call.
- `LCTParty<n>`—An integer that represents a party of the call, where `n` is an integer value starting from `0`.
- `LCTParty<n>_location`—A string that represents the name of the switch to which the DN belongs.

The supervisor-related information is reported in the `Extensions` attribute of the relevant event using the following key-value pairs:

- LCTSupervisor<n>—An integer that represents the supervisor of the call, where n is an integer value starting from 0.
- LCTSupervisor<n>_ location—A name of the switch to which this supervisor belongs.
- LCTSupervisor<n>_ monitoredDN—An integer that represents the agent monitored by this supervisor.
- LCTSupervisor<n>_ mode—Supervision mode.

A supervisor can switch between supervision modes and whenever there is change in supervision mode, SIP Server reports the change in EventPrivateInfo.

Using the EventUserEvent and EventPrivateInfo messages, Workspace Desktop could improve the customer experience by providing the accurate status of call supervision scenarios.

Note: Supervision mode is distributed only in the first EventUserEvent message generated immediately after a supervisor answers the call.

Sample Scenario

The following sample scenario describes the enhanced LCTParty interface with the supervision information:

1. Internal DNs 1001 and 1002 are provisioned on Switch A.
2. DN 1002 subscribes to monitor DN 1001 (mute mode, call scope).
3. Inbound call from DN 21001 on Switch B is routed to DN 1001.
4. Call supervision started.

SIP Server generates EventUserEvent—immediately after a supervisor answers the call—for DNs 1001@A and 1002@A with the following information:

```
EventUserEvent
  AttributeExtensions
    'LCTParty0'             '21001'
    'LCTParty0_location'    'B'
    'LCTParty1'             '1001'
    'LCTParty1_location'    'A'
    'LCTPartiesLength'      2
    'LCTSupervisor0'        '1002'
    'LCTSupervisor0_location' 'A'
    'LCTSupervisor0_mode'   'mute'
    'LCTSupervisor0_monitoredDN' '1001'
    'LCTSupervisorLength'   1
```

Note: In a multi-site environment, the Smart OtherDN Handling feature is not supported if you use Workspace Desktop and the LCTParty interface is activated in SIP Server.

Feature Configuration

Table 83 describes how to configure call info for agents.

Table 83: Configuring Call Info for Agents

Objective	Related Procedures and Actions
1. Configure the `Trunk` DNs.	In the `TServer` section, set the `sip-server-inter-trunk` option to `true` for DNs of type `Trunk` that are allocated for direct signaling between SIP Servers. The `NOTIFY` method will be sent only to sessions that are established through such trunks. For more information, see "Trunk Optimization for Multi-Site Transfers" on page 376.
2. Configure the SIP Server Application.	In the `TServer` section, set the `sip-enable-call-info` option to `true`. To provide call participants' locations and supervisor-related information, set the `sip-enable-call-info-extended` to `true`.

Feature Limitations

- When multi-site supervision is established with the `call` scope and if a monitored agent leaves the call, the requests submitted by the supervisor to switch between supervision modes will be rejected by SIP Server.
- When multi-site supervision is established with the `agent` scope and if consultation call supervision is started, the supervisor will not be aware of the consultation call even though the supervisor will be able to hear audio from the consultation call.

Providing Origination DN Name and Location in EventRinging

Introduced in SIP Server 8.1.101.85

SIP Server reliably provides the origination DN name and location in `EventRinging`. The agent desktop can use this information to collect extended data about the originating party, such as the agent name, and present it to the destination party while the phone is ringing. In particular, Workspace Desktop Edition displays this information in the "toast" window, which notifies an agent about a new incoming call.

This feature applies to all scenarios, including transfers, conferences, and call supervision in both single-site and multi-site deployments.

SIP Server adds two key-value pairs to EventRinging to implement new functionality:

- OriginationDN—The name of the origination DN
- OriginationDN_location—The name of the SIP Server switch to which the origination DN belongs

Event Examples

The value of OriginationDN provided in EventRinging is synchronized with the party name delivered through EventUserEvent of the LCTParty interface.

```
EventRinging
    AttributeExtensions
        'OriginationDN' '21001'
        'OriginationDN_location' 'Home'
    AttributeThisDN '7101'
    AttributeOtherDN '21001'
```

In the example above, the following LCTParty EventUserEvent will be distributed to DN 7101 when the call is established:

```
EventUserEvent
    AttributeExtensions
        'LCTParty0'         '7001'
        'LCTParty0_location' 'Home'
        'LCTParty1'         '21001'
        'LCTParty1_location' 'Home'
        'LCTPartiesLength' 2
    AttributeThisDN     '7101'
```

Origination Party Generation Rules

The following rules apply to the generation of origination party information:

- In calls made through a Routing Point, the Origination party for the TRouteCall destination will be the party that originated the call to the Routing Point.
- In single-step transfer (SST) scenarios, the Origination party for the transfer destination will be the party that originated the call to the transferrer. If the Origination DN of the transferrer has already been released from the call, then any other party except the transferrer will be added as OriginationDN.
- In supervision scenarios, the supervisor desktop will have the same origination DN as distributed for the monitored agent. In addition, if the monitored agent initiates a call, the origination DN for the supervisor will be the party present in the call instead of the monitored agent.

Table 84 shows the origination information (DN and location) distributed in single-site and multi-site scenarios based on the following information:

- **Home** and **East** sites are connected through ISCC.
- **Home** site has the following configuration:
  - Extensions: DN 7101, DN 7102, DN 7103
  - Routing Point: DN 5000
- **East** site has the following configuration:
  - Extension: DN 7901

Table 84: Example: Origination DN and Location in EventRinging

Scenarios	EventRinging Attributes and Extensions			
	AttributeThisDN	OriginationDN	OriginationDN_location	OriginationDN_location
7101 makes a call to 7102.	7102	7101	Home	7101
1. 7101 makes a call to 5000. 2. The call is routed to 7102.	7102	7101	Home	7101
1. 7101 makes a call to 7102. 2. 7102 issues a single-step transfer to 7103.	7103	7101	Home	7101
1. 7101 makes a call to 7102. 2. 7102 issues a single-step conference to 7103.	7103	7102	Home	Not available
1. 7103 monitors 7102. 2. 7101 makes a call to 7102. 3. Call supervision starts.	7103	7101	Home	7101
1. 7101 makes a call to 5000. 2. The call is routed to 7901 with CPNDigits=100100.	7901	7101	Home	100100
1. 7101 makes a call to 7102. 2. 7102 issues a single-step conference to 5000. 3. The call is routed to 7901.	7901	7102	Home	confXXX/msmlXXX

Feature Configuration

Enable the Call Participant Info functionality by setting the `sip-enable-call-info` configuration option to `true` in the `TServer` section of the SIP Server Application.

Quality of Service

SIP Server can set quality-of-service (QoS) bits to a user-defined value to prioritize SIP signaling traffic. Use the option `sip-ip-tos` on the SIP Server application to define the Type of Service (TOS) byte that SIP Server includes in the IP header of the SIP messages that it sends. Note that by configuring this option, you are not enabling QoS per se; instead, you are defining the packets so that the network engineer can then enable QoS.

On most operating systems, applications that are running on behalf of non-privileged user accounts are not permitted to set a non-zero TOS value, so you might have to perform additional actions to enable this functionality. In particular:

- On Linux, the application must have `CAP_NET_ADMIN` capability (that is, run from the root account).
- On Windows Server 2008, set the IP DiffServ bits on outgoing packets by defining the QoS Policy in the QoS Packet Scheduler, which is included in the operating system. For instructions about how to define the IP DiffServ bits on outgoing packets per executable or per port, see the Creating and Editing a QoS Policy document at the Microsoft website.
- On Windows Server 2012, create and configure a QoS policy as described in the Configuring Policy-based Quality of Service (QoS) document at the Microsoft website.

Note: When using this method on Windows, install Microsoft Hotfix referenced in MS14-031: Description of the security update for TCP for Windows (https://support.microsoft.com/en-us/kb/2957189):

- For Windows Server 2008 R2 SP1: http://www.microsoft.com/en-us/download/details.aspx?id=43143
- For Windows Server 2012: http://www.microsoft.com/en-us/download/details.aspx?id=43146
- For Windows Server 2012 R2: http://www.microsoft.com/en-us/download/details.aspx?id=43140

Refer to operating system documentation for additional information.

Remote Agents Support

SIP Server supports remote agents that use legacy PSTN phones. These agents could be working from their homes, or in a branch office that has simple PSTN connectivity.

SIP Server supports the following configurations for remote agents, depending on the remote agent locations:

- Remote agents located behind the softswitch (see Table 85)
- Remote agents located behind the SBC/gateway (see Table 86)
- Remote agents with non-provisioned phone numbers (see Table 87)

To learn about benefits of nailed-up connections and how to configure them, refer to “Nailed-Up Connections for Agents” on page 287.

To reconfigure office-based agents to their remote home-based locations, refer to the *Enabling office-based agents to work from home* topic in the *Supplement to the SIP Server Deployment Guide*.

Configuring Remote Agents

Configuring Remote Agents Located Behind the Softswitch

Table 85 describes how to configure remote agents located behind the softswitch.

Table 85: Configuring Remote Agents Located Behind the Softswitch

Remote agent location	VOIP Service DN: Softswitch configuration	Extension DNs configuration
Behind the softswitch	`[TServer]` • `contact` = <the contact URI that SIP Server uses for communication with the softswitch> • `prefix` = <the initial characters of the number that must match a particular softswitch for that softswitch to be selected> • `service-type` = `softswitch` • `refer-enabled` = `false` • `dual-dialog-enabled` = `false` • `reject-call-notready` = `true` (recommended, not mandatory) • `sip-cti-control` = <ensure that this option is not configured>	The Extension DN (Number property) for each remote agent must be configured with the PSTN number—for example, +1 555 123 1111, and contain no options. `[TServer]` <no options>
With nailed-up connections behind the softswitch	`[TServer]` • `contact` = <the contact URI that SIP Server uses for communication with the softswitch> • `prefix` = <the initial characters of the number that must match a particular softswitch for that softswitch to be selected> • `service-type` = `softswitch` • `refer-enabled` = `false` • `dual-dialog-enabled` = `false` • `reject-call-notready` = `true` (recommended, not mandatory) • `sip-cti-control` = <ensure that this option is not configured>	The Extension DN (Number property) for each remote agent must be configured with the PSTN number—for example, +1 555 123 1111, and contain the following options: `[TServer]` • `line-type` = `1`

Configuring Remote Agents Located Behind the SBC/Gateway

Table 86 describes how to configure remote agents located behind the SBC/gateway.

Table 86: Configuring Remote Agents Located Behind the SBC/Gateway

Remote agent location	Extension DNs configuration
Behind the SBC/gateway	`[TServer]` • `contact` = <the contact URI of the PSTN SBC/gateway, depending on the agent location> • `refer-enabled` = `false` • `dual-dialog-enabled` = `false` • `reject-call-notready` = `true` (recommended, not mandatory) • `sip-cti-control` = <ensure that this option is not configured>
With nailed-up connections behind the softswitch	The Extension DN (Number property) for each remote agent must be configured with the PSTN number—for example, +1 555 123 1111, and contain the following options: `[TServer]` • `contact` = <the contact URI of the PSTN gateway/SBC, depending on the agent location> • `refer-enabled` = `false` • `dual-dialog-enabled` = `false` • `reject-call-notready` = `true` (recommended, not mandatory) • `sip-cti-control` = <ensure that this option is not configured> • `line-type` = `1`

Feature Limitations

Due to the specifics of gateway behavior in performing SIP `REFER` methods, support for remote agents has some limitations. In order to use remote agents, you must perform one of the two following steps:

- Provision customers and remote agents to use physically separate gateways (otherwise, calls from agents to customers take shortcuts within gateways, which means that SIP Server loses track of the call and therefore cannot perform call control). Even in this configuration, direct calls between two remote agents on the same gateway are not visible to SIP Server.

Or,

- Disable the SIP `REFER` method for the gateways where the remote agents are located. This enables SIP Server to see agent-to-customer and agent-to-agent calls.

Configuring Remote Agents with Non-provisioned Phone Numbers

Introduced in 8.1.102.93

SIP Server improves provisioning of remote agent DNs in the Configuration Database. It is no longer required to provision external phone numbers (for example, agent's PSTN numbers) in the Configuration Database. You must create an `Extension` DN for each remote agent where a DN number can be a primary office DN number or any other number if an agent doesn't have a primary office DN.

The external phone number is used to reach the agent during the agent session only, thereby limiting the lifetime of the external phone number to a particular agent session. In other words, after the agent is logged out, any associations with that external phone number are removed.

The non-provisioned phone number to be used for the agent session is passed to SIP Server in the `TAgentLogin` request in `AttributeExtensions` as the `agent-phone` key. `AttributeThisDN` of that request will contain the agent DN configured in the Configuration Database.

This feature requires Workspace Web Edition (WWE) version 8.5.201.95 or later.

Table 87 describes how to configure remote agents with non-provisioned phone numbers.

.

Table 87: Configuring Remote Agents with Non-provisioned Phone Numbers

Remote agent location	VOIP Service DN: Softswitch configuration	Extension DNs configuration
Behind the softswitch	`[TServer]` • `contact` = <the contact URI that SIP Server uses for communication with the softswitch> • `prefix` = <the initial characters of the number that must match a particular softswitch for that softswitch to be selected> • `service-type` = `softswitch` • `refer-enabled` = `false` • `dual-dialog-enabled` = `false` • `reject-call-notready` = `true` (recommended, not mandatory) • `sip-cti-control` = <ensure that this option is not configured>	`[TServer]` <no options>
With nailed-up connections behind the softswitch	`[TServer]` • `contact` = <the contact URI that SIP Server uses for communication with the softswitch> • `prefix` = <the initial characters of the number that must match a particular softswitch for that softswitch to be selected> • `service-type` = `softswitch` • `refer-enabled` = `false` • `dual-dialog-enabled` = `false` • `reject-call-notready` = `true` (recommended, not mandatory) • `sip-cti-control` = <ensure that this option is not configured>	`[TServer]` • `line-type` = `1` • `connect-nailedup-on-login` = `gcti::park`

Feature Limitations

- If a non-provisioned phone number is used for the agent session, the agent can only initiate calls using the agent desktop. 1pcc calls originated from the non-provisioned phone number are not supported.
- For agents with nailed-up connections that use a non-provisioned number for the agent session, an establishment of the nailed-up connection by calling into a contact center routing point is not supported.
- Hunt Groups in Business Continuity (BC) functionality are not supported by this feature. That is, in the BC deployment, agent logging with a non-provisioned external phone number to a DN that is a member of the Hunt Group is not supported.

Remote Media on Genesys SIP Endpoint SDK 8.x

SIP Server supports remote 3pcc control of beep tones and DTMF tones generation using proprietary SIP extensions on custom endpoints built from the Genesys SIP Endpoint SDK 8.x:

- Beep Tones Control—If the DN is configured for it, SIP Server can initiate the playing of a specified audio file on the SIP endpoint for an active call, including during call recording. SIP Server uses `NOTIFY` messages with proprietary extensions to request the beep.
- DTMF Tones Control—If the DN is configured for it, SIP Server can initiate the generation of DTMF tones on the customer SIP endpoint. SIP Server uses `NOTIFY` message with proprietary extensions that provide the digits to be played by the endpoint.

Feature Configuration

Table 88 describes how to enable this feature.

Table 88: Enabling Remote Media Control on SIP Server

Objective	Key Procedures and Actions
1. Configure the DN.	In the SIP Server Switch > `DNs` folder > SIP endpoint DN > `Options` tab > `TServer` section, configure the following option: • `sip-cti-control`—Set this option to either, or both, of the supported values for this endpoint: `beep, dtmf`. **Note:** Use a comma-separated list if enabling both values.
2. Configure the SIP endpoint.	Both remote media features, Beep Tones Control and DTMF Tones Control, require customer configuration of the endpoint itself. For more information setting up a custom endpoint using the Genesys SIP Endpoint SDK 8.x, consult the *SIP Endpoint SDK 8.x API Reference*.

Remote Server Registration

SIP Server supports registering with a remote server under a specified account. The remote server registration is enabled on a per-`Trunk` DN basis. SIP Server registers `Trunk` DNs at a remote server when the `force-register` option is configured.

SIP Server also uses the values of the following options when registering with a remote server:

- `contact`, when determining where to send the `REGISTER` request.
- `password`, when the `REGISTER` request is challenged.

See Table 5, "Configuring Endpoints," on page 83 for more information about these options.

Remote Talk

The Remote Talk feature enables the answering of an incoming call remotely by a T-Library client, by sending the `TAnswerCall` request to SIP Server. For this feature to work, the `sip-cti-control` option must be set to `talk`.

The SIP endpoint must support the BroadSoft Application Server interface to use the Remote Talk feature for remote call control.

Secure SIP Signaling

Introduced in 8.1.103.08

SIP Server supports the secure SIP signaling schema, or `sips`, in accordance with RFC 5630.

When enabled, SIP Server forms the `Request-URI`, `From`, `To`, and `Contact` headers to include the `sips` schema when sending a SIP message to a device that requires that `sips` schema. The `Via` header of the message contains the transport TLS. When generating a response to an incoming message containing the `sips` schema, SIP Server forms the header `Contact` to include `sips`.

If the Request-URI with the `sips` schema also contains the `transport` parameter `transport=tcp` or `transport=tls`, communication will be established in secure TLS over TCP.

SIP Server applies the `sips` schema rules selectively, on a per call leg basis. In other words, if one SIP peer must communicate using secure SIP signaling while the other SIP peer does not support it, SIP Server is able to interconnect these peers using their supported protocol. However, devices communicating with SIP Server using the `sips` schema must be configured to enforce the `sips` schema.

Feature Configuration

To enable the sips schema for secure SIP signaling, add the sips parameter to the contact option of the required device, as follows:

```
contact=sips:[number@]hostport[;transport={tls/tcp}]
```

Genesys recommends that you configure transport=tls.

The sips schema is supported on the following types of DNs:

- Trunk
- Extension
- ACD Position
- Voice over IP Service with service-type=softswitch

Examples of the contact values with the sips schema:

```
sips:fly.example.com;transport=tls
sips:192.168.8.57;transport=tcp
```

Enforcing the sips Schema by SIP Registration

Self-registered DNs are configured with the option contact="*". When an incoming (from an endpoint) SIP REGISTER request contains the sips schema, SIP Server communicates with that endpoint using the sips schema. The transport parameter will be removed from the SIP REGISTER request.

Feature Limitations

- The sips schema is not yet supported by SIP Proxy.
- SIP Server guarantees consistency in using the sips schema only if it is configured and matches incoming traffic. In other words, the trunk through which an INVITE request containing sips arrives must have the sips schema configured and the self-registered DN must have the option contact="*" configured.
- If required to communicate with Media Server over TLS, Genesys recommends using the sip schema (not sips in the contact) to keep it backward compatible.

Sending Outgoing INVITEs with Multipart Body

Introduced in SIP Server 8.1.101.83

SIP Server now supports passing geo-location information formed by the routing strategy in the multi-part body of the outgoing `INVITE` message. The new functionality is triggered from the routing strategy by adding two key-value pairs to the `AttributeExtensions`: `SIP_MIME_HEADERS` and `Geolocation`:

- The `SIP_MIME_HEADERS` extension key consists of the following parameters separated by a colon (see "Mapping Examples" on page 350):
  - The name of the extension key containing an actual payload to be included in the outgoing `INVITE` body. The current supported extension key for this feature is `Geolocation`.
  - The content type for this payload, one of the IANA-registered MIME types. The current supported content type for this feature is `application/pidf+xml`.
- The value of the `Geolocation` extension key will be included as the body of the outgoing multipart `INVITE` message. No format check, no re-encoding and no other modifications to payload are made by SIP Server; the payload is included in the `INVITE` body as is.

SIP Server generates an outgoing `INVITE` message using the information provided in the two extensions described above, as specified by RFC 6442.

The feature can be triggered on any calls routed to the external number.

Mapping Examples

Example of TRouteCall

```
RequestRouteCall
      AttributeThisDN '5002'
      AttributeConnID 22660268d90ab001
      AttributeOtherDN    '22002'
      AttributeRouteType  1   (RouteTypeDefault)
      AttributeReferenceID    10
      AttributeExtensions
         'SIP_MIME_HEADERS' 'Geolocation:application/pidf+xml'
         'Geolocation'      '<?xml version="1.0" encoding="UTF-8"?>
                             <presence
xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
xmlns:gml="http://www.opengis.net/gml"
entity="pres:point2d@example.com">
                             <tuple
id="22c0e6a14348456597c8f02b5915a29b">
                             <status>
                             <gp:geopriv>
```

```
                              <gp:location-info>
                              <gml:Point
srsName="urn:ogc:def:crs:EPSG::4326"
xmlns:gml="http://www.opengis.net/gml">
                             <gml:pos>43.6198128 -70.2696997</gml:pos>
                              </gml:Point>
                              </gp:location-info>
                              </gp:geopriv>
                              </status>
                              <timestamp>2015-07-
16T13:07:06Z</timestamp>
                              </tuple>
                              </presence>'
```

Example of the corresponding outgoing INVITE

```
INVITE sip:22002@192.168.73.38:63081 SIP/2.0
From: <sip:msml=5593f1ad00000001@UTE_HOME:11001>;tag=CE972381-9AD5-
46EA-B8E9-43E45959890D-13
To: <sip:5002@UTE_HOME:11001>
Call-ID: 6FE4A45E-37B2-468B-B618-8A9D41F5B751-8@UTE_HOME
CSeq: 1 INVITE
Via: SIP/2.0/UDP UTE_HOME:11001;branch=z9hG4bKD0725BBB-9A0A-4A89-
981C-163DBD1F47A9-16
Contact: <sip:SVC_Mediaserver@UTE_HOME:11001>
X-Genesys-CallInfo: routed
Allow: ACK, BYE, CANCEL, INFO, INVITE, MESSAGE, NOTIFY, OPTIONS,
PRACK, REFER, UPDATE
Max-Forwards: 69
X-Genesys-CallUUID: UQS8MJGDDD0KD8IKDCUQC17F20000001
Session-Expires: 1800;refresher=uac
Min-SE: 90
Supported: geolocation,timer
Geolocation: cid:1430852104988
Content-Type: multipart/mixed;
boundary=845F3842_73B5_48B3_AC8A_15B65DA517FA
Content-Length: 947

--845F3842_73B5_48B3_AC8A_15B65DA517FA
Content-Type: application/sdp

v=0
o=PhoneSimulator 1 1 IN IP4 192.168.73.29
s=incoming INVITE
c=IN IP4 192.168.73.29
t=0 0
m=audio 63209 RTP/AVP 0
a=rtpmap:0 PCMU/8000/1

--845F3842_73B5_48B3_AC8A_15B65DA517FA
Content-Type: application/pidf+xml
```

```
Content-ID: 1430852104988
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
xmlns:gml="http://www.opengis.net/gml"
entity="pres:point2d@example.com">
<tuple id="22c0e6a14348456597c8f02b5915a29b">
<status>
<gp:geopriv>
<gp:location-info>
<gml:Point srsName="urn:ogc:def:crs:EPSG::4326"
xmlns:gml="http://www.opengis.net/gml">
<gml:pos>43.6198128 -70.2696997</gml:pos>
</gml:Point>
</gp:location-info>
</gp:geopriv>
</status>
<timestamp>2015-07-16T13:07:06Z</timestamp>
</tuple>
</presence>

--845F3842_73B5_48B3_AC8A_15B65DA517FA--
```

SIP Authentication

SIP Server supports SIP authentication for both incoming and outgoing calls in networks that require mutual authentication. If configured for mutual authentication, SIP Server can both challenge incoming `INVITE` requests and respond to challenges received from the switch for outbound `INVITE` request that SIP Server sends.

SIP Server also supports the authentication procedure for outgoing `REFER` requests in case of `401 Unauthorized` or `407 Proxy Authentication Required` responses that contain the `Authenticate` response header.

How It Works

SIP Server uses the HTTP Digest authentication method, in which `401 Unauthorized` or `407 Proxy Authentication` challenges are sent in response to `INVITE` and `REFER` requests where authorization is required.

Inbound Calls

For inbound calls, SIP Server issues these challenges when an `INVITE` is sent to a DN that is configured to demand authentication. If the response to this challenge includes the required authorization parameters, SIP Server can accept the follow-up `INVITE`.

Outbound Calls

For outbound calls, SIP Server receives the challenge from the switch. If authentication is configured on the outbound Trunk or softswitch, SIP Server can respond to this challenge by sending a new `INVITE` that includes the required authorization parameters.

For example, if the following outbound `INVITE`

```
INVITE
From: A(1001)
To: B(2002)
```

results in the challenge

```
407 Proxy Authentication
```

SIP Server, if the outbound Trunk or softswitch is configured for it, will resend the `INVITE` with additional parameters:

```
INVITE
From: A(1001)
To: B(2002); Proxy-Authorization: <Authorization Parameters>
```

If the parameters are correct, the new `INVITE` will be accepted by the switch.

Feature Configuration

Table 89 describes how to enable this feature.

Table 89: Enabling SIP Authentication

Objective	Key Procedures and Actions
1. Configure the inbound endpoint.	In the endpoint DN, in the `TServer` section, configure the following options: • `authenticate-requests`—Set this option to the value `INVITE`. Incoming `INVITE` requests to this DN will result in an authentication challenge. • `password`—Enter the password for realm authentication. The `INVITE` sent in response to the challenge must include this password, otherwise the call will not be authorized.
2. Configure the Trunk or softswitch.	In the `Trunk` or softswitch DN, in the `AuthClient` section, configure the following options: • `username`—Enter a username to be used in the response to the Digest challenge. • `password`—Enter the password to be used in generating the response. **Note:** On a softswitch DN, the `AuthClient` section configuration, will be mainly used by endpoints located behind the softswitch to authenticate requests to those endpoints.

SIP Proxy Support

Genesys SIP Proxy provides an alternative high-availability option without requiring a virtual IP address. In addition, it provides an interface for SIP communication between SIP devices and SIP Server components.

In a standalone deployment, each SIP Proxy serves one SIP Server HA pair per site. See the *SIP Proxy 8.1 Deployment Guide* for details.

Feature Configuration

Table 90 describes the required configuration for SIP Server to operate with SIP Proxy.

Table 90: Integration with SIP Proxy

Objective	Key Actions and Procedures
1. Configure the SIP Proxy Application.	In the SIP Proxy `Application` object > `Server Info`, set the following options: • `Host`—Specify the host on which this SIP Proxy is installed. • `Port IDs`—Specify the following SIP Proxy ports: ◦ `sip-port`, Connection Protocol: `sip` ◦ `http-port`, Connection Protocol: `http` (Optional) In the `Application Options` tab, create a section named `sipproxy`. In the `sipproxy` section, add the following options: • `applications`—For a multi-site environment with SIP Proxy support, specify the application names of all primary SIP Servers in the environment, separated by a comma. • `serving-sipserver`—Specify the application name of the primary SIP Server to which all requests from endpoints and media gateways will be forwarded. • `sipproxy-role`—Set this option to `10`. On the `Tenants` tab, add the tenants as necessary.
2. Configure primary and backup SIP Server Applications.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following options: • `sip-address`—Set this option to the IP address of the SIP Server interface. • `sip-outbound-proxy`—Set this option to `true`. • `sip-enable-rfc3263`—Set this option to `true`. • `sip-enable-gdns`—Ensure this option is set to `true`.

Table 90: Integration with SIP Proxy (Continued)

Objective	Key Actions and Procedures
3. Create a `Voice over IP Service` DN for each switch involved in a standalone environment.	Create a `Voice over IP Service` DN named, for example, `sip-outbound-proxy`. In the `Options` > `TServer` section, configure the following options: • `service-type`—Set this option to `sip-outbound-proxy`. • `contact`—Set this option to the SIP Proxy DNS-SRV name. • `external-contact`—Set this option to the SIP Proxy address using the `host:port` format. • `oos-check`—Specify how often, in seconds, SIP Server checks SIP Proxy for out-of-service status. • `oos-force`—Specify the time interval, in seconds, that SIP Server waits before placing an unresponsive SIP Proxy in out-of-service state when the `oos-check` option is enabled. **Note:** The Active Out-of-Service Detection feature (`oos-check` and `oos-force` options) must be enabled on a VoIP Service DN with `service-type=sip-outbound-proxy` for SIP Proxy support. See "Active Out-of-Service Detection" on page 240 for details.

SIP Traffic Monitoring

SIP Server actively monitors the level of SIP traffic that it receives, to initiate a switchover to the backup SIP Server if no messages are received after a configurable length of time.

How it Works

1. SIP Server sends `OPTIONS` messages to SIP devices as part of the Active Out-of-Service Detection (Active OOS) feature (see "Endpoint Service Monitoring" on page 239).
2. SIP Server actively monitors the length of time since it last received a SIP message—including the responses to the Active OOS `OPTIONS` messages.
3. If the length of time between SIP messages surpasses the maximum length configured out of all of the actively monitored DNs, then SIP Server reports `SERVICE_UNVAVAILABLE` to the Local Control Agent (LCA).

   Note: SIP Server initiates the switchover if no SIP messages are received during a period of time calculated based on `oos-check` and `oos-force` values of all DNs configured for Active Out-of-Service Detection.

4. The Solution Control Server (SCS) initiates the switchover from primary to backup SIP Server instance—the backup becomes primary and starts monitoring SIP traffic.

Feature Configuration

Table 91 describes the required configuration for SIP traffic monitoring.

Table 91: Configuring SIP Traffic Monitoring

Objective	Key Actions and Procedures
Enable Active Out-of-Service Detection.	You must configure at least one `Voice over IP Service` device for Active OOS Detection. In the `Voice over IP Service` DN, in the `TServer` section, configure the following options: • `oos-check`—Enter (in seconds) how often you want SIP Server to send `OPTIONS` messages to this device. • `oos-force`—Enter (in seconds) how long you want the device to be placed in out-of-service. Active Out-of-Service Detection must be enabled for one or more DNs. When SIP traffic monitoring is enabled, the primary SIP Server reports the `SERVICE_UNAVAILABLE` status to LCA/SCS when all devices configured with the Active OOS check have failed and no other SIP messages have been received for a period of time calculated based on the `oos-check` and `oos-force` values of all DNs configured for Active Out-of-Service Detection. For details, see "Endpoint Service Monitoring" on page 239.
Enable SIP traffic monitoring.	In the `TServer` section of the SIP Server `Application` object, configure the following option: • `sip-pass-check`—Set this to `true`.

Feature Limitations

- Monitoring of received messages can result in false-positive alarms triggering a switchover from primary to backup SIP Server instances in cases of a global outage or planned maintenance will cause all SIP messages to stop. To avoid unnecessary switchovers, Genesys recommends distributing the monitored DNs throughout the network.
- When updating or installing SIP Server, Genesys recommends you not enable the `sip-pass-check` option, as any oos-monitored DNs would be unavailable or non-operational during this period.

Shared Call Appearance

Introduced in SIP Server 8.1.101.57

SIP Server supports Shared Call Appearance (SCA) that enables a group of SIP phones to receive inbound calls directed to a single destination (shared line); that way, any phone from this group can answer the call, barge-in to the active call, or retrieve the call placed on hold.

Support for BC deployments added

Starting with version 8.1.101.75, SCA is supported in Business Continuity deployments. See the *SIP Server 8.1 High-Availability Deployment Guide* for details.

How It Works

The shared line has sub-lines called appearances. Each shared line has one or more appearances; each appearance can handle one call a time. The current status of each call (appearance) is displayed on each phone in the SCA group that includes outbound calls made from any phone in this group, which appear as they are placed from the same origination device.

There are several standards which enable implementation of SCA within the SIP protocol. Genesys SIP Server implemented the BroadWorks SCA standard that supports barge-in and is supported by leading phone manufacturers. Refer to your SIP phone documentation for information about SCA standards supported by your phone.

These are common scenarios where SCA can be used:

- Executive/Assistant—The call appearances on the executive's phone also appear on the assistant's phone. The assistant may answer incoming calls to the executive and then place the call on hold for the executive to pick up. The assistant can always see the state of all calls on the executive's device.
- Key System Emulation—Multiple lines are shared across two or more phones. A call answered on one phone can be put on hold and picked up on another phone. Another phone can be added/joined/bridged to an existing appearance resulting in a conference call.
- Single Line Extension—Several phones are formed in to a group to which incoming calls arrive. When one device answers, the other phone are informed. If another phone in the group goes off hook, it is immediately bridged or joined in with the call.
- Changing devices—A user is on a call on one phone and wishes to change phones and continue the call on another phone. The user places the call on hold, notes the appearance number of the call, then walks to another phone. Users are able to identify the same appearance number on the other phone, pick up the call, and continue the conversation.

Note: This feature may also be referred to as Bridged Line Appearance (BLA) or Shared Line Appearance (SLA).

Shared Call Appearances are configured using two types of DNs:

- Primary shared line DN—The Address of Record (AoR), such as 7000 in the example above.
- Secondary DN—Other DN associated with the Primary shared line DN.

User Experience

- Incoming calls to a Shared Call Appearance ring on all the associated phones.
- The status of every call is shown on all phones associated with the Shared Call Appearance.
- Calls are always associated with a “line appearance”. Incoming calls will be assigned the lowest numbered idle line appearance. All phones associated with the Shared Call Appearance should have the same number of “line appearances” configured, typically with each line appearance having a dedicated “line key” button.
- A user may seize (go off hook) a particular line appearance if it is idle by pressing the corresponding line key button. For example, pressing the second line key will seize (go off hook) the second line appearance when it is idle.
- Held calls may be retrieved by any phone associated with the Shared Call Appearance.
- An active call on a phone associated with the Shared Call Appearance may be joined at any time by another phone associated with the Shared Call Appearance. This is sometimes referred to as a “barge-in.” The parties are then conferenced together.
- Each phone associated with the Shared Call Appearance might have only one active call at a time, and other calls will be held.
- Outgoing calls from any line appearance of the Shared Call Appearance will present an outgoing caller ID with the identity of the Shared Call Appearance. (A phone could have other lines not associated with the SCA, and these are not impacted, they would present a different caller ID).

Note: According to the BroadWorks SCA standard, one DN cannot be a member of multiple shared lines. If, for example, an executive assistant needs to share lines with two executives, two independent shared lines must be configured on the assistant's phone. All of them are displayed at the screen and operable.

Sample Call Flow

A sample call flow for a Shared Call Appearance scenario is as follows:

1. Two phones are configured with a Shared Call Appearance of 7000 and all are idle. In this example, they are referred to as Phone A and B, and both are configured to show two line appearances.
2. An incoming call to 7000 rings on both phones using the first line appearance.
3. A user at Phone A answers the call. Phone B reflects the call is active on another phone on the first line appearance.
4. A second incoming call to 7000 rings on both phones on the second line appearance.
5. The user at Phone A places the first call on hold. Phone B reflects the initial call is held on the first line appearance.
6. The user at Phone A answers the second call. Phone B reflects the second call is active on another phone on the second line appearance.
7. The user at Phone B retrieves the held call from the first line appearance. Phone A reflects the call is now active on another phone on the first line appearance.

SCA and Other Feature Interaction

- Call Recording can be set for a particular shared line DN, Primary and/or Secondary DN.
- Call Monitoring can be set for a particular shared line DN, Primary and/or Secondary DN. However, neither Primary nor Secondary DN can monitor other DNs. If, during monitoring, a call placed on hold is retrieved by another shared line DN, the monitoring will be dropped.
- Greetings can be set for a particular shared line DN, Primary and/or Secondary DN.
- Greetings and Barge-In—A shared line user can barge-in to an established call with two parties while a greeting is in progress, after which all three parties will be connected.
- Hunt Groups—A shared line DN cannot be a member of a Hunt Group.
- Routing—Only routing to a Primary shared line DN is supported (and all phones will ring). Routing to a Secondary DN directly is not supported. A shared line DN can make a call to a Routing Point using one of the shared line appearances—the same way as for any call.
- Call Pickup—An inbound SCA call cannot be picked up by a DN rather than a shared line DN. However, if an inbound call is ringing on a regular non-shared line DN, it can be picked up by a shared line DN.

- Call Park/Retrieve—Shared line users can park a call, and the call can be retrieved from any phone (shared line or regular phones) using the Primary shared line number. There can be only one parked call per shared line at a time. Shared line users can retrieve calls that were parked by regular phones.
- Dial Plan—For inbound calls, SIP Server applies dial plans only to resolve call destinations; that is, only digit translation of the selected rule is performed, no additional parameters of the selected rule (`timeout`, `ontimeout`, `onbusy`, and so on) is applied. If a destination is the Primary shared line DN, a call delivered to the SCA number is treated as a regular SCA call, i.e. is ringing on Primary and Secondary DNs. No more dial plan rules are applied after that. For outbound calls, shared line DNs dial plans are applied—for example, if a Secondary DN makes an outbound call, the dial plan configured for that Secondary DN is applied.

SCA Messaging

SCA related data is transported using the Call-Info and Line-Seize Event Packages. They are used in shared line call-related messages (`INVITE`, `180 Ringing`, `SUBSCRIBE`, and so on).

SIP Server reports T-Library events separately for each Primary and each Secondary DN. No events are generated for a shared line itself.

Feature Configuration

Table 92 describes how to configure Shared Call Appearance. (See also "How Configuration Changes Take Effect" on page 361 and "Configuration Example" on page 362.)

Table 92: Configuring Shared Call Appearance

Objective	Key Actions and Procedures
Configure a Primary shared line DN.	1. Create a DN of type `Extension` with the number where all incoming calls will be delivered. 2. In the `Options` > `TServer` section, set the following options: • `shared-line`—Set this option to `true`. • `shared-line-capacity`—(Optional) Set this option to specify a number of shared line appearances, which limits the maximum number of simultaneous calls per shared line. • `authenticate-requests`—Set this option to register for enabling an authentication procedure on DN registration. • `password`—Set this option to a valid password to be used for authentication of the Primary shared line DN.

Table 92: Configuring Shared Call Appearance (Continued)

Objective	Key Actions and Procedures
Configure a Secondary shared line DN.	1. Create a DN of type `Extension` with the number to be used as a Secondary DN. • In the `Options` > `TServer` section, set the `shared-line-number` option to the value of the Primary DN. 2. On the SIP phone that supports SCA specify the following properties (the exact property names vary): • DN number—Must be set to the same value as the DN number in the DN object for a Secondary DN. • Line type—Must be set to Shared Line, BroadSoft SCA, or equivalent. • Authentication username—Must be set to the same value as the Primary DN. • Authentication password—Must be set to the same value as the password option configured for the Primary DN. 3. Repeat the above steps for each Secondary DN to be used as a shared line user.

How Configuration Changes Take Effect

If a regular DN (neither Primary nor Secondary shared line DN) is changed to be a Primary or Secondary DN in the Genesys configuration, SIP Server does the following:

- Continues processing DN's existing calls as non-shared line DN calls.
- Delivers and processes new inbound calls as SCA calls. Outbound calls from this DN can be barged-in or retrieved by other shared line users.
- Does not send `NOTIFY` messages with appearance statuses to this DN until it subscribes to SCA statuses. To force the DN to subscribe, it must be reconfigured as a BroadWorks SCA DN. Until then, it is not able to barge-in or retrieve calls served by other shared line users.

If a Primary or Secondary DN is changed to be a non-shared line DN, SIP Server does the following:

- Continues processing of existing calls for this DN.
- Processes new inbound/outbound calls as non-shared line calls.

Stops sending `NOTIFY` messages with appearances statuses to this DNs.

Configuration Example

In the configuration example, the Primary shared line DN is 7000. The Secondary DNs are 7001 and 7002.

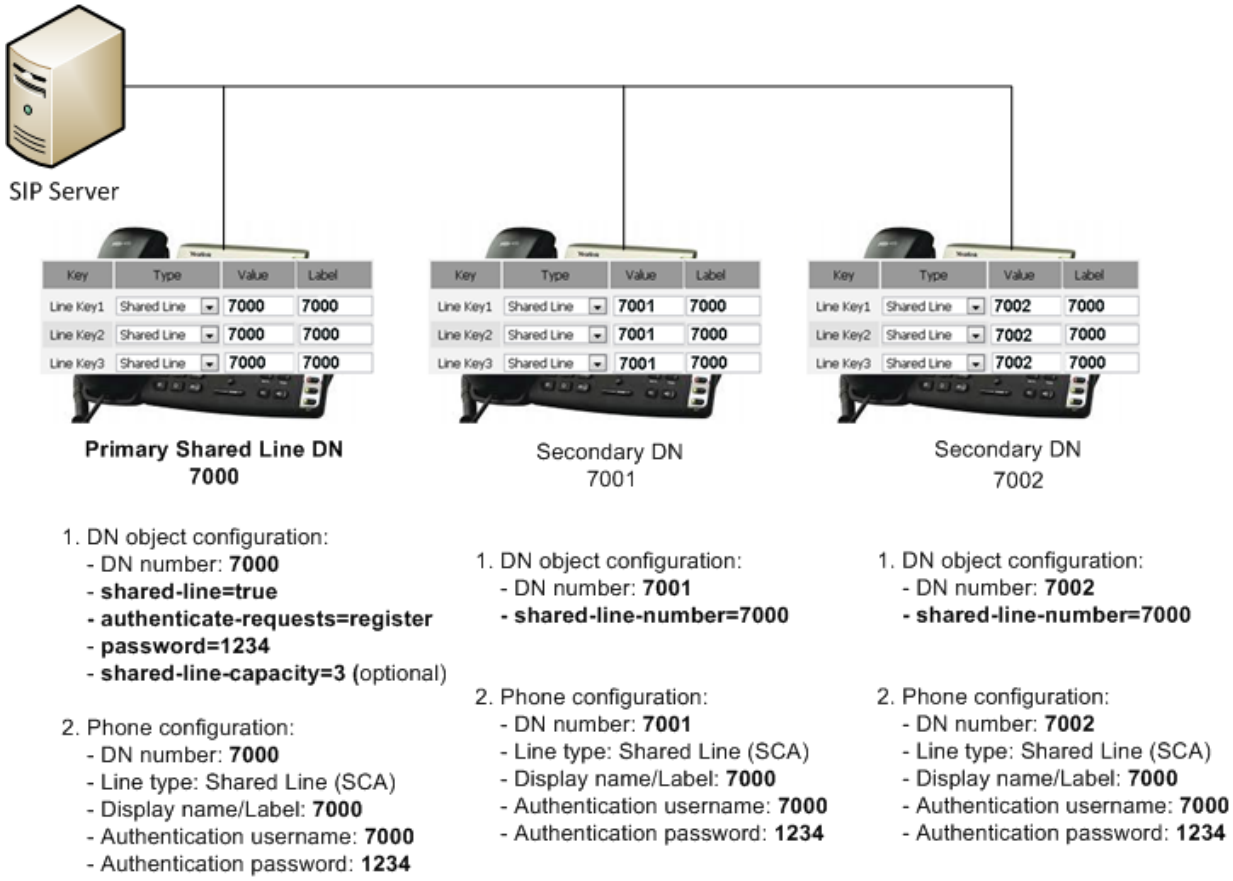

Figure 30: Shared Call Appearance Configuration Example

Feature Limitations

- Only 1pcc operations are supported.
- One DN cannot be a member of multiple shared lines.
- Calls to Secondary DNs are not supported. Customers may chose to disable calls to Secondary DN numbers through a dial plan.
- Private Hold SCA Broadsoft functionality is not supported.
- Agent login to SCA DNs (Primary or Secondary) is not supported.
- Multi-site scenarios with the `direct-notoken` ISCC transaction type to a shared line destination DN is not supported. (No `EventRinging` reporting if the call is answered by a Secondary DN.)
- `TRouteCall` to a Secondary DN is not supported. See “SCA and Other Feature Interaction” on page 359.

- ICON version 8.1.400.08 or earlier might not report redirect scenarios for SCA calls correctly.
- In inbound call scenarios, no 3pcc requests can be processed before a call is answered by a shared line user.
- SCA DNs (Primary or Secondary) cannot be located behind the softswitch.
- Semi-attended transfers and Mute transfers to the shared line are not supported.
- The ringing state of a call on DNs in the shared line appearance deployment is not properly synchronized from the primary SIP Server to its backup. If a switchover occurs while the call is ringing on several DNs, the call may be dropped.

Smart OtherDN Handling

For T-Library clients that provide the Agent ID value as the `OtherDN` in requests to SIP Server, SIP Server can convert this `OtherDN` value using its knowledge of the association between the Agent ID and the DN to ensure the correct execution of the request by the switch.

Supported Requests

Table 93 shows the requests that assume the use of the `OtherDN` value as a switch directory number, and can therefore support Smart OtherDN Handling.

Table 93: Requests That Support Smart OtherDN Handling

TRequest	Meaning of OtherDN Attribute	AgentID-to-DN Conversion
TMakeCall	Call destination	Yes
TMakePredictiveCall[a]	Call destination	No
TRedirectCall	New destination for a call	Yes
TInitiateTransfer	Call destination	Yes
TSingleStepTransfer	New destination for a call	Yes
TInitiateConference	New destination for a call	Yes
TSingleStepConference	New destination for a call	Yes
TDeleteFromConference	Conference member to be deleted	Yes
TCallSetForward[b]	Request target	Yes

Table 93: Requests That Support Smart OtherDN Handling (Continued)

TRequest	Meaning of OtherDN Attribute	AgentID-to-DN Conversion
TGetAccessNumber[c]	DN for which Access Number is requested	No
TSetCallAttributes[c]	Not specified	No
TMonitorNextCall	Agent DN to be monitored	Yes
TCancelMonitoring	Agent DN that was monitored	Yes
TRouteCall[d]	New destination for a call	
• RouteTypeUnknown		Yes
• RouteTypeDefault		Yes
• RouteTypeOverwriteDNIS		No
• RouteTypeAgentID		No

a. `TMakePredictiveCall` assumes the directory number should be outside the switch; however, this request could also support Smart OtherDN Handling.

b. `TCallSetForward` has a separate flag in the configuration option for enabling conversion.

c. T-Server cannot intercept these requests.

d. Only the listed route types are applicable for OtherDN conversion.

Feature Configuration

Table 94 describes how to configure Smart OtherDN Handling for SIP Server.

Table 94: Configuring Smart OtherDN Handling

Objective	Key Actions and Procedures
Enable for all applicable calls.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following: • `convert-otherdn`—See the option description for a list of valid values.

Table 94: Configuring Smart OtherDN Handling (Continued)

Objective	Key Actions and Procedures
Enable on a call-by-call basis.	The extension key `ConvertOtherDN` can be used to enable this feature on a call-by-call basis. Configure the routing strategy T-Library client to include the `ConvertOtherDN` key in the `Extensions` attribute of the T-Library request to SIP Server. Set the value of this key to one of the following: • `0`—disables all conversions for the call. • `1`—forces the relevant conversion for the call.

Feature Limitation

In a multi-site environment, the Smart OtherDN Handling feature is not supported if you use Workspace Desktop and the LCTParty interface is activated in SIP Server.

SRV Address Support in Contact and Record-Route Headers

Introduced in SIP Server 8.1.102.50

SIP Server supports the SRV FQDN—FQDN resolving to SRV records—received in the `Contact` or `Record-Route` headers of a SIP message. SIP Server also supports the SRV FQDN in the `contact` option on a Trunk DN.

If the target destination received in the URI of the `Contact` or `Record-Route` headers of a `200 OK` message is not a numeric IP address, and no port is present, SIP Server performs an SRV query to obtain the target's IP `address:port`. The `OPTIONS` messages are sent over all transports representing SRV records. The `ACK` messages and all further SIP requests are sent to the active transport with the highest priority. SIP Server uses the original transport if it is among the active transports with the highest priority. If no active SRV records are found, the SIP transaction fails.

If the target destination received in the URI of the `Contact` header of an `INVITE` message is not a numeric IP address, and no port is present, SIP Server performs an SRV query to obtain the target's IP `address:port`. The `OPTIONS` messages are sent over all transports representing SRV records. Further SIP requests are sent to the active transport with the highest priority. SIP Server uses the original transport if it is among the active transports with the highest priority. If no active SRV records are found, SIP Server uses the transport of the original `INVITE` message for further SIP requests.

When SIP Server is deployed with SIP Proxy (the Application-level option `sip-outbound-proxy` is set to `true`) and it must send a SIP request to a

destination configured with the SRV FQDN or list of active transports, SIP Server selects an active target destination and adds the private `X-Genesys-Route` header with a value of `sip:IpAddress:Port[;transport=tcp/tls]`. SIP Proxy uses the value of the `X-Genesys-Route` header as the next destination for forwarding the request. For SIP Proxy, this header has priority over the target specified in `Request-URI` or `Route` headers. SIP Server uses the same transport value for the `X-Genesys-Route` header until a transport becomes out of service.

Feature Configuration

Table 95 describes how to configure SIP Server to perform an SRV query.

Table 95: Configuring SIP Server to perform an SRV Query

Objective	Key Actions and Procedures
Configuring SIP Server to perform an SRV query.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following: • Set `sip-enable-gdns` to `true`. • Set `sip-enable-rfc3263` to `true`. • If SIP Proxy is used, set `sip-enable-x-genesys-route` to `true`. In a multi-site SRV/DNS-based configuration: • Set `sip-address-srv` to the SRV FQDN. • Set `sip-address` to the hostname of the SIP Server interface. • Set `sip-port` to any valid port. • Set the `contact` option to the SRV FQDN on inter-site Trunk DNs.

Feature Limitations

SIP Server does not support the SRV FQDN in `REGISTER` messages.

Strict SIP Endpoint Registration

Introduced in SIP Server 8.1.104.02

In standalone mode, SIP Server can restrict SIP endpoint registration if its IP address is not included in a list of trusted IP addresses. When SIP Server receives a SIP `REGISTER` request from a SIP endpoint, it verifies the endpoint's IP address. You configure a list of trusted addresses using the `sip-registrar-allowlist` configuration option. If the `REGISTER` request is arrived from an untrusted IP address, SIP Server rejects the request with an error code defined by the `sip-registrar-reject-code` option.

Feature Configuration

Table 96 describes how to configure SIP Server to perform an SRV query.

Table 96: Configuring SIP Server to Restrict SIP Endpoint Registration

Objective	Key Actions and Procedures
Configuring the SIP Server Application.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following: • Set `sip-registrar-allowlist` to a list of IP addresses. • Set `sip-registrar-allowlist-origin` to `via` or `contact`. • Set `sip-registrar-reject-code` to a valid SIP error message, or leave it the default 403 error code.

Transport Layer Security for SIP Traffic

SIP Server supports secure communication for both the SIP traffic as well as the T-Library communication that it engages in. SIP Server uses the Transport Layer Security (TLS) protocol to secure both modes of communication—configured separately in the `TServer` section for SIP messaging, and in the common configuration options for T-Library communication.

Note: This guide presents information for configuring TLS for SIP messaging only. For details about how to configure the common TLS options for T-Library communication, see the *Genesys Security Deployment Guide.*

About TLS

SIP Server supports the standard TLS protocol which offers confidentiality, integrity protection, and data compression to client/server applications. For a detailed description of how this protocol works, as well as how security works generally, refer to the relevant RFCs:

- "RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2"
- "RFC 4568: Session Description Protocol (SDP) Security Descriptions for Media Streams"
- "RFC 3711: The Secure Real-time Transport Protocol (SRTP)"

Feature Configuration

Table 97 describes how to configure TLS for the SIP messaging that SIP Server engages in.

Table 97: Configuring TLS

Objective	Key Actions and Procedures
Prepare server and clients.	Use the Microsoft Management Console (MCC) tool to generate and install server certificates. For details, see the *Genesys Security Deployment Guide*.
Prepare Solaris, Linux, or AIX server and clients.	For Solaris, Linux, or AIX, use the Genesys Security Pack to generate certificates. For details, see the *Genesys Security Deployment Guide*.
Configure the SIP Server Application.	In the `TServer` section of the SIP Server `Application` object, configure the following TLS-related options: • `sip-port-tls`—Set this to the SIP port on which SIP Server listens for incoming requests using TLS encryption. **Note:** If you configure ONLY this parameter, SIP Server will use the default host certificate, if one is available. In this case, SIP Server will take the default values for `sip-tls-cert`, `sip-tls-cert-key`, and `sip-tls-trusted-ca` from the `Host` object. • `sip-tls-cert`—For Windows, set this to the thumbprint obtained from the **user certificate** generated for the host. For UNIX, set this option to the path and filename of the `.pem` encoded file that contains the **host certificate**. • `sip-tls-mutual`—Set this to `true` for SIP Server to request the client certificate to initiate a mutual TLS connection. • `sip-tls-target-name-check`—Set this to `host` and SIP Server compares the subject field in the server's certificate to the target host name. If no match is found, the connection fails. • `sip-tls-cipher-list`—Specifies the list of ciphers.
Configure the SIP Server Application (continued).	**Solaris, Linux, or AIX-Only Options** In addition to the options above, Solaris, Linux, or AIX deployments also require the following configuration: • `sip-tls-cert-key`—For Solaris, Linux, or AIX only, set this to the path and filename of the `.pem` encoded file that contains the **host private key**. • `sip-tls-crl`—For Solaris, Linux, or AIX only, set this to the name of the file containing one or more certificates in PEM format, to define the **Certificate Revocation List**. • `sip-tls-trusted-ca`—For Solaris, Linux, or AIX only, set this to the path and filename of a `.pem` encoded **Certificate Authority (CA)** file, containing one or more certificates in PEM format.

Table 97: Configuring TLS (Continued)

Objective	Key Actions and Procedures
Configure the device DN.	To enable TLS for SIP communication to any SIP device described in this guide, configure the destination DN that represents the device as follows: • contact—Append the value for the contact option with the following string: transport=tls For example, For an Extension DN, in the TServer section, set the contact option to the IP address and port number of the host computer, followed by the tls string: 100.100.100.101:5060;transport=tls

Treating Incoming Calls As Inbound Calls

Introduced in SIP Server 8.1.103.35,

SIP Server can treat incoming calls from external callers (agents behind SIP trunks) as inbound calls.

To enable this feature:

- Set the enforce-1pcc-inbound option to true.
- (Optional) Set the internal-call-domains option to a list of IPv4 CIDR blocks or FQDN separated by semicolons (;).

SIP Server Feature Processing Logic

To take advantage of this feature and, if you use the enforce-external-domains option in your environment, Genesys recommends that you gradually transition from using the enforce-external-domains option to using the enforce-1pcc-inbound option.

The enforce-external-domains option has higher priority than the enforce-1pcc-inbound option. If configuration options of both approaches are applied, SIP Server verifies the new incoming call INVITE message multiple times, as follows:

1. SIP Server verifies the domain part of the From header of the INVITE message against the value of the enforce-external-domains option:
   - If a match is found in the enforce-external-domains option, SIP Server treats the call as inbound.
   - If a match is not found, SIP Server proceeds to Step 2.
2. SIP Server verifies the value of the enforce-1pcc-inbound option:

- If the value of the `enforce-1pcc-inbound` option is set to `true`, SIP Server proceeds to Step 3.
- Otherwise, SIP Server proceeds to Step 5.

3. SIP Server verifies the value of the `internal-call-domains` option:
   - If the value of the `internal-call-domains` option is empty, SIP Server treats the call as inbound.
   - If the value of the `internal-call-domains` option is not empty, SIP Server proceeds to Step 4.
4. SIP Server verifies the `Via` header of the `INVITE` message against the value of the `internal-call-domains` option:
   - If a match is found in the `internal-call-domains` option, SIP Server proceeds to Step 5.
   - If a match is not found in the `internal-call-domains` option, SIP Server treats the call as inbound.
5. SIP Server verifies only the username part in the From header in the `INVITE` message against the internal DNs:
   - If the username matches an Extension or ACD Position DN, SIP Server treats the call as internal.
   - If the username matches a Routing Point or Trunk Group DN, SIP Server rejects the call.
   - If a match is not found, SIP Server treats the call as inbound.

Tromboning Control

In multi-site routing, where transfers from other T-Server or SIP Server instances are made through ISCC, trunk tromboning can sometimes occur. By default, SIP Server performs internal resource matching on incoming `INVITE` requests, analyzing the headers to see if there are matches to any internal resources on the corresponding switch. However, there are cases where the `username` in the `From` or `Contact` header of the incoming `INVITE` does match the name of an internal DN, but by coincidence. In this case, internal resource matching should be turned off for this call, to prevent SIP Server from incorrectly treating an external call as internal.

Duplicated DN Names

If two SIP Server instances at different sites have DNs with matching numbers, tromboning can occur. For example,

1. SIP Server A and SIP Server B both have an `Extension` DN with the number `9999`.
2. SIP Server A sends an `INVITE` on behalf of DN 9999 to SIP Server B.
3. SIP Server B employs internal resource matching, and mistakenly considers the call to be internal.

To control this behavior, configure the `enforce-external-domains` option in SIP Server to include the computer names or IP addresses for all SIP gateways or hosts associated with other T-Servers or SIP Servers that SIP Server may communicate with over ISCC. SIP Server checks this list of computer names or IP addresses against the computer names or IP addresses specified in the URI of the `From` header of incoming `INVITE`. If there is a match, then the DN is considered external and the DN name is formed using the `DN.domain` format.

About the DN.domain format

In cases where SIP Server finds a naming match between an external and an internal DN, it forms the DN name for the external DN using the `DN.domain` format. This format clearly differentiates the two DNs for further call processing.

Typical DN Name

For example, a typical DN might appear in T-Library messaging as follows (see **bold**):

`AttributeOtherDN '2099'`

External DN Name

While if the same DN is found to be external, SIP Server would form the DN name in this way:

`AttributeOtherDN '2099.10.208.139.30'`

SIP Server uses this `DN.domain` format in the following cases:

- If the DN is considered external, and it matches an internal DN that is either registered or marked as `in service`, SIP Server uses the `DN.domain` notation for the external DN.
- If an `INIVTE` from a particular DN contains the `X-Genesys-PartyInfo` header (used for communication between multi-site SIP Servers) and this DN matches an internal registered or `in service` DN, then SIP Server uses the `DN.domain` format to identify the external DN.

Bounced Calls Between T-Servers

Bounced calls are calls sent back and forth between SIP Server and other SIP or T-Server instances. For example,

1. SIP Server A sends a call on behalf of DN 9999 to SIP Server B.
2. The call arrives at a Routing Point on SIP Server B, which routes the call back to SIP Server A.
3. Instead of considering this a new external call, SIP Server A instead matches it to DN 9999, starting a consultation call for that DN.

When routing an internal 1pcc call, SIP Server preserves the hostname of the incoming call in the `From` header of the outgoing `INVITE`. This hostname often belongs to SIP Server itself. In this case, you cannot add SIP Server's own address to the `enforce-external-domain` option, otherwise 1pcc calls from internal DNs will also be excluded from resource matching.

To control this behavior, configure override-domain-from on the Trunk DN that points to the second SIP Server instance (for example, in the switch for SIP Server A, configure the trunk pointing to SIP Server B with override-domain-from). You must then add the value of this option to the enforce-external-domains list.

Feature Configuration

Table 98 describes how to configure tromboning control.

Table 98: Configuring Tromboning Control

Objective	Related Procedures and Actions
1. Configure basic anti-tromboning.	In the SIP Server Application object > Application Options tab > TServer section, configure the following option: • enforce-external-domains—Enter a semicolon-separated list of hostnames or IP addresses for each external SIP or T-Server from which you expect to receive ISCC transfers.
2. Configure anti-tromboning for bounced call routing.	1. In the Trunk DN that points to the second SIP Server instance, configure the following option: ◆ override-domain-from—Enter an identifier for the originating SIP Server. For example, the name of the switch corresponding to the SIP Server object. 2. In the second SIP Server object, add the identifier you created in Step 1 to the enforce-external-domains list.

Trunk Capacity Control

SIP Server enables control of the number of outgoing and incoming calls to be handled by a specific trunk or a group of trunks in single-site deployments. SIP Server rejects calls when trunk capacity is reached. Only traffic to and from a single SIP Server HA pair is controlled. In Business Continuity deployments, capacity control must be configured at each site.

Capacity Control of Outgoing Calls

When capacity control is enabled on a trunk, SIP Server keeps a count of every incoming and outgoing call, including every SIP or T-Library request, that it receives. When this count equals the value specified by the capacity configuration option, SIP Server starts rejecting only outgoing calls, generating accompanying messages depending on the call control type, as follows:

- 1pcc calls are rejected with a SIP error code specified in the capacity-sip-error-code configuration option.

- 3pcc calls are rejected with an `EventError` containing `ErrorCode` specified in the `capacity-tlib-error-code` configuration option.

Example

```
[TServer]
capacity=100
```

With this setting:

- If there are 50 incoming and 50 outgoing calls established through the trunk, SIP Server rejects an attempt to make an outgoing call through this trunk, but accepts incoming calls arriving to this trunk.
- If total calls are less than 100, both incoming and outgoing calls are allowed.

Capacity Control on a Group of Trunks

Trunks can be defined as one capacity group by using the `capacity-group` configuration option. When capacity control is enabled on a group of trunks, SIP Server keeps a count of every incoming and outgoing call, including every SIP or T-Library request, that it receives. When this count equals the value specified by the `capacity` configuration option, SIP Server starts rejecting only outgoing calls, generating accompanying messages depending on the call control type, as follows:

- 1pcc calls are rejected with a SIP error code specified in the `capacity-sip-error-code` configuration option.
- 3pcc calls are rejected with an `EventError` containing `ErrorCode` specified in the `capacity-tlib-error-code` configuration option.

Example

DN of type `Trunk` with the name `Trunk1` and the following options:

```
[TServer]
capacity=200
capacity-group=TrunkGroup1
prefix=8340
```

DN of type `Trunk` with the name `Trunk2` and the following options:

```
[TServer]
capacity-group=TrunkGroup1
prefix=8341
```

With these settings, the number of calls to the `Trunk1` and `Trunk2` will be limited to 200. When the limit is reached, SIP Server rejects attempts to make an outgoing call through these trunks, but accepts incoming calls arriving to these trunks.

Capacity Control of Incoming and Outgoing Calls

To control both incoming and outgoing calls, configure the capacity-limit-inbound configuration option on the same Trunk DN where the capacity option is defined. This capacity control mode is applicable only to DNs of type Trunk.

Example 1

```
[TServer]
capacity=100
capacity-limit-inbound=true
```

With these settings:

- If total calls are less than 100, incoming and outgoing calls are allowed.
- When the limit is reached (for example, 60 incoming and 40 outgoing calls), incoming and outgoing calls are rejected.

Example 2

DN of type Trunk with the name Trunk1 and the following options:

```
[TServer]
capacity=200
capacity-limit-inbound=true
capacity-group=TrunkGroup1
prefix=8340
```

DN of type Trunk with the name Trunk2 and the following options:

```
[TServer]
capacity-group=TrunkGroup1
prefix=8341
```

With these settings, the number of calls to Trunk1 and Trunk2 will be limited to 200. When the limit is reached, incoming and outgoing calls are rejected.

Feature Configuration

Table 99 describes how to configure Trunk Capacity Control for SIP Server.

Table 99: Configuring Trunk Capacity Control

Objective	Key Actions and Procedures
Configure capacity control of outgoing calls on a trunk.	**DN Level.** On a DN (type Trunk, or type Voice over IP Service with service-type=softswitch), specify the following configuration option in the TServer section: • capacity **Application Level.** Specify these options in the TServer section of the SIP Server Application, as required: • (Optional) capacity-sip-error-code • (Optional) capacity-tlib-error-code

Table 99: Configuring Trunk Capacity Control (Continued)

Objective	Key Actions and Procedures
Configure capacity control of incoming and outgoing calls on a trunk.	**DN Level.** On a DN of type `Trunk`, specify the following configuration options in the `TServer` section: • `capacity` • `capacity-limit-inbound` **Application Level.** Specify these options in the `TServer` section of the SIP Server Application, as required: • (Optional) `capacity-sip-error-code` • (Optional) `capacity-tlib-error-code`
Configure capacity control of outgoing calls on a group of trunks.	**DN Level.** On a DN (type `Trunk`, or type `Voice over IP Service` with `service-type=softswitch`) that defines capacity and a trunk group to which capacity is applied, specify the following configuration options in the `TServer` section: • `capacity` • `capacity-group` For all other trunks included in the same group to which capacity is applied, specify the following configuration option in the `TServer` section: • `capacity-group` **Application Level.** Specify these options in the `TServer` section of the SIP Server Application, as required: • (Optional) `capacity-sip-error-code` • (Optional) `capacity-tlib-error-code`
Configure capacity control of incoming and outgoing calls on a group of trunks.	**DN Level.** On a DN of type `Trunk`, specify the following configuration options in the `TServer` section: • `capacity` • `capacity-limit-inbound` • `capacity-group` For all other trunks included in the same group to which capacity is applied, specify the following configuration option in the `TServer` section: • `capacity-group` **Application Level.** Specify these options in the `TServer` section of the SIP Server Application, as required: • (Optional) `capacity-sip-error-code` • (Optional) `capacity-tlib-error-code`

Trunk Optimization for Multi-Site Transfers

SIP Server supports trunk optimization for multi-site transfers. When the trunk optimization functionality is in use, the `OtherDN` attribute contains correct information and is reported properly in `EventPartyChanged` messages in the following scenarios:

Scenario 1

Figures 31 and 32 show the state of the call before and after the multi-site transfer.

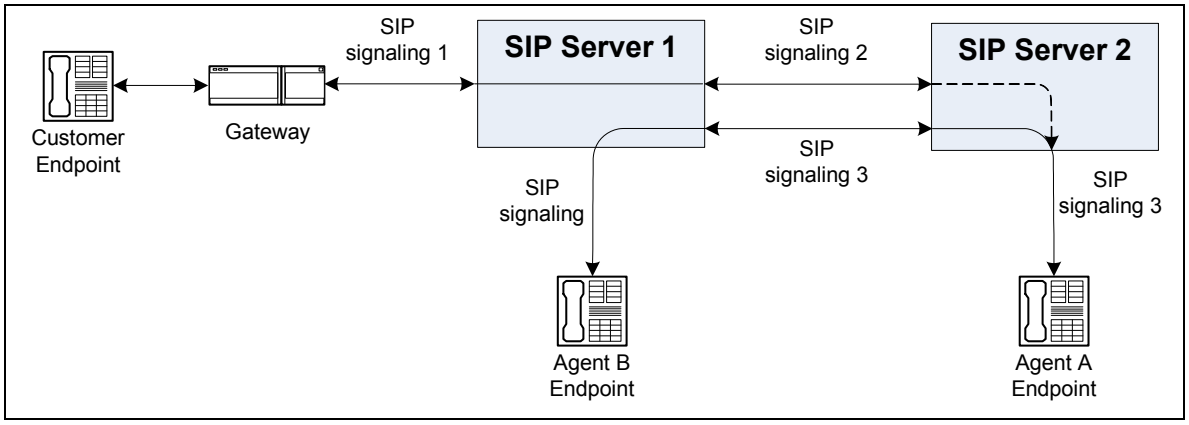

Figure 31: Call Before REFER with Replaces Transfer

1. An inbound call is routed to Agent A at the SIP Server 2 site.
2. Agent A initiates a two-step transfer to Agent B at the SIP Server 1 site.

In this scenario, SIP Server uses a SIP `REFER` request with the `Replaces` header to report call data for Agent B.

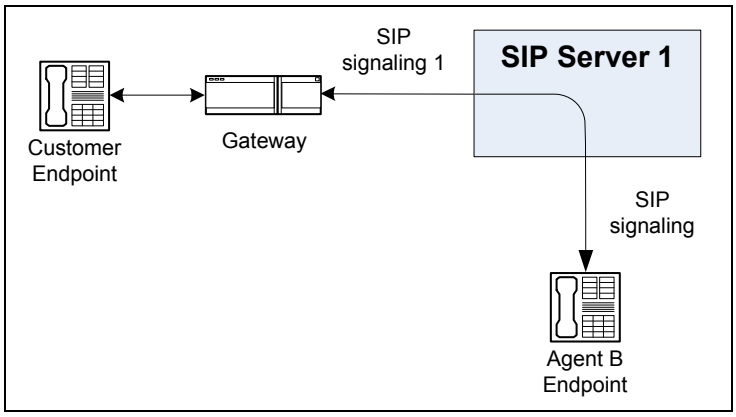

Figure 32: Call After REFER with Replaces Transfer

After the transfer is completed, both the transferring agent (Agent 1) and secondary SIP Server (SIP Server 2) are released from the call.

Scenario 2

Figures 33 and 34 show the state of the call before and after the multi-site transfer.

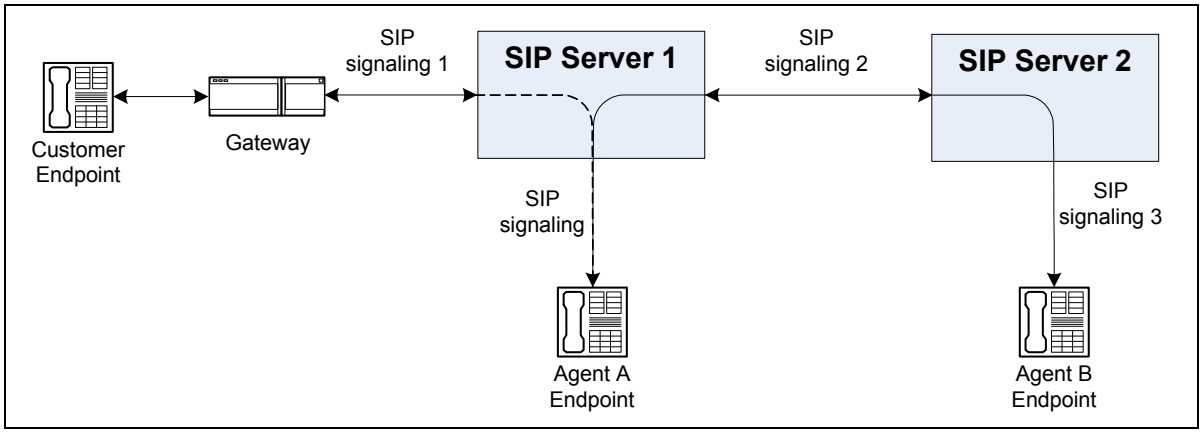

Figure 33: Call Before INVITE with Replaces Transfer

1. An inbound call is routed to Agent A at the SIP Server 1 site.
2. Agent A initiates a two-step transfer to Agent B at the SIP Server 2 site.

In this scenario, SIP Server uses a SIP `INVITE` request with the `Replaces` header to report call data for Agent B.

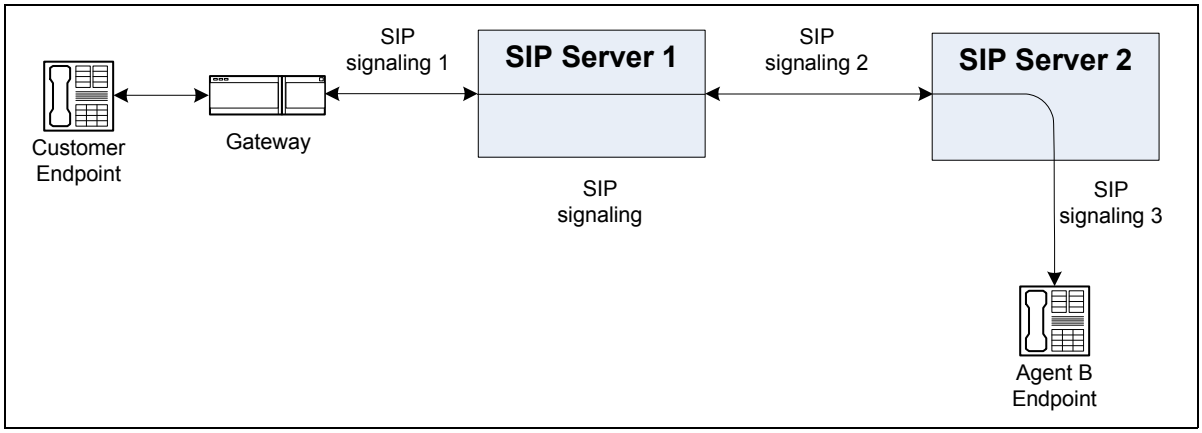

Figure 34: Call After INVITE with Replaces Transfer

In this case, the consultation call between the agents are merged on SIP Server 1, with user data propagated to the destination SIP Server (SIP Server 2). After the transfer is completed, SIP Server 1 remains in the signaling path—only the transferring agent (Agent A) is released from the call.

Scenario 3

Figures 35 and 36 show the state of the call before and after the multi-site transfer.

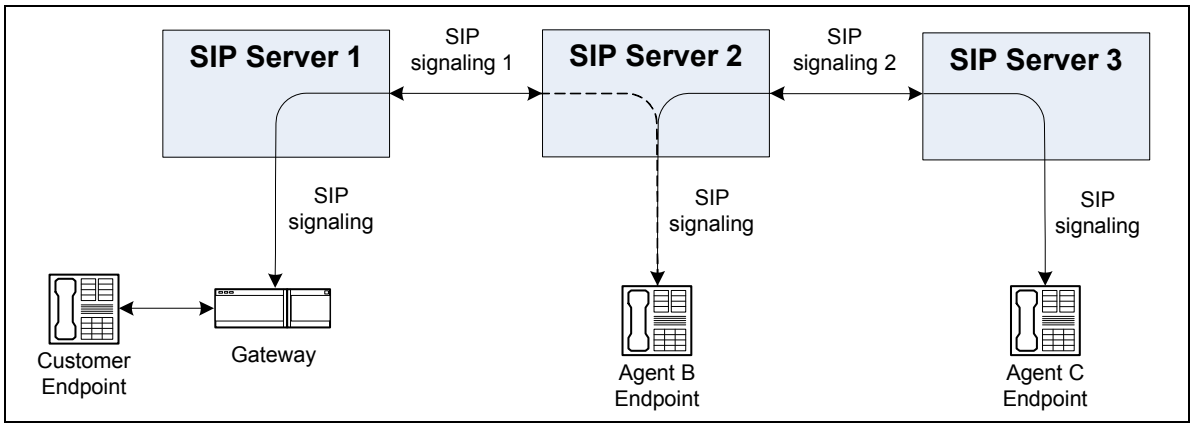

Figure 35: Call Before INVITE with Replaces Transfer

1. From a SIP Server 1 site, a call arrives to Agent B at the SIP Server 2 site.
2. Agent A initiates a two-step transfer to Agent C at the SIP Server 3 site.

In this scenario, SIP Server uses a SIP `INVITE` request with the `Replaces` header to report call data for Agent C.

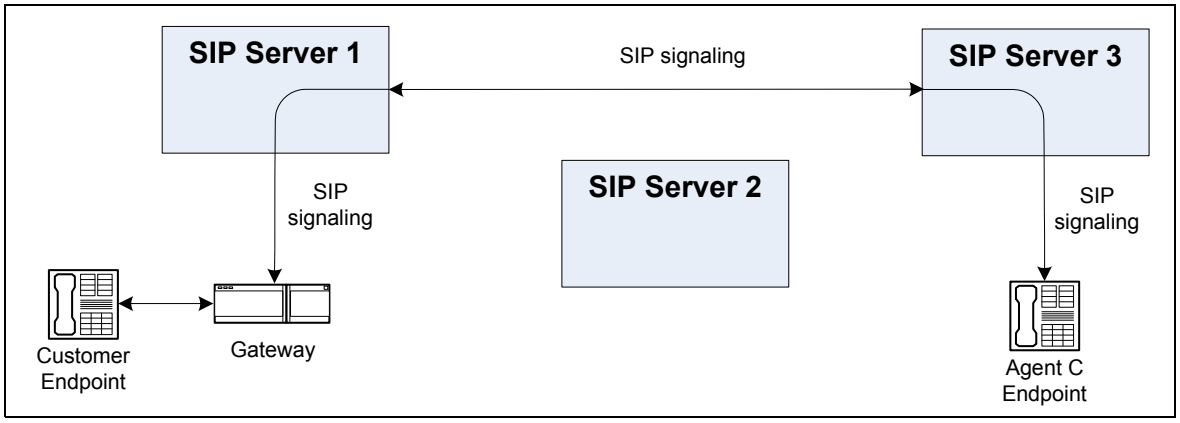

Figure 36: Call After INVITE with Replaces Transfer

After the transfer is completed, SIP Server 2 is removed from the signaling path. An `EventPartyChanged` message is generated for Agent C on SIP Server 3, based on information received in the `INVITE` request with the `Replaces` header.

ISCC Path Optimization

Introduced in SIP Server 8.1.101.33

For trunk optimization scenarios, SIP Server supports the ISCC path optimization feature, which excludes unnecessary hops in the user data propagation path and enhances reliability of data propagation. The ISCC path optimization is enabled by default and is controlled by the `path-optimization`

parameter. This feature does not change event distribution to clients, but rather modifies the path by which these events are conveyed.

See "ISCC Path Optimization" on page 695 for details.

Feature Configuration

Table 100 describes how to configure trunk optimization.

Table 100: Configuring Trunk Optimization

Objective	Related Procedures and Actions
1. Create `Trunk` DNs.	In each SIP Server configuration (origination and destination), in the corresponding SIP Switch, configure a DN of type `Trunk`. These `Trunk` DNs will be used for direct signaling between SIP Servers. For each `Trunk` DN, in the `TServer` section of the `Options` tab, configure the following options: • `refer-enabled`—Set this option to `false`. This will ensure that the `REFER` method for this `Trunk` DN will be used only for multi-site transfer optimization scenarios. • `oosp-transfer-enabled`—Set this option to `true`. • `sip-server-inter-trunk`—Set this option to `true`.
2. Configure the SIP Server Application.	In multi-site routing, to avoid reporting an access resource as `AttributeOtherDN` in related events, in the `extrouter` section of the SIP Server `Application` object, set the `cast-type` option to an ISCC direct transaction type (such as `direct-uui`).

Feature Limitation

This functionality requires direct signaling (no media gateways or session border controllers) between any two SIP Server instances, with no alteration of the SIP attributes (`CALL-ID`, `to` header, `from` header); as these are used for unique call context matching.

User to User Information (UUI)

SIP Server supports the SIP `User-to-User` header, as specified in the RFC draft "A Mechanism for Transporting User to User Call Control Information in SIP."

SIP Server does not generate UUI by itself. It only receives UUI and passes it through without modifications. UUI is considered correct if it complies with the following rules:

- UUI must contain the `encoding` parameter.
- The length of the `User-to-User` data must not be greater than the value specified by the `sip-max-uui-length` option.

SIP Server receives UUI in a SIP message or in a TRouteCall request. To extract UUI from the TRouteCall request, configure mapping the User-to-User header from the T-Library request to a SIP message. You can also configure mapping the User-to-User header from a SIP message to a T-Library request. See "Mapping SIP Headers and SDP Messages" on page 261 for more information.

The User-to-User header field can be included in INVITE and BYE messages.

SIP Server supports the uui tag in the Supported or Require header.

If SIP Server is configured to use the MSML service for all media services operations using Genesys Media Server, then SIP Server also passes UUI from the initial INVITE message to GVP.

SIP Server passes UUI in one of the following ways:

- In the User-to-User header—This method is applicable to INVITE, BYE, and REFER messages. To enable passing the User-to-User header from the REFER requests, this header must be configured in the sip-pass-refer-headers configuration option.
- In the URI parameter of the Refer-To header—SIP Server checks the Refer-To header of the REFER request for User-to-User data. If UUI is present, SIP Server includes it in the Refer-To header of the outgoing REFER or in the User-to-User header of the subsequent INVITE.
- In the URI parameter of the Contact header—SIP Server checks the Contact header of the 302 Moved temporary response for User-to-User data. If UUI is present, SIP includes it in the User-to-User header of the outgoing INVITE message.

Examples

1. If the REFER request contains the following information:

   ```
   Refer-To: <sip:1234@10.0.0.1:5060?User-to-
   User=1234567890abcdef%3Bencoding%3Dhex%3Bpurpose%3Disdnnetwork%3Bco
   ntent%3Disdn-uui>>
   ```

   SIP Server includes this in the INVITE request:

   ```
   INVITE sip:1234@10.0.0.1:5060... ·
   ...
   User-to-User:1234567890abcdef;encoding=hex;purpose=isdn-
   network;content=isdn-uui
   ```

2. If the 302 Moved Temporary response contains the following information:

   ```
   Contact: <sip:1234@10.0.0.1:5060?User-to-
   User=1234567890abcdef%3Bencoding%3Dhex%3Bpurpose%3Disdnnetwork%3Bco
   ntent%3Disdn-uui>
   ```

   SIP Server includes this in the INVITE request:

   ```
   INVITE sip:1234@10.0.0.1:5060 ·
   ...
   ```

```
User-to-User:
1234567890abcdef;encoding=hex;purpose=isdnnetwork;content=isdn-uui
```

Feature Limitations

- SIP Server does not support multiple `User-to-User` headers in one message.
- User-to-User information is not synchronized between the primary and backup SIP Servers in the HA pair. As a result, User-to-User information might be lost because of a SIP Server switchover.

Video Blocking

SIP Server provides the ability to block video streams from SDP offers during the call negotiation/establishment process so video will not be played when a call is established.

With this feature enabled:

- If an SDP offer contains both audio and video media types, only the audio stream is available for the call.
- If an SDP offer contains only a video media type and no other media types are available for negotiation, the call is rejected.

Example

The following is an example of the SDP body message containing both audio and video media types (highlighted):

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.dalycity.example.com
s=
c=IN IP4 host.dalycity.example.com
t=0 0
m=audio 49170 RTP/AVP 0 8 97
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
m=video 51372 RTP/AVP 31 32
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
```

When video blocking is enabled, SIP Server blocks (removes) the video media stream, as indicated in the following:

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.atlanta.example.com
```

```
s=
c=IN IP4 host.atlanta.example.com
t=0 0
m=audio 49170 RTP/AVP 0 8 97
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/800
```

Feature Configuration

The `sip-filter-media` configuration option enables this feature. The option can be set at both Application (`sip-filter-media`) and DN (`sip-filter-media`) levels. The option setting at the DN level takes precedence over the Application-level setting.

Feature Limitation

SDP media stream type filtering is not performed when SIP Server is placed out of the signaling path (OOSP).

Video Support

SIP Server supports the following scenarios related to Video Call functionality:

- Push Video
- Video Call on Hold
- Video Call Transfer
- Video Call Treatment
- Outbound Video Call
- Video Conference with active speaker detection (with Genesys Media Server only)

Push Video

Push Video functionality enables a person to play a video file to another call participant during a call. SIP Server can support video streams using Genesys Media Server and T-Library functions.

Start Video

To start playing a video file, SIP Server uses the `TSingleStepConference` function to push video from an agent to a customer. This function must contain the following attributes:

- `OtherDN`—Represents a video source. It is always defined as the `gcti::video` string.
- `Extensions`—Must contain the following key-value pairs:
  - `VideoFile`—A string that contains the name of the video file that will be played for the customer. If this key-value pair is not specified, the default video file will be played. The default video file is configured in the SIP Server `Application` object, using the `default-video-file` configuration option.
  - `AgentVideo`—A string that identifies the origin of the video stream played to the agent. The values are as follows:
    - `from-third-party`—The agent receives video from a third party—that is, the party that participated in the call before the operation started.
    - `to-third-party`—The agent receives the same video stream as played to the third party—that is, video from the file specified by the `VideoFile` parameter. (The `from-video-file` value can be used as an alias.)

    In either case, both the customer and agent hear each other and the audio that comes with the video file. The customer, the agent, and the audio source from the video file are three participants in the audio conference. When the pushed video ends, the customer and the agent continue a regular two-party conversation.

    If the `AgentVideo` key is not specified, or if it is empty, the `to-third-party` value will be used.

Stop Video

There are several ways to stop playing a video file:

- By deleting a party from a conference
- By releasing the `gcti::video` device
- When the video file ends

Deleting a Conference Call

SIP Server uses the `TDeleteFromConference` function to stop a video stream from a conference call. In this scenario, the `OtherDN` attribute is always defined as the `gcti::video` string.

Releasing the Device

SIP Server uses the `TReleaseCall` function to stop a video stream by releasing the `gcti::video` device. In this scenario, the `ThisDN` attribute is always defined as the `gcti::video` string.

When the File Ends

Genesys Media Server will end the SIP dialog for the `gcti::video` device when the video file ends, but it will not end the other SIP dialogs that belong to the conference.

Other Supported Scenarios

Video Call on Hold

The video call can be put on hold by using the `THoldCall` function. Genesys Media Server analyzes the endpoint capabilities submitted inside the endpoint's SDP message, and when supported, plays a video file.

Video Call Transfer

SIP Server supports a video call transfer by providing a regular offer/answer SDP message exchange between an endpoint during 1pcc operation.

Video Call Treatment

SIP Server supports a video call treatment in which a video file can be played to a customer when their call is on a Routing Point. The treatment prompts can be defined in a URS strategy that points to video files. For the video calls, Genesys Media Server plays both video and audio when a video prompt is specified.

Outbound Video Call

An agent can initiate an outbound video call using the `TMakeCall` function. In the SIP Server configuration, the `refer-enabled` option must be set to `true`, or the `make-call-rfc3725-flow` option must be set to `1`. The `INVITE` message to an external destination will contain SDP information with the agent's endpoint video capabilities.

When the agent initiates an outbound video call, and a recipient accepts it, the video file starts playing. If the recipient's endpoint does not have video capabilities or refuses the video connection, only an audio connection—without the video—is established.

Feature Configuration

Table 101 describes how to configure video support.

Table 101: Configuring Video Support

Objective	Related Procedures and Actions
1. Install a PC video camera.	Follow the instructions in the video camera documentation.
2. Configure a SIP endpoint to support video functionality.	Follow the instructions specific to the SIP endpoint you are using. Complete the wizard steps, and select the installed video camera on the corresponding wizard page.
3. Configure a SIP Server Application.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, specify the `default-video-file` configuration option. This option contains the name of the video file that is played to the caller if a single-step conference to the `gcti::video` device does not contain a `VideoFile` key in the `Extensions` attribute.
4. Configure a `gcti::video` device.	For Push Video: Under a configured `Switch` object > `DNs` folder, create a new DN object by setting the following properties: • `Number`—Enter `gcti::video`. • `Type`—Select `Trunk`.
5. Configure a video service.	For Push Video: 1. Under a configured `Switch` object > `DNs` folder, create a new DN object by setting the following properties: ◆ `Number`—Enter the DN name. This name is currently not used for any messaging, but it must still be unique. ◆ `Type`—Select `Voice over IP Service`. 2. In the `TServer` section, configure the following options: ◆ `contact`—Specify the value using the Genesys Media Server application settings in the following format: `IP address:SIP port` ◆ `request-uri`—Specify the value to be used as a template for the source of the video stream and as the value of the `Request-URI` parameter in the `INVITE` message: `annc@<sm_or_mediaserver_hostport>;play=<file>`;repeat=<number>) ◆ `service-type`—Set this option to `video`.

Working with Multiple Devices

When multiple devices within a deployment are able to provide a particular service—for example, `Trunk` DNs representing several possible gateways for placing an outbound call, or `Voice over IP Service` DNs at different locations configured to provide music treatments for an inbound call—SIP Server uses a selection algorithm to choose the most appropriate device to provide the service.

Device Selection Procedure

SIP Server makes its selection from the pool of compatible devices in the following order of descending priority:

1. "Prefix match"
2. "Current availability"
3. "Partition-id parameter"
4. "Geo-location attribute"
5. "Cpd-capability parameter"
6. "Priority attribute"
7. "Percentage of used capacity"
8. "Round-robin"

Prefix match

SIP Server first narrows the pool of devices by matching the prefix option as defined on the DNs. SIP Server chooses the longest match possible. For example, if SIP Server finds some devices with a prefix of `90` and other devices with a prefix of `900`, SIP Server narrows the pool to those devices that have the longer prefix of `900`.

Current availability

SIP Server further narrows the pool to those devices that are currently in service. For example, if the maximum capacity for the device has been reached, or if the device has been marked as temporarily unavailable, SIP Server does not consider the device.

Note: If at this point in the selection process SIP Server finds no currently available device, it cannot go back and select a shorter prefix match. For example, if the devices that have a prefix of `900` become unavailable, SIP Server cannot then consider devices that have a prefix of `90`. Instead, it returns an error.

Partition-id parameter

SIP Server uses the `partition-id` parameter to select a service based on the partition to which the call belongs.

Geo-location attribute

Services that have the same geo-location as the target are preferred over all others. If no matching geo-location is found, or if no geo-location is configured

on the target, no differentiation is made from among the pool of available resources. For more information, see "Selection Based on Geo-Location" on page 389.

Cpd-capability parameter

When making an outbound predictive call, SIP Server narrows the pool of available outbound gateways to those configured for `cpd-capability`. If no `Trunk` DN with `cpd-capability` is found, SIP Server will try to perform CPD on GVP instead, using the media server capability of the GVP Media Control Platform (MCP).

Priority attribute

Services that have higher priority (the lower the configured priority number, the higher the priority) are always preferred over services of lower priority. If the higher priority device becomes unavailable because its maximum capacity is reached, only in this case will a lower priority service be considered.

Percentage of used capacity

According to this rule, SIP Server selects the service that has the least amount of used capacity. For example, SIP Server will select a service that has used only 10% of its maximum configured capacity over a service with 50% used capacity. SIP Server compares services in pairs—if either service in the pair is not configured for capacity, SIP Server uses the round-robin method instead.

Round-robin

SIP Server selects the service that has the gone the longest length of time without being selected for a call. SIP Server considers only the time of selection. It does not consider either the reasons for previous selections, or the call end times.

Note: In cases where SIP Server selects a `Trunk` DN that represents a gateway, but all lines for that gateway are busy, occupied, or otherwise out-of-service, SIP Server will silently try to reach the destination by using another gateway `Trunk` DN, if one is available. SIP Server remembers the failed `Trunk` DN and, if more than one `Trunk` DN is configured, avoids trying it again for the duration of the call.

Feature Configuration

Table 102 describes how to define the device-selection process—which priorities in the algorithm will be considered when selecting a device.

Table 102: Defining the Device-Selection Process

Objective	Related Procedures and Actions
1. Selection based on prefix match.	*(Optional) If disabled, SIP Server does not consider the prefix when selecting a device.* This step in the procedure applies to `Trunk` DN selection only. • Gateway—To include the prefix match when selecting a gateway, set the `prefix` option on the `Trunk` DN for the gateway to the initial digits of the dialed number that will map to this gateway.

Table 102: Defining the Device-Selection Process (Continued)

Objective	Related Procedures and Actions
2. Selection based on availability.	No configuration required. SIP Server always checks for the availability of compatible devices. If no available device is found, SIP Server returns an error.
3. Selection based on partition.	SIP Server matches originating device with available resource devices using the default SIP Server partition. To change the default partition, do the following: 1. To define the default partition for all DNs, set the `partition-id` option in the SIP Server `Application` object. All DNs on this switch will be considered to belong to this partition, unless otherwise defined. 2. To assign a particular DN to a different partition, set the `partition-id` option in the DN object to the partition to which it belongs.
4. Selection based on geo-location.	*(Optional) If disabled, SIP Server does not consider the geo-location when selecting a device.* 1. Configure all applicable `Voice over IP Service` and `Trunk` DNs at a particular premise with the same `geo-location` value. For a list of DN types that support geo-location, see Table 113, "DN Configuration Objects," on page 583. 2. In the SIP Server `Application` object, set the `find-trunk-by-location` option to `true`. 3. (Optional) Configure a routing or treatment block in the routing strategy to include the geo-location `Extensions` attribute with the same value as the DNs at the same premise. **Note:** This method takes precedence over `geo-location` configured at the DN level. For more information, see Procedure: Setting the geo-location for a call, on page 390.
5. Selection based on cpd-capability.	To identify a particular gateway or media server as CPD-capable, configure as follows: • media gateway—On the `Trunk` DN for the gateway, set the `cpd-capability` option to one of the following supported gateway types: ◆ `audiocodes` ◆ `paraxip` • GVP Media Server—On the `Trunk Group` DN for the GVP media server, set `cpd-capability` to the following value: ◆ `mediaserver`

Table 102: Defining the Device-Selection Process (Continued)

Objective	Related Procedures and Actions
6. Selection based on priority setting.	*(Optional) If disabled, SIP Server does not consider the priority setting when selecting a device.* To include the priority setting when selecting a device, configure the applicable DN as follows: • `priority`—In the `TServer` section, set this option to the desired priority level.
7. Selection based on used capacity.	*(Optional) If disabled, SIP Server does not consider the capacity setting when selecting a device.* This step in the procedure applies to `Trunk` DN selection only. To include the capacity setting when selecting a Trunk, configure all applicable DNs as follows: • `capacity`—On the `Trunk` DN, set the `capacity` option to the usage capacity that is available for this device. • `capacity-group`—You can use this option to assign a defined capacity to multiple `Trunk` DNs.

Selection Based on Geo-Location

Participants within VoIP conversations are divided into two groups:

- Internal parties that represent agents or supervisor SIP endpoints. These communicate directly with each other by signaling directly with SIP Server and by RTP streams. In the Configuration Layer, they are configured as DNs of type `Extension`.
- External parties that represent customers or agents at remote sites. These communicate with contact center devices using a media gateway or other proxy services. External parties do not have a direct representation at the Configuration Layer; instead, they must be represented as a `Trunk` DN or a "media gateway" that is associated with a SIP Server `Switch` object.

You need to choose a gateway at the same premise where the agent SIP endpoint is located to minimize network load for RTP traffic and for VoIP media services, such as music on hold, central mixing conferencing, or voice recording. Devices in the same premise must be configured with the same value of the `geo-location` option.

- An internal party has the same option value as the corresponding `Extension` DN object when a call is established.
- An external party has the same option value as the corresponding `Trunk` DN object when a call is established.
- A media server has the same option value as the corresponding `Voice over IP Service` DN object when a call is established.

To include the `geo-location` attribute in the procedure that SIP Server uses to select a gateway or trunk for the outbound call, you must set the `find-trunk-by-location` option to `true`.

To determine the gateway for an external party of an inbound call, the IP address from the `Via` header of the incoming `INVITE` message must match the host address of the `contact` option of the `Trunk` DN. If a match is successful, the `geo-location` label for the matched trunk will be used as the `geo-location` label for the external party. In order for this match to work, the contact of the corresponding trunk must be the same, because it is expected to be inside the `Via` header of the incoming `INVITE` message (most likely a decimal IP address). If SIP Server cannot find a `Trunk` DN with a `contact` option to match the identifier provided in the `Via` header, then SIP Server rejects the call and issues a `404 Not Found` message.

For other services, such as `music`, `treatment`, or `recorder` SIP Server searches for the same `geo-location` label as the party requesting such service.

Procedure: Setting the geo-location for a call

Purpose: To configure the DNs or routing strategy that are required to apply the geo-location for the call.

For geo-location matching to work, both parties in the SIP call must be configured with the same geo-location label. The semantics for the label are arbitrary, but they must be consistent across the premise.

Outgoing Calls

For outgoing calls, the agent DN that is making the call must be configured with a geo-location label. When this agent places a call, SIP Server searches available `Trunk` DNs (according to the selection algorithm) for any that are configured with the matching `geo-location` label.

Incoming Calls

For incoming calls, the geo-location label for a call can be applied either through the `Trunk` DN where the call first arrives, or through a routing strategy that is loaded on the `Routing Point` DN where the call first arrives. Geo-location that is assigned from the routing strategy takes precedence.

Consultation Calls

For consultation calls, SIP Server looks for an external destination based on the geo-location value as set on the originating DN for the consultation call. If the consultation scenario involves a Routing Point, then SIP Server checks for geo-location according to the following priority (listed in order of precedence from highest to lowest):

- The `Extensions` parameter in the `RequestRouteCall`.
- The `geo-location` option as set on the `Routing Point` DN.
- The `geo-location` option as set on the originating DN for the consultation call.

Internal Calls

For internal calls, in HoldCall scenarios, the geo-location label for a call is assigned through the party to which the Music-on-Hold (MOH) is played. If geo-location is not configured in that party, then SIP Server chooses the geo-location configured in the originating party. If the scenario involves a Routing Point, then SIP Server checks for geo-location according to the following priority (listed in order of precedence from highest to lowest):

- The `geo-location` option as set on the DN to which MOH is played.
- The `Extensions` parameter in the `RequestRouteCall`.
- The `geo-location` option as set on the `Routing Point` DN.
- The `geo-location` option as set on the originating DN.

Voice over IP Service

For call treatments, you can configure the `Voice over IP Service` DN that provides the service (music on hold, for example) with a geo-location label that matches the label for a gateway at the same premise. This allows SIP Server to choose a server at the same location as the gateway, minimizing RTP traffic. For a list of DN types that support geo-location, see Table 113, "DN Configuration Objects," on page 583.

Start of procedure

1. **For outbound calls**—Configure the agent DNs and gateway DNs at the same premise with identical `geo-location` labels.
   a. On the `Extension` DN for the agent, on the `Options` tab, in the `TServer` section, set the `geo-location` to the identifier that you want to use for this premise.
   b. On the `Trunk` DN for a gateway at the same premise, on the `Options` tab, in the `TServer` section, set the `geo-location` to the same identifier as you did for the agent DNs.
2. **For inbound calls**—Configure the `Voice over IP Service` DNs for any SIP services that you expect to provide for calls arriving at the same premise as the server that provides the service.
   a. On the `Voice over IP Service` DN, on the `Options` tab, in the `TServer` section, set the `geo-location` to the identifier that you use for this premise.
   b. If the `Trunk` DN for the gateway at this premise is not already configured for it, set the `geo-location` to the identifier that you use for this premise.

      OR

      In the routing strategy, configure the `Function` object to attach the Extensions attribute to the call: set the key name to `geo-location` and set the value to the identifier you use for the desired premise.

End of procedure

Geo-Location Support by GVP

Introduced in SIP Server 8.1.101.62

Genesys software applies geo-location to multiple configuration objects. This enables Resource Manager to select the closest Media Server to the caller or agent, minimizing WAN traffic and telecom costs. SIP Server passes geo-location data to Resource Manager when Genesys Media Server is configured as:

- a Trunk DN
- a Trunk Group DN
- a Voice Treatment Port (VTP) DN
- an MSML Voice over IP Service (VOIP) DN
- a Voicemail VOIP DN

Table 103 matches integration modes with DN types.

Table 103: SIP Server-Genesys Media Server Integration Modes: Required DN Types

Integration Mode	Configure Genesys Media Server as this DN Type:
GVP Inbound mode	Trunk DN
Outbound Integration mode	Trunk Group DN and VTP DN
Voicemail Integration mode	VOIP Service DN (`service-type=voicemail`)
Media Server mode	VOIP Service DN (`service-type=msml`)

SIP Server puts the geo-location value of a call into the `X-Genesys-geo-location` header of the `INVITE` that it sends to Resource Manager, but only under these conditions:

- if the call's geo-location is defined as a call property.

  OR

- if the call's geo-location is passed as an extension in a T-Library request (such as `TApplyTreatment` and `TRouteCall`).

If neither is true, then SIP Server does not pass the geo-location to Resource Manager.

For example: some countries require that an incoming call's geo-location be passed as a call property, and other countries do not require it. Now you can configure Media Server to account for that.

For more information about setting geo-location for a call, see "Selection Based on Geo-Location" on page 389.

Deployment Examples

GVP Inbound mode

- Single Media Server with MCP farms located at a different geo-location
- Multiple Media Servers each with MCP farms located in different geo-locations

Outbound Integration mode

- Single Media Server handling multiple geo-location farms
- Multiple Media Servers handling multiple geo-location farms

CTI through IVR Server (IVR-centric)

- Single Media Server handling multiple media farms
- Multiple Media Servers located in multiple locations handling multiple MCP farms

Voicemail Integration mode

- Single Media Server and multiple MCP farms
- Multiple Media Servers located at different locations handling multiple MCP farms

Strict Geo-location matching scenario

See “Geo-Location for MSML-Based Services: Strict Matching” on page 393.

Geo-Location for MSML-Based Services: Strict Matching

Introduced in SIP Server 8.1.101.62

SIP Server supports strict geo-location matching for MSML-based services by ensuring that a call with a particular geo-location is served only by an MSML service within the same geo-location or by an MSML service within the alternate location (if configured). If a correctly geo-located MSML service is unavailable, SIP Server does not provide the required service.

SIP Headers

To prevent GVP from using a wrongly located MCP farm, SIP Server adds (in addition to the `X-Genesys-geo-location` header) the `X-Genesys-strict-location` header with a value of `enforce` to the `INVITE` that it sends to GVP.

Table 104: Geo-location: Strict Matching SIP Headers

Header Name	Header Value	Meaning
`X-Genesys-geo-location`	It contains the call geo-location label if an MSML service matching that label is available. It contains the overflow-location label if the initial MSML service is not available.	Instructs the Resource Manager to choose the MCP that serves a particular location.
`X-Genesys-strict-location`	`enforce`	Informs the Resource Manager that it must reject `INVITE` if there is no MCP available that serves a particular the geo-location.

Alternate Geo-Location

The alternate geo-location defined by the `overflow-location-map` option allows you to pair an alternate MSML-based service with a geo-location label. The alternate (overflow) location will be tried if the primary geo-location is not available or fails. In addition, the `overflow-location` key can be provided in `AttributeExtensions` of the `TRouteCall` and `TApplyTreatment` client requests. The value of the `overflow-location` key in `AttributeExtensions` takes precedence over the `overflow-location-map` option value. If present in the request, the overflow-location key enables a strict MSML-based service location for the call even if it is disabled at the Application level. If the `overflow-location` key is empty, SIP Server removes the previously assigned overflow-location (if set at an Application level) and enables MSML strict geo-location matching.

If SIP Server finds the corresponding service and sends an `INVITE` to that service but does not receive a positive response, SIP Server might retry an `INVITE` attempt once more—but only within the service that has the same geo-location or geo-location equal to the value of the `overflow-location` key.

Failure Alarms

SIP Server can generate an MSML geo-location failure alarm whenever an attempt to provide an MSML service fails because of the geo-location strict matching. The option `msml-location-alarm-timeout` specifies how often SIP Server sends that alarm (`52052` code). An alarm message contains a list of failed geo-locations along with a number of failures occurred within the timeout. There is no message to reset the alarm. It is supposed to be reset by

the Management Layer timeout (should be greater than the timeout defined by the option above) when SIP Server stops detecting new MSML geo-location failures.

Feature Configuration

Table 105 describes how to enable geo-location with strict matching.

Table 105: Configuring Geo-Location with Strict Matching

Objective	Related Procedures and Actions
Configure the SIP Server Application.	In the `TServer` section of the SIP Server Application, configure the following options: • `enable-strict-location-match`—Set this option to msml or true. • (Optional) `overflow-location-map`—Set this option to `geo-location label=overflow-location` label. • (Optional) `msml-location-alarm-timeout` • `msml-support`—Set this option to `true`.
Configure MSML DN(s).	See "Configuring Genesys Media Server" on page 92.
Configure a Trunk DN.	1. Create a DN of type `Trunk`. 2. In the `TServer` section, configure the following option: ◦ `geo-location`—Set this option to the same geo-location value as any of the MSML DNs.
(Optional) Configure a treatment block in the routing strategy.	Include the `geo-location` extension key with the same value as any of the MSML DNs. **Note:** This method takes precedence over geo-location configured at the DN level.

Feature Limitations

- The feature works only for MSML-based devices (no NETANN support).
- If an MSML service is selected for a device with `contact=::msml` on a corresponding DN (`Trunk, Voice Treatment Port, Trunk Group,` or Voicemail DN), the feature works properly only in the Active-Active RM pair deployment.

- If an MSML service is selected for a device with `contact=::msml` on a corresponding DN (`Trunk`, `Voice Treatment Port`, `Trunk Group`, or Voicemail DN), SIP Server does not try the alternate (overflow) location if an initial `INVITE` to the primary geo-location fails. This limitation does not apply to the selection of the destination for the initial `INVITE`.

Genesys Voice Platform Integration

For detailed information about SIP Server integration with the Genesys Voice Platform (GVP), see the following documents:

- The *Genesys 7.5 GVP–SIP Server Integration Guide*—This guide provides an overview of the GVP–SIP Server integration in its various modes—In-Front, Behind, and Standalone—as well as the relevant procedures for completing the integration. This document applies to the 7.5 release of SIP Server, and the 7.5 and 7.6 releases of GVP.
- The *Genesys 8.1 Voice Platform Solution Integration Guide*—This guide provides an overview of the Voice Platform Solution (VPS), with the aim of integrating the various components that make up the solution. This document applies to the 7.6 release of SIP Server and the 8.1 release of GVP.

Active-Active Resource Managers

Resource Managers (RM) can be deployed in an Active-Active High-Availability cluster, in which both RM instances run together as the active instance, each with a unique IP address. The active pair synchronizes active session information, so that both instances can correctly route incoming requests.

Figure 37 shows a sample deployment in which SIP Server performs load balancing between two RM instances using a round-robin algorithm.

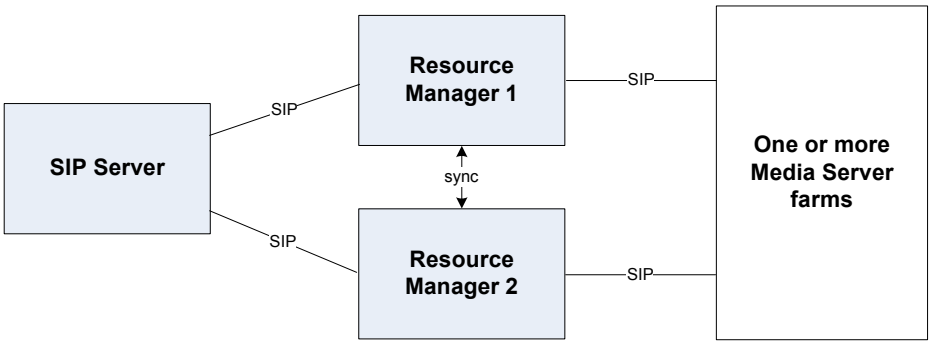

Figure 37: SIP Server/Active-Active RM Pair Deployment

Multiple Resource Manager Pairs

SIP Server can also work with multiple Active-Active RM pairs, with each RM pair managing separate Media Server farms. If these RM pair and Media Server/MCP farms are deployed across multiple locations, SIP Server uses geo-location to select among the locations.

How It Works

In Active-Active RM pair deployments, SIP Server performs load balancing between two RM instances. To check current RM availability, SIP Server sends periodic `OPTIONS` messages to each RM instance.

In the configuration, each Active-Active RM pair is represented by a single DN of type `Voice over IP Service` with `service-type=msml` (or multiple DNs if there are more RM pairs). If a particular RM instance does not respond to the `OPTIONS` message, SIP Server marks that instance as out-of-service. If both instances in the RM pair fail to respond, then SIP Server marks the MSML DN as out-of-service.

When SIP Server needs to send a request to Media Sever (for example, to start a continuous playback treatment), SIP Server selects an active RM instance. If for some reason this RM instance fails, SIP Server sends any further mid-dialog requests to the alternate RM instance. Active-active synchronization between the RM pair ensures that all call details are shared by both instances, providing uninterrupted service.

Building the Request-URI

SIP Server connects to the RM pair using multiple IP addresses or Fully-Qualified Domain Names (FQDN).

SIP Server builds the Request-URI for Active-Active RM deployments differently, depending on the configuration:

- If the contact of the RM instance is configured as an FQDN in the `contact` option, SIP Server sends this FQDN in the Request-URI of the new `INVITE`.
- If the IP address of the RM instance is configured in the `contact-list` option, SIP Server builds the Request-URI of the outgoing `INVITE` with the selected transport IP address, port, and protocol information.

OPTIONS Message Optimization

SIP Server integrates with GVP using different modes, where RM is configured as a different DN for each mode. SIP Server can work with GVP in the following modes:

- Media server mode:
  - SIP Server uses GVP to play treatments, MOH, and so on.
  - RM is configured as an MSML DN of type `Voice over IP Service`.

- GVP Inbound mode:
  - GVP is used as self service and SIP Server forwards requests to GVP.
  - RM is configured as a `Trunk` DN.
- Outbound mode:
  - SIP Server uses GVP for CPD and to play treatments.
  - RM is configured as a `Trunk Group` DN.

If SIP Server were to send `OPTIONS` messages to each of these DNs, this would increase the load on the network. To avoid unnecessary network load, you configure the contact of the RM in the MSML DN. SIP Server sends `OPTIONS` messages to this DN to monitor availability. It also creates an active transport list for this DN.

For all other DNs that need to use this RM contact, configure the contact using the special value `::msml`. When SIP Server receives this special contact, it selects the MSML DN and gets the contact information from there. The `::msml` value is mandatory only in Active-Active RM pair integration scenarios and must not be used in other integrations, such as Active-Standby RM integration.

Note: Genesys recommends using the TCP protocol for all transport between SIP Server and GVP. This is mandatory if TLib-to-SIP user data mapping is configured between SIP Server and GVP.

Feature Configuration

Active-Active RM pair support can be configured using the following methods:

- FQDN-based: The RM contact is configured using the FQDN that resolves to two SRV records; each SRV record resolves to the IP address/port for one RM in the Active-Active RM pair.
- IP-based: The RM contact is configured using IP address/port information.

Table 106 describes the configuration required for both FQDN- and IP-based Active-Active RM integration scenarios.

Table 106: Integrating with Active-Active RM Pair

Objective	Related Procedures and Actions
1. Configure the SIP Server Application.	In the SIP Server `Application` object > `Application Options` tab > `TServer` section, configure the following options: • `msml-support`—Set this option to `true`. • `sip-enable-rfc3263`—For FQDN-based method, set this option to `true`. • `sip-enable-gdns`—For FQDN-based method, set this option to `true`. • `sip-invite-treatment-timeout`—A minimum value of this option should be equal to the sum of the values of the `oos-check` and `oos-force` options (set at the DN level) multiplied by two. SIP Server must try to pass a media service request through both RM instances in the Active-Active RM pair before the `INVITE` transaction expires.
2. Configure Resource Manager Applications.	Configure two instances of Resource Managers to work in Active-Active mode. In the first RM `Application` of the pair (Node 1) > `Application Options` tab > `cluster` section, configure the following options: • `ha-mode`—Set this option to `active-active`. • `mymemberid`—Set this option to `1`. • `member.1`—Set this option to the IP address of this RM, followed by the cluster-communication-port number, and separated by a colon (:). • `member.2`—Set this option to the IP address of the other RM node (Node 2), followed by the cluster-communication-port number, and separated by a colon. • `hotstandby`—Set this option to `TRUE`. • `members`—Ensure this option is set to `1 2` (separated by a space).

Table 106: Integrating with Active-Active RM Pair (Continued)

Objective	Related Procedures and Actions
(continued)	In the second RM `Application` of the pair (Node 2) > `Application Options` tab > `cluster` section, configure the following options: • `ha-mode`—Set this option to `active-active`. • `mymemberid`—Set this option to `2`. • `member.1`—Set this option to the IP address of the other RM node (Node 1), followed by the cluster-communication-port number, and separated by a colon. • `member.2`—Set this option to the IP address of this RM, followed by the cluster-communication-port number, and separated by a colon. • `hotstandby`—Set this option to `TRUE`. • `members`—Ensure this option is set to `1 2` (separated by a space).
3. Configure the MCP Application.	In the MCP `Application` > `Application Options` tab > `sip` section, configure the following option: • `transport.staticroutelist`—Set this option to the IP addresses of the RM pair, separated by a comma, since they are within the same route group. Do not specify a port number.
4. Create a DN to represent the Active-Active RM pair.	Create a `Voice over IP Service` DN to handle the connection between SIP Server and the RM instances: 1. Set the `service-type` for this DN to `msml`. 2. Configure the contact information as follows: For FQDN-based configuration: ◦ `contact`—Enter the FQDN for the RM instances. See "DNS Name Resolution" on page 217 for details. For IP-based configuration: ◦ `contact-list`—Enter a comma-separated list of IP addresses and ports of RM instances. 3. Enable Active-Out-of-Service Detection for this DN: ◦ Configure the `oos-check` and `oos-force` options. See "Active Out-of-Service Detection" on page 240 for details. 4. Configure geo-location for this DN: ◦ `geo-location`—Enter the location for the RM instances. See "Geo-location attribute" on page 386 for details. 5. Configure subscription for this DN: ◦ `subscription-id`—Set this option to the name of the Tenant to which this DN belongs. **For SIP Proxy deployments:** In addition to the above, add the following configuration option for FQDN-based configuration: • `replace-uri-contact`—Set this option to `true`.

Table 106: Integrating with Active-Active RM Pair (Continued)

Objective	Related Procedures and Actions
5. Enable GVP Inbound.	To enable inbound GVP where SIP Server is located in front of GVP and forwards inbound calls to GVP, complete the following steps: 1. Create a `Trunk` DN to represent GVP. 2. Configure the following options: • `contact`—Set this option to `::msml`. • `prefix`—Enter the initial digits to match the DNIS number. • `sip-busy-type`—Set this option to `2`. • `geo-location`— Enter the location for the RM instances. **For SIP Proxy deployments:** In addition to the above, add the following configuration option for FQDN-based configuration: • `replace-uri-contact`—Set this option to `true`.
6. Enable GVP Outbound.	To enable outbound GVP where GVP is used as a media server to provide services used for outbound, complete the following steps: 1. Create a `Trunk Group` DN to handle outbound calls through GVP. 2. Configure the following options: • `contact`—Set this option to `::msml`. • `subscription-id`—Enter the name of the tenant. (**Note:** Starting with GVP 8.1.2, multi-tenancy is supported. Set the name of this Trunk Group to the name of the tenant to which this DN applies.) • `make-call-rfc3725-flow`—Set this option to `1`. • `refer-enabled`—Set this option to `false`. • `ring-tone-on-make-call`—Set this option to `false`. • `request-uri`—Enter a URI in the following format: `sip:msml@<RMHost>:<RMPort>;media-service=cpd;gvp-tenant-id=[<tenant name>]` 3. If the outbound integration includes Voice Treatment Ports (VTPs), you must also configure each VTP DN with the following option: • `contact`—Set this option to `::msml`. **For SIP Proxy deployments:** In addition to the above, add the following configuration option for FQDN-based configuration: • `replace-uri-contact`—Set this option to `true`.
7. For deployments with SIP Proxy, configure SIP Proxy.	See "SIP Proxy Support" on page 354 for configuration details.

Active-Active RM Integration Limitations

- In both modes (IP-based and FQDN-based), SIP Server does not support configuring priority and weight for each IP address. All entries are considered of equal priority and weight.
- Genesys recommends that you set the `sip-invite-treatment-timeout` option to a value equal to twice the sum of the values of the `oos-check` and `oos-force` options set at the DN level. This ensures that SIP Server tries both Resource Manager instances before deciding that a particular site is not responding.
- In deployments where GVP is configured as a trunk containing `contact=::msml,` when GVP initiates an outbound call to SIP Server, SIP Server cannot match the GVP trunk as an origination trunk.

  To apply the `sip-busy-type` option for outbound calls initiated from GVP, use the following workaround:

  Configure two trunks—one with the RM 1 `contact` and another with the RM 2 `contact`. SIP Server selects one of these trunks as an origination trunk and applies those options for GVP-initiated outbound calls.

Genesys Media Server

For information about SIP Server integration with the Genesys Media Server (GVP for media services only), see "Configuring Genesys Media Server" on page 92, as well as the *Genesys Media Server Deployment Guide*.

GVP Integration Limitation

When SIP Server is working in High-Availability (HA) mode, multiple switchovers between backup and primary SIP Server instances can interfere with SIP Server correctly sending `SUBSCRIBE` messages to Genesys Voice Platform. To ensure that SIP Server is able to send these messages, you must enable Active Out-of-Service Detection for the `Trunk Group` DN that represents GVP, as follows:

- On the `Trunk Group` DN that represents GVP, enable the options `oos-check` and `oos-force`.

PlayApplication Treatment Parameters

The `APP_ID` treatment parameter of the `PlayApplication` treatment is used in pre-8.0 GVP integrations and is supported only in the Voice over IP Service DN with `service-type=application`.

The `APP_ID` parameter is not supported in the MSML Voice over IP Service DN (`service-type=msml`) and the `APP_URI` parameter must be used instead.

Passing Extended Recording Metadata to GVP

Introduced in SIP Server 8.1.103.92

SIP Server in standalone mode supports passing the additional GVP parameters (which have Agent Assist supporting key-value pairs (KVPs) and Streaming KVPs) from `AttributeExtensions` of `TRouteCall` to MCP in the recording `INFO` messages, under existing recording metadata.

The initial characters of recording GVP parameters must match the prefix specified in the `record-metadata-prefix` configuration option. Those GVP parameters are added only if the following conditions are met:

- The KVP's prefix matches the `record-metadata-prefix` option value.
- The total number of matching KVPs does not exceed 5. If exceeded, no KVPs are attached to the call data (metadata storage) and no additional GVP parameters are passed to MCP in the recording `INFO` message.
- Call recording is enabled.

GENESYS

Chapter

6 T-Library Functionality Support

This chapter describes the T-Library functionality that SIP Server supports. It contains the following sections:

Using T-Library Functions

Table 107 presents the T-Library functionality supported in SIP Server. The table entries use these notations:

N—Not supported

Y—Supported

E—Event only is supported

I—Supported, but reserved for Genesys Engineering

In Table 107, when a set of events is sent in response to a single request, the events are listed in an arbitrary order. An asterisk (*) indicates the event that contains the same `Reference ID` as the request. For more information, refer to the *Genesys Events and Models Reference Manual* and the *Platform SDK 8.x .NET (or Java) API Reference* for complete information on the T-Server events, call models, and requests.

Table 107 reflects only the switch functionality that Genesys software supports and might not include the complete set of events that the switch offers.

Certain requests listed in Table 107 are reserved for Genesys Engineering and are listed here merely for completeness of information.

Table 107: Supported Functionality

Feature Request	Request Subtype	Corresponding Event(s)	Supported
General Requests			
TOpenServer		EventServerConnected	Y
TOpenServerEx		EventServerConnected	Y
TCloseServer		EventServerDisconnected	Y
TSetInputMask		EventACK	Y
TDispatch		Not Applicable	Y
TScanServer		Not Applicable	Y
TScanServerEx		Not Applicable	Y
Registration Requests			
TRegisterAddress[a]		EventRegistered	Y
TUnregisterAddress[a]		EventUnregistered	Y
Call-Handling Requests			
TMakeCall[b]	Regular	EventDialing	Y
	DirectAgent		N
	SupervisorAssist		N
	Priority		N
	DirectPriority		N
TAnswerCall		EventEstablished	Y[c]
TReleaseCall		EventReleased	Y
TClearCall		EventReleased	Y
THoldCall		EventHeld	Y
TRetrieveCall		EventRetrieved	Y
TRedirectCall[d]		EventReleased	Y
TMakePredictiveCall[e]		EventDialing* EventQueued	Y

Table 107: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
Transfer/Conference Requests			
TInitiateTransfer[b]		EventHeld EventDialing*	Y
TCompleteTransfer		EventReleased* EventPartyChanged	Y
TInitiateConference[b]		EventHeld EventDialing*	Y
TCompleteConference		EventReleased* EventRetrieved EventPartyChanged EventPartyAdded	Y
TDeleteFromConference		EventPartyDeleted* EventReleased	Y
TReconnectCall		EventReleased EventRetrieved*	Y
TAlternateCall		EventHeld* EventRetrieved	Y
TMergeCalls	ForTransfer	EventHeld EventReleased* EventRetrieved EventPartyChanged	N
	ForConference	EventHeld EventReleased* EventRetrieved EventPartyChanged EventPartyAdded	N
TMuteTransfer[b,f]		EventHeld EventDialing* EventReleased EventPartyChanged	Y
TSingleStepTransfer[b]		EventReleased* EventPartyChanged	Y
TSingleStepConference		EventRinging* EventEstablished	Y

Table 107: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
Call-Routing Requests			
TRouteCall[b]	Unknown	EventRouteUsed	Y
	Default		Y
	Label		N
	OverwriteDNIS		N
	DDD		N
	IDDD		N
	Direct		N
	Reject		Y
	Announcement		N
	PostFeature		N
	DirectAgent		N
	Priority		N
	DirectPriority		N
	AgentID		N
	CallDisconnect		N
Call-Treatment Requests			
TApplyTreatment	Unknown	(EventTreatmentApplied + EventTreatmentEnd)/ EventTreatmentNotApplied	N
	IVR		N
	Music		Y
	RingBack		Y
	Silence		Y
	Busy		Y
	CollectDigits		Y
	PlayAnnouncement		Y

Table 107: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TApplyTreatment (continued)	PlayAnnouncementAnd-Digits		Y
	PlayApplication		Y
	VerifyDigits		N
	RecordUserAnnouncement		Y
	DeleteUserAnnouncement		N
	CancelCall		N
	SetDefaultRoute		N
	TextToSpeech		N
	TextToSpeechAndDigits		N
	FastBusy		Y
TGiveMusicTreatment		EventTreatmentApplied	N
TGiveRingBackTreatment		EventTreatmentApplied	N
TGiveSilenceTreatment		EventTreatmentApplied	N
DTMF (Dual-Tone Multi-Frequency) Requests			
TCollectDigits		EventDigitsCollected	N
TSendDTMF		EventDTMFSent	Y
Voice-Mail Requests			
TOpenVoiceFile		EventVoiceFileOpened	N
TCloseVoiceFile		EventVoiceFileClosed	N
TLoginMailBox		EventMailBoxLogin	N
TLogoutMailBox		EventMailBoxLogout	N
TPlayVoice		EventVoiceFileEndPlay	N
Agent & DN Feature Requests			
TAgentLogin		EventAgentLogin	Y
TAgentLogout		EventAgentLogout	Y

Table 107: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TAgentSetIdleReason		EventAgentIdleReasonSet	N
TAgentSetReady		EventAgentReady	Y
TAgentSetNotReady		EventAgentNotReady	Y
TMonitorNextCall	OneCall	EventMonitoringNextCall	Y
	AllCalls		Y
TCancelMonitoring		EventMonitoringCanceled	Y
TCallSetForward	None	EventForwardSet	Y
	Unconditional		Y
	OnBusy		N
	OnNoAnswer		N
	OnBusyAndNoAnswer		N
	SendAllCalls		N
TCallCancelForward		EventForwardCancel	Y
TSetMuteOff[g]		EventMuteOff	Y
TSetMuteOn[g]		EventMuteOn	Y
TListenDisconnect		EventListenDisconnected	Y
TListenReconnect		EventListenReconnected	Y
TSetDNDOn		EventDNDOn	Y
TSetDNDOff		EventDNDOff	Y
TSetMessageWaitingOn		EventMessageWaitingOn	N
TSetMessageWaitingOff		EventMessageWaitingOff	N
Query Requests			
TQuerySwitch[a]	DateTime	EventSwitchInfo	N
	ClassifierStat		N

Table 107: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TQueryCall[a]	PartiesQuery	EventPartyInfo	N
	StatusQuery		Y
TQueryAddress[a]	AddressStatus	EventAddressInfo	Y
	MessageWaitingStatus		N
	AssociationStatus		N
	CallForwardingStatus		N
	AgentStatus		Y
	NumberOfAgentsInQueue		Y
	NumberOfAvailableAgents-InQueue		Y
	Number Of Calls InQueue		Y
	AddressType		Y
	CallsQuery		Y
	SendAllCallsStatus		N
	QueueLoginAudit		Y
	NumberOfIdleClassifiers		N
	NumberOfClassifiersInUse		N
	NumberOfIdleTrunks		N
	NumberOfTrunksInUse		N
	DatabaseValue		N
	DNStatus		Y
	QueueStatus		Y

Table 107: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TQueryLocation[a]	AllLocations	EventLocationInfo[h]	I
	LocationData		I
	MonitorLocation		I
	CancelMonitorLocation		I
	MonitorAllLocations		I
	CancelMonitorAllLocations		I
TQueryServer[a]		EventServerInfo	Y
User-Data Requests			
TAttachUserData		EventAttachedDataChanged	Y
TUpdateUserData		EventAttachedDataChanged	Y
TDeleteUserData		EventAttachedDataChanged	Y
TDeleteAllUserData		EventAttachedDataChanged	Y
ISCC (Inter Server Call Control) Requests			
TGetAccessNumber[b]		EventAnswerAccessNumber	I
TCancelRegGetAccess-Number		EventReqGetAccess-NumberCanceled	I
Special Requests			
TReserveAgent		EventAgentReserved	I
TSendEvent		EventACK	I
TSendEventEx		EventACK	I
TSetCallAttributes		EventCallInfoChanged	I
TSendUserEvent		EventACK	Y
TPrivateService		EventPrivateInfo	Y
Network Attended Transfer/Conference Requests[i]			
TNetworkConsult		EventNetworkCallStatus	Y
TNetworkAlternate		EventNetworkCallStatus	Y

Table 107: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TNetworkTransfer		EventNetworkCallStatus	Y
TNetworkMerge		EventNetworkCallStatus	N
TNetworkReconnect		EventNetworkCallStatus	Y
TNetworkSingleStep-Transfer		EventNetworkCallStatus	N
TNetworkPrivateService		EventNetworkPrivateInfo	N
ISCC Transaction Monitoring Requests			
TTransactionMonitoring		EventACK	Y
		EventTransactionStatus	E

a. Only the requestor receives a notification of the event associated with this request.

b. This feature request may be made across locations in a multi-site environment. However, if the location attribute of the request contains a value relating to any location other than the local site—except when the response to this request is `EventError`—there will be a second event response that contains the same reference `ID` as the first event. This second event will be either `EventRemoteConnectionSuccess` or `EventRemoteConnectionFailed`.

c. Supported for SIP endpoints that have the Remote Talk feature activated.

d. SIP Server treats unsuccessful redirect operations differently, depending on whether they were initiated through 3pcc (`TRedirectCall`) or 1pcc (`302 Moved`). Incase of `TRedirectCall`, if the destination rejects the redirect for some reason (capacity, DND, for example), the call remains alive. In case of an unsuccessful SIP `302 Moved` attempt, the call is rejected. The following sample scenario explains the difference:
- DN1 has reached its configured capacity.
- DN2 is dialing DN3, and DN3 is ringing.
- DN 3 redirects to DN1.
In case of a 3pcc redirect, the call between DN2-DN3 will remain in the ringing state. In case of a 1pcc redirect, the call between DN2-DN3 is released.

e. SIP Server does not use the `extensions` parameter. Any data in this parameter is ignored.

f. SIP Server supports `TMuteTransfer` requests with the following limitations:
- The `TMuteTransfer` operation does not support the out-of-signaling-path (OOSP) transfer mode.
- The `TMuteTransfer` operation to a Hunt Group will be completed only after a Hunt Group member picks up the call.

g. SIP Server supports `TSetMuteOn` and `TSetMuteOff` only for established conferences, to allow for service observing.

h. Two subtypes are supported by `EventLocationInfo`: `LocationInfoLocationMonitorCanceled` and `LocationInfoAllLocationsMonitorCanceled`.

i. All T-Servers support NAT/C requests with `AttributeHomeLocation` provided that this attribute identifies a network location that is capable of processing such requests. Refer to the *Network T-Server Deployment Guides* to determine whether a specific Network T-Server can process these requests.

Using the Extensions Attribute

SIP Server supports the use of the `Extensions` attribute as documented in the *Genesys Events and Models Reference Manual* and the *Platform SDK 8.x .NET (or Java) API Reference*. See those documents for complete information on the T-Server events, call models, and requests. Additionally, the `Extensions` described in Table 108 are also supported.

Table 108: Use of the Extensions Attribute

Extension[a]		Used In	Description
Key	**Type**		
Advice of Charge			
AOC-Destination-DN	String	TPrivateService	If set, the value of this key points to an existing party in the call in the established state. See "Providing AoC Notifications for Established Calls" on page 102 for details.
Call Divert Destination			
after-call-divert-destination	String	TRouteCall	A destination DN where SIP Server will divert the call in cases where the caller remains on the line when all other parties have left. See "Call Divert Destination" on page 116 for details.
Call Progress Analysis			
AnsMachine	String	TMakePredictiveCall	If set, the value of this key (can only be set to `drop`) overrides the Application-level parameter `am-detected`. If the CPD result shows that the predictive call reached an answering machine, SIP Server drops the call. See "Outbound IP Solution Integration" on page 306 for details.
FaxDest	String	TMakePredictiveCall	If set, the value of this key (can only be set to `drop`) overrides the Application-level parameter `fax-detected`. If the CPD result shows that the predictive call reached a fax machine, SIP Server drops the call. See "Outbound IP Solution Integration" on page 306 for details.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
SilenceDest	String	TMakePredictiveCall	If set, the value of this key (can only be set to `drop`) overrides the Application-level parameter `silence-detected`. If the CPD result shows that silence is detected, SIP Server drops the call. See "Outbound IP Solution Integration" on page 306 for details.
Call Recording			
record	String	TRouteCall, TPrivateService	For `TRouteCall`: • When set to `destination`, call recording is initiated on the routing destination DN (agent), and will continue until the agent leaves the call. • When set to `source`, call recording is initiated on the DN that sent a call to the Routing Point (customer), and will continue until the customer leaves the call. See "Call Recording—NETANN-Based" on page 121 for details. To disable call recording, in the routing strategy, configure the `TRouteCall` request to include the `record` key with the appropriate value, as follows: • `disable_source`—to disable recording on the origination DN. • `disable_destination`—to disable recording on the destination DN. See "DN Recording Override" on page 132 for details.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
record (cont.)			For `TPrivateService`: • When set to `source`, call recording is initiated for `ThisDN` (the source DN for the request). • When set to `destination`, call recording is initiated for `OtherDN` (identified in the request). See "Call Recording—MSML-Based" on page 125 for details.
Call Supervision			
MonitorMode	String	TMonitorNextCall, TRouteCall, TSetMuteOn, TSetMuteOff, EventPrivateInfo	Specifies the monitoring mode as follows: • `mute, normal`—A mute connection. • `connect`—A three-party conference call (open supervision). • `coach`—Only the agent can hear the supervisor (whisper coaching). If `MonitorMode` is set to `coach` in the `TSetMuteOff` or `TSetMuteOn` request, the monitoring mode is changed to whisper coaching for the current supervision session. **Note:** `TSetMuteOn` and `TSetMuteOff` support only the `coach` value. See "Call Supervision" on page 139 for details.
MonitorScope	String	TMonitorNextCall, TRouteCall	Specifies the required intrusion/observation scope. Values: • `agent`—The monitoring is initiated for a specific agent. The supervisor is disconnected when the call is transferred or released, but will be connected to the next call that is routed to the same agent. • `call`—The monitoring is initiated to track an entire customer call. If the call is transferred to another agent, queue, or VRU, the monitoring function continues with the call until the customer disconnects the call.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
AssistMode	String	TSingleStep-Conference	Specifies the required assistance mode. Values: • `connect`—This is the default value - a three-party conference call. • `coach`—Only the agent can hear the supervisor (whisper coaching).
Call Transfer			
Transfer-Type	String	TRouteCall	The routing strategy can use this key to select the SIP call flow that will be used to deliver the call to a routing destination. Valid values: • `invite`—An `INVITE` transaction is used to connect an origination party with a routing destination. • `refer`—A `REFER` method is used to complete the call routing. The `Host` name in the `Refer-To` header points to SIP Server. Therefore, SIP Server remains in the signaling path when two parties are connected after the routing. **Note:** The `REFER` method cannot be used if the call is not answered by the remote party, and no treatment is applied to the call. • `oosp` (Out Of Signaling Path)—SIP Server is taking itself out of the signaling path by sending the `REFER` request or a `302` response, pointing to the routing destination, to the origination party. Two parties are connected directly when routing is complete. SIP Server no longer controls the call. See "Selecting SIP Call Flows from the Routing Strategy" on page 163 for details.
Dial Plan			
original-dialplan-digits	Integer	EventQueued, EventRouteRequest, EventRinging	If set, this key specifies the call's original destination (dialed) number before the dial plan is applied to the call. See "Dial Plan" on page 195 for details.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
DNIS_OVER	Integer	EventQueued, EventRouteRequest	Contains overdialed digits removed from DNIS when the `dnis-max-length` dial-plan rule parameter is specified. See "Removal Overdialed Digits From DNIS" on page 198 for details.
UseDialPlan	String	TRouteCall	If set, specifies how SIP Server applies the dial plan: • `full`—The dial plan is applied to the destination of `TRouteCall`, including the digit translation and forwarding rules. • `partial`—Valid for both SIP Server and SIP Feature Server dial plans. Only the digit translation is applied to a dial-plan target. Forwarding rules, such as forwarding on no answer (`ontimeout`), forwarding on busy (`onbusy`), forwarding on DND (`ondnd`), forwarding on no response (`onunreach`), and forwarding on not SIP registered (`onnotreg`) are not applied. • `false`—No dial plan is applied to the destination of `TRouteCall`. • `agentid`—No dial plan is applied to the destination of `TRouteCall`; only an agent ID provided by SIP Feature Server is added to the response **Notes:** • For ISCC calls, this extension is applied only to calls routed through an External Routing Point (`cast-type=route-notoken`) . • This extension is not supported in Business Continuity deployments.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
Divert On Ringing			
divert-on-ringing	String	TRouteCall	Valid values: • `true`—SIP Server generates `EventRouteUsed` and `EventDiverted` messages when any SIP `18x` response (`180 Ringing` or `183 Session Progress`) arrives for the `INVITE` request at the routing destination. • `false`—SIP Server postpones `EventRouteUsed` and `EventDiverted` messages until the call is answered by the routing destination with a SIP `200 OK` message. If the call is not answered within the value specified by the `after-routing-timeout` option, the destination SIP dialog is canceled and an `EventError` message is generated.
Dummy SDP			
sdp-c-host	String	TRouteCall	The value will be propagated as the connection address in the `c=` line of Dummy SDP. Typically, this would be the IP address of the media server host. See "Dummy SDP" on page 224 for details.
sdp-m-port-low	Integer	TRouteCall	Any integer that represents a valid UDP port. This value represents the low part in the range of ports to be included in the `m=` line of the SDP. Key rules: • The actual port used can equal this low value. • Only even-numbered ports are used. See "Dummy SDP" on page 224 for details.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
sdp-m-port-high	Integer	TRouteCall	Any integer that represents a valid UDP port. This value represents the high part in the range of ports to be included in the `m=` line of the SDP. Key rules: • The actual port used can equal the low value, but cannot equal this high value. • Only even-numbered ports are used. For example, if the range is 4000 to 4010, the port in the `INVITE` can be `4000, 4002, 4004, 4006, and 4008` only. • If you leave this parameter unspecified (empty), then the value of the `sdp-m-port-low` will be used. See "Dummy SDP" on page 224 for details.
Early Media			
charge-type	Integer	TApplyTreatment	If set, the value of this key overrides the value set in the `charge-type` configuration option for the current call. `1`—Free. When SIP Server receives this key in the initial TApplyTreatment request, SIP Server forces audio treatments to be played in early media, free of charge, instead of media in the established state, in deployments where the `charge-type` option is set to `2` (charged). Consecutive audio treatments are played in early media until the new `TApplyTreatment` request containing the `charge-type` key set to `2` (charged) arrives. `2`—Charged. SIP Server forces audio treatments to be played in the established state and ignores the charge-type key value in consecutive `TApplyTreatment` requests. See "Controlling Early Media with a Routing Strategy" on page 233 for details.
Emulated Agents			
AgentLogoutOnUnregister	String	TAgentLogin	If set to `true`, SIP Server logs out emulated agents on unregister. See "Emulated Agents Support" on page 236.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
BusinessCallType	String	TMakeCall, TInitiateTransfer, TMuteTransfer, TInitiateConference, TMakePredictiveCall, TAnswerCall	Specifies the call business type to be used by SIP Server for the new call or the answering party. Valid values are: • `0/private`—Private call • `1/business`—Business call • `2/work`—Work-related call
Greetings			
agent-greeting	String	TRouteCall	Specifies the name of the media file that will be used as a greeting for the agent. See "Personal Greetings" on page 319 for details.
customer-greeting	String	TRouteCall	Specifies the name of the media file that will be used as a greeting for the customer. See "Personal Greetings" on page 319 for details.
agent-greeting-type	String	TRouteCall	When set to `vxml`, enables VXML support for agent greeting. See "VXML Support for Agent Greetings" on page 321 for details.
record-agent-greeting	String	TRouteCall	Specifies whether the agent greeting or customer greeting must be recorded when both recording and greeting are enabled for the call. • If set to `true`, the agent greeting is recorded. • If set to `false`, the customer greeting is recorded. If set, the key value takes precedence over the `record-agent-greeting` option value. See "Recording an Agent Greeting" on page 324 for details.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
Music on Hold			
music	String	THoldCall, TAlternateCall, TInitiateTransfer, TInitiateConference	Specifies the file name for music on hold for the call. See "Customizing Music on Hold and in Queue" on page 179 for details.
music-on-hold	String	TRouteCall	Specifies the name of the file that is played for the music-on-hold treatment when one of the parties in the call is placed on hold. **Valid value:** The subdirectory and name of the audio file in the MCP root directory, using the following format: `<subdirectory>/<music file name>`; for example: `music/in_queue_welcome.wav` See "Customizing Music on Hold and in Queue" on page 179 for details.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
Nailed-up Connections			
connect-nailedup-on-login	String	TAgentLogin	If set, the value of this key overrides any value set in the `connect-nailedup-on-login` option setting but only for a current login session. Specifies SIP Server actions when receiving a `TAgentLogin` request from a DN with the configured nailed-up connection, as follows: • When this key is set to a Routing Point number, SIP Server immediately establishes a nailed-up connection between an agent's endpoint and the specified Routing Point. After processing the `TRouteCall` request to the `gcti::park` device, SIP Server parks the agent on `gcti::park`, establishing the persistent SIP connection with the agent's endpoint. • When this key is set to `gcti::park`, SIP Server parks the agent on the `gcti::park` device directly, establishing the persistent SIP connection with the agent's endpoint. • When this key is set to an empty value, SIP Server disables this feature for a particular agent in a current login session. See "Nailed-Up Connections for Agents" on page 287 for details.
ReasonCode	String	EventAgentNotReady	Value: `NailedUpConnectionTerminated` Specifies that the nailed-up connection is terminated.
agent-phone	String	TAgentLogin	Specifies the phone number to be used for the agent session. See "Configuring Remote Agents with Non-provisioned Phone Numbers" on page 345 for details.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
No Answer Supervision			
AFTER_ROUTING_TIMEOUT_ACTION	String	TRouteCall	If set, the value of this key overrides any value set in the `after-routing-timeout-action` configuration option for the current call. Values: • `none`—SIP Server takes no action. • `notready`—When an agent is logged in to a routing destination that does not answer the call, SIP Server sets this agent to `NotReady` state. • `logout`—When an agent is logged in to a routing destination that does not answer the call, SIP Server logs this agent out. See "No-Answer Supervision" on page 302 for details.
after-routing-timeout	Integer	TRouteCall	A positive integer that overrides the value of the `after-routing-timeout` configuration option, which specifies the length of time (in seconds) that SIP Server waits before diverting the call from the `Routing Point` DN to the destination DN after `TRouteCall` was processed. When the call is not diverted before the timeout expires, SIP Server generates an `EventError` message. If set to a value of `0` (zero), this extension is ignored and the value of the `after-routing-timeout` configuration option is used instead.
NO_ANSWER_ACTION	String	TRouteCall	If set, the value of this key overrides any value set in any of the following configuration options for the current call: • `no-answer-action` • `agent-no-answer-action` See "No-Answer Supervision" on page 302 for details.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
NO_ANSWER_OVERFLOW	Comma-separated list	TRouteCall	If set, the value of this key overrides any value set in any of the following configuration options for the current call: • `no-answer-overflow` • `agent-no-answer-overflow` • `extn-no-answer-overflow` • `posn-no-answer-overflow` See "No-Answer Supervision" on page 302 for details.
NO_ANSWER_TIMEOUT	String	TRouteCall	If set, the value of this key overrides any value set in any of the following configuration options for the current call: • `no-answer-timeout` • `agent-no-answer-timeout` • `extn-no-answer-timeout` • `posn-no-answer-timeout` See "No-Answer Supervision" on page 302 for details
ReasonCode	no-answer	EventAgentNotReady	If configured, reports `ReasonCode` set to `no-answer` when an agent is placed in the `Not Ready` state after not answering a call. See "Reporting ReasonCode" on page 303 for details.
Preview Interactions			
preview-interaction	none, false, tlib, true, chat	TRouteCall	Valid values: • `none, false`: Disables the Preview Interaction protocol. • `tlib, true`: Enables preview interaction through T-Library messaging. • chat: Enables preview interaction through SIP Instant Messaging (IM). See "Preview Interactions" on page 335 for details.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
Providing Call Participant Info (See "Providing Call Participant Info" on page 336 for details)			
LCTPartiesLength	Integer	EventUserEvent	Specifies how many parties are involved in a single call.
LCTParty<n>	Integer	EventUserEvent	Represents a party of the call, where *n* is an integer value starting from `0`.
LCTParty<n>_location	String	EventUserEvent	Provides the DN location name; that is, the name of the switch to which this DN belongs.
LCTSupervisor<n>	Integer	EventUserEvent	Represents the supervisor of the call, where *n* is an integer value starting from `0`.
LCTSupervisor<n>_ location	String	EventUserEvent	Provides the name of the switch to which this supervisor belongs.
LCTSupervisor<n>_ monitoredDN	Integer	EventUserEvent	Represents the agent monitored by this supervisor.
LCTSupervisor<n>_ mode	String	EventUserEvent	Provides Supervision mode.
Providing Caller ID			
CPNDigits	String	TMakeCall, TMakePredictiveCall, TInitiateConference, TInitiateTransfer, TRouteCall, TSingleStepTransfer, TSingleStepConference	If set, the value of this Extension overrides the username provided in the URI in the `From` header of the `INVITE` message. This Extension is not applicable when performing a `TMakeCall` request using the `REFER` method. See "Providing a Caller ID" on page 336 for details.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
DisplayName	String	TMakeCall, TMakePredictiveCall, TInitiateConference, TInitiateTransfer, TRouteCall, TSingleStepTransfer, TSingleStep-Conference	Allows users to define their own display name and Calling Party Number (CPN) for OCS outbound dialing If set, the value of this Extension is used as the display name, in addition to the CPN digits, when providing the Caller ID. This setting only applies if the `CPNDigits` key-value pair is also present in the Extensions. SIP Server maps this info from the request to the `From` header of the SIP `INVITE`. For example, `From: "DisplayName" <sips:CPNDigits@company.com>;tag=a48s`
call_timeguard_timeout	Integer	TMakePredictiveCall	If set, the value of this key specifies the maximum time, in milliseconds, allowed for post-connect CPD results to be received from the device performing CPD (either Genesys Media Server or Media Gateway). See "Outbound IP Solution Integration" on page 306 for details.
Reliable Responses			
sip-enable-100rel	true, false	TRouteCall	If the value of this key in the call request is `false`, SIP Server does not place the `100Rel` in the header for the corresponding `INVITE`. This prevents the destination DN from sending reliable provisional responses, so that the SDP in the provisional response does not force SIP Server to interrupt an ongoing voice treatment. The `Extensions` setting takes priority over the `sip-enable-100rel` option configured at the SIP Server Application-level. However, the `Extensions` setting does not take effect for calls distributed over Trunk DNs where the option `sip-server-inter-trunk` set to `true`.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
Remote Supervision			
feature	String "remote-observing"	TRouteCall	This extension triggers registration of a routed party as a supervisor with parameters specified in additional `Extensions` described in this subsection. See "Remote Supervision" on page 153 for details.
dn	String	TRouteCall	An optional DN number that can be used during the monitoring session, as a substitute for the external PSTN number that the supervisor used to dial in.
agent-dn	String	TRouteCall	The target that the supervisor wants to monitor. This parameter can be a Routing Point, ACD Queue, or an agent DN.
login-id	String	TRouteCall	(Optional) A Login ID, which will be used for remote client authorization.
monitor-type	AllCalls	TRouteCall	(Optional) The supervisor will monitor all consecutive calls for the selected target, until the supervisor decides to hang up and end the monitoring session. In between monitored calls, the supervisor's call is parked.
password	String	TRouteCall	(Optional) A password, which will be used for remote client authorization for a specified Login ID.
post-feature-dn	String	TRouteCall	(Optional) A Routing Point, to which the remote supervisor will be connected after the supervision session.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
Routing			
geo-location	String	TRouteCall, TApplyTreatment	If `TRouteCall` or `TApplyTreatment` contains the `geo-location` extension, SIP Server makes this geo-location to be the most preferable on the current call. It means that each time when the switch resource is selected based on the geo-location parameter, resources with the preferred geo-location take precedence. If the preferable geo-location is activated on the call, but `TRouteCall` or `TApplyTreatment` requests does not contain the geo-location extension specified, SIP Server deactivates the preferred geo-location mode, and switch resources are selected in accordance with a regular selection procedure.
overflow-location	String	TRouteCall, TApplyTreatment	**Note:** The `overflow-location` extension key applies only if the `geo-location` extension key is defined in the same request. See "Geo-Location for MSML-Based Services: Strict Matching" on page 393.
busy-on-reject	String (`true`, `false`)	TRouteCall	If SIP Server receives `TRouteCall` with `busy-on-reject=true` in `AttributeExtensions` and if the destination responds with the `486 Busy Here` message, it generates `EventDestinationBusy` for the origination party.
Smart OtherDN Handling			
ConvertOtherDN	String	See "Smart OtherDN Handling" on page 363.	A value of `0` disables all conversions for the call. A value of `1` forces the relevant conversion for the call.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
Server and User-Agent Headers			
User-Agent	String	TMakeCall, TMakePredictiveCall, TSingleStep-Transfer, TSingleStep-Conference, TInitiateTransfer, TInitiateConference	Enables inserting the `User-Agent` header. See "Enabling Server and User-Agent Headers" on page 185.
Sending Outgoing INVITEs with Multipart Body			
SIP_MIME_HEADERS	String	TRouteCall	Value: `Geolocation:application/pidf+xml` Passes geolocation content from `TRouteCall` into an outgoing `INVITE` message. See "Sending Outgoing INVITEs with Multipart Body" on page 350.
Geolocation	String	TRouteCall	Carries geolocation content of the body to be included into an outgoing `INVITE` message. See "Sending Outgoing INVITEs with Multipart Body" on page 350.
Trunk Capacity			
Dest-Capacity	Integer	TRouteCall	Specifies the trunk capacity as defined in the URS routing strategy. SIP Server checks the current load of the targeted Trunk, and if the Dest-Capacity value is exceeded, SIP Server replies to the RouteRequest with a `403 Forbidden` error message.
T-Server Common Part Extensions			
sdn-licenses-in-use	Integer	EventServerInfo	Specifies how many SDN licenses are currently in use.
sdn-licenses-available	Integer		Specifies how many SDN licenses are currently available.

Table 108: Use of the Extensions Attribute (Continued)

Extension[a]		Used In	Description
Key	**Type**		
Video Support			
VideoFile	String	TSingleStep-Conference	A string that contains the name of the video file that will be played for the customer. If this key-value pair is not specified, the default video file will be played. The default video file is configured in the SIP Server `Application` using the `default-video-file` configuration option. See "Video Support" on page 382 for details.
AgentVideo	String	TSingleStep-Conference	A string that identifies the origin of the video stream played to the agent. The values are as follows: • `from-third-party`—The agent receives video from a third party—that is, the party that participated in the call before the operation started. • `to-third-party`—The agent receives the same video stream played to the third party—that is, video from the file specified by the `VideoFile` parameter. See "Video Support" on page 382 for details.

a. If you use the IRD for creating a routing strategy, it might require adding a prefix to the key name: {d} for an integer or {s} for a string. Refer to the *Universal Routing documentation*.

Error Messages

Table 109 presents the complete set of error messages SIP Server distributes in `EventError`, which SIP Server generates when it cannot execute a request because of an error condition.

Table 109: Error Messages for SIP Server

Code	Symbolic Name	Description
40	TERR_NOMORE_LICENSE	No more licenses are available.
41	TERR_NOT_REGISTERED	Client has not registered for the DN.
42	TERR_RESOURCE_SEIZED	Resource is already seized.

Table 109: Error Messages for SIP Server (Continued)

Code	Symbolic Name	Description
43	TERR_IN_SAME_STATE	Object is already in requested state.
50	TERR_UNKNOWN_ERROR	Unknown error code. Request cannot be processed.
51	TERR_UNSUP_OPER	Operation is not supported.
52	TERR_INTERNAL	Internal error.
53	TERR_INVALID_ATTR	Attribute in request operation is invalid.
54	TERR_NO_SWITCH	No connection to the switch.
55	TERR_PROTO_VERS	Incorrect protocol version.
56	TERR_INV_CONNID	Connection ID in request is invalid.
57	TERR_TIMEOUT	Switch or T-Server did not respond in time.
58	TERR_OUT_OF_SERVICE	Switch or T-Server is out of service.
59	TERR_NOT_CONFIGURED	DN is not configured in the Configuration Database.
61	TERR_INV_CALL_DN	DN in request is invalid.
71	TERR_INV_CALD_DN	Invalid called DN.
93	TERR_DEST_INV_STATE	Destination invalid state.
96	TERR_CANT_COMPLETE_CONF	Call cannot add new conference party.
118	TERR_SERV_UNAVAIL	Requested service is unavailable.
119	TERR_BAD_PASSWD	Password was invalid. May occur when SIP Server receives a `404 Not Found` message from the Media Gateway after an unsuccessful attempt to route a call.
122	TERR_CANT_REG_DNS	Cannot register DNs on the switch.
123	TERR_DN_NOT_EXIST	DN for association does not exist.
128	TERR_BAD_DN_TYPE	Invalid DN type for DN registration.
166	TERR_RES_UNAVAIL	(JTAPI object) resource is not available.
168	TERR_INV_ORIG_ADDR	Originating address in request was invalid.
173	TERR_UNSUCC_CONFER	Unsuccessful conference.

Table 109: Error Messages for SIP Server (Continued)

Code	Symbolic Name	Description
177	TERR_TARG_DN_INV	DN target (in route call) was invalid.
185	TERR_SET_WRONG_STATE	Set is in the wrong state for invocation.
192	TERR_AGENT_ID_INV	Agent ID is invalid.
195	TERR_CFW_DN_INV	Call forwarding address is invalid.
212	TERR_ADMIN_DEV_DIS	Device disabled by administration.
223	TERR_BAD_PARAM	Bad parameter is passed to function.
226	TERR_OUT_OF_MEM	Out of memory (local).
231	TERR_DN_BUSY	DN is busy.
232	TERR_DN_NO_ANSWER	No answer at a DN.
233	TERR_CALL_REJECTED	Call has been rejected.
236	TERR_TIMEOUT_PRF_OP	Timeout performing operation.
237	TERR_DISCON_CALL	Call has disconnected.
243	TERR_CLNT_NOT_MON	Internal error—client corrupted in T-Server.
259	TERR_INV_PASSWD	Invalid credentials (login_id or password).
282	TERR_NO_VCHAN_AVAIL	No voice channel available.
291	TERR_OTHER_TEL_OPER	Other telephony operation is in progress.
302	TERR_INV_DTMF_STRING	DTMF string invalid.
355	TERR_OPER_INVALID	Operations are invalid on system messages.
410	TERR_INAPPR_TRTM	Invalid treatment type.
415	TERR_INV_DEST_DN	The destination DN in the request is invalid.
470	TERR_PARTY_NOT_ON_CALL	Party in request is not involved in a call.
496	TERR_INV_CALL_STATE	Party in request is in the `call` state.
506	TERR_RECVD_INV_STATE	Call/Party is in invalid state for this time
527	TERR_ALRDY_SIGN_IN	The agent's sign-in number is already active at another console.
549	TERR_OTHER_INV_FRMT	Other invalid format.
565	TERR_INVALID_STATE	Invalid state.

Table 109: Error Messages for SIP Server (Continued)

Code	Symbolic Name	Description
649	TERR_ASAI_OUT_CALL_BARRED	Outgoing call has been barred.
700	TERR_INV_LOGIN_REQ	Agent cannot log in at this time.
701	TERR_INV_LOGOUT_REQUEST	Agent cannot logout.
702	TERR_INV_READY_REQ	Agent cannot go to `ready` state.
726	TERR_DMS_INV_AUTHCODE	Invalid AuthCode.
1141	TERR_CSTA_OPER_REQ_INCOMPAT	Request incompatible with object.
1143	TERR_CSTA_OPER_OBJ_NOT_KNOWN	Object is unknown.
1150	TERR_CSTA_OPER_INV_CALL_ID	Invalid call identifier.
1151	TERR_CSTA_OPER_INV_DEV_ID	Invalid device identifier.
1152	TERR_CSTA_OPER_INV_CONN_ID	Invalid connection identifier.
1161	TERR_CSTA_INCOMP_INCORR_STATE	Incorrect object state.
1165	TERR_CSTA_INCOMP_NO_CALL_TO_CLEAR	No call to clear.
1183	TERR_CSTA_SUBRES_OUTST_LIMIT_EXC	Rejects the second consecutive call party control request if it comes in less than one second after the first one.
1605	TERR_INVALIDPARTY	Party in request was invalid on switch.
2100	TERR_SIPCS_ERR	Unspecified kind of error.
2101	TERR_SIPTS_PENDING_INV_TRN	Operation cannot be completed due to the pending INVITE transaction.
2130	TERR_SIPTS_INCOMPATIBLE_OPTION_VALUE	Operation cannot be initiated due to current value of the configuration option.
2131	TERR_SIPTS_REFER_ENABLED_SHOULD_BE_TRUE	Operation cannot be initiated if “refer-enabled=false”.
2132	TERR_SIPTS_NOT_SUPPORTED_IN_CURRENT_SCENARIO	Operation is not supported in current scenario.
3002	TERR_PRIVVIOLATION	User doesn’t have security privilege on the switch.
3005	TERR_UNSUCC_ROUTECALL	`TRouteCall` request was unsuccessful.

Known Limitations

Several known limitations result from the current SIP Server and softswitches/gateways interface:

- Due to the specifics of gateway behavior in performing SIP `REFER` methods, support for remote agents has some limitations. In order to use remote agents, you must perform one of the two following steps:
  - Provision customers and remote agents to use physically separate gateways (otherwise, calls from agents to customers take shortcuts within gateways, which means that SIP Server loses track of the call and therefore cannot perform call control). Even in this configuration, direct calls between two remote agents on the same gateway are not visible to SIP Server.

  Or,
  - Disable the SIP `REFER` method for the gateways where the remote agents are located. This enables SIP Server to see agent-to-customer and agent-to-agent calls.
- SIP Server supports `TSetMuteOn` and `TSetMuteOff` only for established conferences, to allow for service observing.
- SIP Server does not support User Datagram Protocol (UDP) messages of more than 16 KB in length. If SIP Server encounters a message larger than 16 KB, it truncates the message without warning. This can cause problems in scenarios that require larger UDP messages. For example, when using the Busy Lamp Field (BLF) feature, SIP Server can sometimes receive UDP messages of up to 35 KB. In this scenario, the 16 KB UDP limitation restricts SIP Server support to a maximum of 20 monitored users over a single BLF subscription.
- An `EventReleased (switch::)` message is issued when the last internal party leaves a call.
- Third-party call control (3pcc) blind conference calls are not supported.
- The `TDeleteFromConference` request is not supported for first-party call control (1pcc) conference calls.
- SIP Server does not report a first-party call control (1pcc) conference with mixing on an endpoint.
- SIP Server does not process third-party call control (3pcc) requests that have the same value for `AttributeThisDN` and `AttributeOtherDN`, even if the requests have a non-empty value for `AttributeLocation`, meaning that the destination DN (`AttributeOtherDN`) is located remotely. These requests are rejected with `EventError` with `Error Code` of `415` (`Invalid Destination DN`).
- SIP Server does not support registering several SIP devices on the same DN—only a single SIP device must be registered on a DN at a time.

- When SIP Server receives multiple media lines in the SDP of a SIP message, it adds additional media lines in the SDP that it sends in a SIP message to an endpoint.
- SIP Server supports only two calls on a particular DN. If a third 3pcc call is initiated from that DN, SIP Server generates an error message.

Third-Party Equipment—Known Limitations

The known limitations when SIP Server is operating with a third-party equipment are as follows:

- The Siemens OpenScape Voice switch is supported with the following limitations:
  - 1pcc (first-party call control) calls are only supported starting with switch version V5.
  - The remote answer feature (`TAnswerCall`) is not available with version 2.0. However, the remote answer feature is supported with version 2.2.
  - For 3pcc calls with OpenScape Voice (HiPath 8000) switch versions, SIP Server must be configured to use re-INVITE-based call control methods.
  - Genesys recommends setting the `dual-dialog-enabled` configuration option to `false` if Siemens optiPoint phones are used in `re-INVITE` mode for third-party call control (3pcc) operations.
  - Sometimes SIP Server cannot retrieve a call within a mixed phone environment. To avoid this problem, set the `sip-hold-rfc3264` option with a proper value on the DN.
- The following media gateways support `re-INVITE`-based call transfers only:
  - Alcatel 7515
  - Cisco A5350
  - Cisco A5400
  - Asterisk
  - Sonus
- When using the Paraxip gateway for Outbound IP, SIP Server cannot disable AM detection using the `TMakePredictiveCall` request. If you set the Extension `answer_type_recognition` to `no_am_detection` in the `TMakePredictiveCall` request, SIP Server might still report AM as the CPD result in the `EventEstablished` that it generates for the call.

GENESYS

Chapter

7 SIP Server Configuration Options

This chapter describes the configuration options that are unique to SIP Server and contains the following sections:

SIP Server also supports common log options described in Chapter 10 on page 715 and options common to all T-Servers described in Chapter 11 on page 737.

Application-Level Options

Unless specified otherwise, set configuration options in the SIP Server `Application` object, using one of the following navigation paths:

- In Genesys Administrator Extension (GAX)—`Application` object > `Application Options` tab
- (Obsolete) In Genesys Administrator—`Application` object > `Options` tab > `Advanced View (Options)`
- (Obsolete) In Configuration Manager—`Application` object > `Properties` dialog box > `Options` tab

For instructions on how to manage (add, update, remove) configuration options, refer to the *Genesys Administrator Extension Help* at: https://docs.genesys.com/Documentation/GA/9.0.0/user/ConfigMgmt

TServer Section

This section must be called `TServer`.

For ease of reference, the options have been arranged in alphabetical order.

acw-in-idle-force-ready

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Emulated Agents" on page 234

Specifies whether, after timed manual wrap-up (when option `timed-acw-in-idle` is set to `true`), SIP Server forces the agent to the `Ready` state. With value `false`, SIP Server returns the agent to the state prior to requesting manual wrap-up.

Note: For compatibility with the previous SIP Server releases, you can use the name `cwk-in-idle-force-ready` for this option as an alias.

acw-persistent-reasons

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

If set to `true`, SIP Server populates `AttributeReason` in unsolicited `EventAgentReady/EventAgentNotReady` messages generated by the After Call Work (ACW) feature.

If set to `false`, SIP Server does not populate `AttributeReason` in `EventAgentReady/EventAgentNotReady` messages.

after-routing-timeout

Default Value: `10`
Valid Values: Any positive integer
Changes Take Effect: Immediately
Related Feature: "No-Answer Supervision" on page 302

Specifies the length of time (in seconds) that SIP Server waits before diverting the call from the `Routing Point` DN to the destination DN after `TRouteCall` was processed. If the call is not diverted before the specified number of seconds, the `EventError` message is issued, containing the `Reference ID` of the `TRouteCall` request.

When set to `0` (zero), the `after-routing-timeout` timer is disabled.

The `after-routing-timeout` option is also dependent on the `divert-on-ringing` option:

- When the `divert-on-ringing` option is set to `true`, the call is considered as "diverted" when the `180 Ringing` message arrives from the destination DN.

- When the `divert-on-ringing` option is set to `false`, the call is considered as "diverted" when the `200 OK` message arrives from the destination DN.

Note: You can override this option by configuring the routing strategy to include the `after-routing-timeout` key-value pair to the `Extensions` attribute of the `TRouteCall`.

after-routing-timeout-action

Default Value: `none`
Valid Values:

`none`	SIP Server takes no action.
`notready`	When an agent is logged in to a routing destination that does not answer the call, SIP Server sets this agent to NotReady state.
`logout`	When an agent is logged in to a routing destination that does not answer the call, SIP Server logs this agent out.

Changes Take Effect: On the next call
Related Feature: "No-Answer Supervision" on page 302

Defines SIP Server's default action if the `after-routing-timeout` expires. If `after-routing-timeout` is disabled (set to `0`), then SIP Server ignores the `after-routing-timeout-action` value.

When you set this option to a valid non-default value, it takes priority over the `agent-no-answer-action` and `no-answer-action` parameters, which are not applied to an agent logged in to a routing destination if the `after-routing-timeout` expires. In addition, none of the following parameters are applied if the `after-routing-timeout` is in progress: `agent-no-answer-overflow`, `no-answer-overflow`, or `extn-no-answer-overflow`.

agent-allow-empty-password

Default Value: `true`
Valid Values: `true, false`
Change Take Effect: For the next agent login
Related Feature: "Agent Login and State Update on SIP Phones" on page 329

When set to `true`, SIP Server allows an agent to log in from a SIP phone without the password. When set to `false`, SIP Server rejects agent logging from a SIP phone without the password.

agent-emu-login-on-call

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Emulated Agents Support" on page 236

Specifies whether the SIP Server allows an emulated agent login on a device where there is a call in progress. Note that SIP Server always allows an emulated agent logout on a device where there is a call in progress.
The option can be set in the Configuration Layer in the following places in order of precedence (highest to lowest):

1. The `TServer` section of an agent DN.
2. The `TServer` section of a device.
3. The `TServer` section of the application.

The value can also be set by using the `AgentEmuLoginOnCall` extension in the `TAgentLogin` requests. The value specified by the extension, where present, takes precedence over the settings configured in the Configuration Layer.

agent-group

Default Value: No default value
Valid Value: Any agent-group value
Changes Take Effect: At the next agent login session

Specifies a value for an agent group that will be used for SIP Server reporting.

SIP Server obtains the value for this option in the following order of precedence:

1. In the `TServer` section of the `DN` object.
2. In the `TServer` section of the SIP Server `Application` object.

agent-logout-on-unreg

Default Value: `false`
Valid Values:

`true`	SIP Server will log out emulated and native agents on unregister.
`false`	SIP Server will not log out emulated or native agents on unregister.
`emu-only`	SIP Server will log out only emulated agents on unregister.

Changes Take Effect: After an agent logs out and then logs in again
Related Feature: “Emulated Agents Support” on page 236

Specifies whether SIP Server performs an automatic logout of an agent whenever their client application unregisters the DN from the SIP Server. This happens whenever a client application disconnects from the SIP Server.

The option can be set in the Configuration Layer in the following places in order of precedence (highest to lowest):

1. The `TServer` section of the device representing the agent’s group (such as an ACD Queue).
2. The `TServer` section of an agent DN.
3. The `TServer` section of a device.
4. The `TServer` section of the Application level.

The Configuration Layer settings may be overridden by adding the extension `AgentLogoutOnUnregister` to the `TAgentLogin` request.

The initial `TAgentLogin` request can override the current agent association by adding the `AgentLogoutOnUnregister` key of the `Extensions` attribute with a value of `true`.

Note: This option is not applicable if the `logout-on-disconnect` option is set to `true`.

agent-logout-reassoc

Default Value: `false`
Valid Values:

`true`	SIP Server automatically associates a new client application with the agent.
`false`	SIP Server does not automatically associate a new client application with the agent.

Changes Take Effect: After an agent logs out and then logs in again
Related Feature: "Emulated Agents Support" on page 236

Specifies whether SIP Server automatically associates a new client application with the agent, when the application either:

- Registers on the agent DN, or;
- Sends a login request while SIP Server is currently waiting to log the agent out due to the previously associated client disconnecting.

Note: The new client application must have the same application name as the previously disconnected client.

agent-no-answer-action

Default Value: `none`
Valid Values:

`none`	SIP Server takes no action on agents when calls are not answered.
`notready`	SIP Server sets agents to `NotReady` when calls are not answered.
`logout`	SIP Server automatically logs out agents when calls are not answered.

Changes Take Effect: Immediately
Related Feature: "No-Answer Supervision" on page 302

Defines SIP Server's default action if a logged-in agent fails to answer a call within the time defined in the `agent-no-answer-timeout` option. See also the `NO_ANSWER_ACTION` extension in section "Using the Extensions Attribute" on page 414 for more information about how this option is used.

SIP Server obtains the value for this option in the following order of precedence:

1. `no-answer-action` if defined at an `Agent Login` level.
2. `agent-no-answer-action` if defined at a SIP Server `Application` level.

agent-no-answer-overflow

Default Value: `none`
Valid Values:

`none`	SIP Server does not attempt to overflow a call on an agent desktop when the when the time specified in the `agent-no-answer-timeout` option expires.
`recall`	SIP Server returns the call to the last distribution device (the device reported in the `ThisQueue` attribute of the call) when the when the time specified in the `agent-no-answer-timeout` option expires.
`release`	SIP Server releases the call.
Any valid overflow destination	SIP Server returns the call to the specified destination when the time specified in the `agent-no-answer-timeout` option expires.

Changes Take Effect: Immediately
Related Feature: “No-Answer Supervision” on page 302

Specifies a sequence of overflow destinations (separated by a comma) that SIP Server attempts to overflow to when the time specified in the `agent-no-answer-timeout` option expires. SIP Server attempts to overflow in the order specified in the list.

When all overflow attempts fail, SIP Server abandons overflow. See also the `NO_ANSWER_OVERFLOW` extension in section “Using the Extensions Attribute” on page 414 for more information about how this option is used.

When the list of overflow destinations contains the `recall` value and the call was not distributed, SIP Server skips to the next destination in the list.

SIP Server obtains the value for this option in the following order of precedence:

1. `no-answer-overflow` if defined at an `Agent-Login` level and applies to logged-in agents only.
2. `agent-no-answer-overflow` if defined at a SIP Server `Application` level.

agent-no-answer-timeout

Default Value: `15`
Valid Value: Any integer from `0`–`600`
Changes Take Effect: Immediately
Related Feature: “No-Answer Supervision” on page 302

Defines the default time (in seconds) that SIP Server allows for a logged-in agent to answer a call before executing the actions defined in the `agent-no-answer-overflow` and `agent-no-answer-action` options.

If set to `0`, the Agent No-Answer Supervision feature is disabled. See the `NO_ANSWER_TIMEOUT` extension in section "Using the Extensions Attribute" on page 414 for more information about how this option is used.

SIP Server obtains the value for this option in the following order of precedence:

1. `no-answer-timeout` if defined at an `Agent Login` level and applies to logged-in agents only.
2. `agent-no-answer-timeout` if defined at a SIP Server `Application` level.

agent-only-private-calls

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Emulated Agents" on page 234

Specifies whether SIP Server blocks the classification of a call's business type as private when there is no agent on the call. If set to `false`, calls with no agents present are classified as `private`, enabling No-Answer Supervision (NAS) to be applied for private calls.

If set to `true`, calls remain classified as `unknown` and NAS is not applied to such calls.

agent-strict-id

Default Value: `false`
Valid Values: `true, false, passwd`
Changes Take Effect: Immediately
Related Feature: "Emulated Agents" on page 234

Specifies whether SIP Server enables any `AgentID` to be used during login (value `false`), or only those configured in the Configuration Layer (value `true`). If set to `passwd`, SIP Server verifies that a value of an attribute `password` from a `TAgentLogin` request matches the value of the `Password` field from the Advanced Tab of the Agent Login Configuration Layer object.

alternate-route-profile

Default Value: An empty string
Valid Values: A Routing Point DN with non-empty Default DNs list
Changes Take Effect: For the next default routing
Related Feature: "Alternate Routing for Unresponsive URS/ORS" on page 109

Defines a Routing Point DN with a Default DNs list in its configuration. This list is used for alternate routing for all Routing Points with an empty Default DNs list.

am-detected

Default Value: `drop`
Valid Values:

`drop`	The call is released.
`connect`	The connected call remains connected.

Changes Take Effect: Immediately
Related Feature: "Outbound IP Solution Integration" on page 306

Specifies the behavior of SIP Server where CPD is operating, and an answering machine is detected on an Outbound call. SIP Server provides the CPD result in `UserData` attached to the call as a key-value pair with key `AnswerClass` containing the value `AM`. This `UserData` in `EventRouteRequest` provides extra information to the strategy, so that the strategy can decide to drop the `AM` call if required.

audio-codecs

Default Value: `telephone-event, PCMU, PCMA, G723, G729, GSM`
Valid Values: Any valid codec; if the codec has a dynamic payload and clock rate other than 8000, you must include the clock rate along with the codec name, separated by a slash (/). For example, SIREN/16000.
Changes Take Effect: On the next call
Related Option: `sip-enable-sdp-codec-filter` on page 521

Specifies a list of codecs that SIP Server uses to modify the Session Description Protocol (SDP) message body during SIP re-negotiation. SIP Server picks codecs in this list from the incoming SDP, and passes only these codecs on to the remote side. As a result, all call center audio traffic is established based on the codecs listed in this option.

This option is also used to supply a list of codecs in the SDP offer for the initial `INVITE` to the caller for a 3pcc `MakeCall` operation.When creating the `INVITE`, SIP Server will use from this list only those codecs with a static payload.

You can also specify this option at the DN-level. If `sip-enable-sdp-codec-filter` is set to `true` in the DN configuration, SIP Server, as it propagates the SDP to and from the device represented by this DN, will use as its list of available codecs the value configured in the `audio-codecs` option on the DN rather than on the application. If `sip-enable-sdp-codec-filter` is set to `true` at both the application and the DN level, the `audio-codecs` configured in the DN should contain a subset of the `audio-codecs` configured in the application.

Note: This option only takes effect for SDP renegotiation if `sip-enable-sdp-codec-filter` is set to `true`. The option is still used for 3pcc `MakeCall` operations even if `sip-enable-sdp-codec-filter` is set to `false`.

---

auto-logout-ready

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: “Automatic Inactive Agent Logout” on page 114

Enables a stricter enforcement of the automatic agent-logout policy (as set in the related `auto-logout-timeout` option). If this option is set to `true`, SIP Server logs the agent out regardless of the agent state. If this option is set to `false`, SIP Server does not log agents out when in the following agent states: `Ready`, `NotReady/ACW`, `NotReady/AuxWork`, `NotReady/LegalGuard`.

You can configure this option in the `TServer` section of the following objects (listed in order of precedence):

- `Agent Login` object
- DN object (`ACD Position` or `Extension` DN) that represents the device to which the agent is logged in.
- DN object (`Routing Point` or `ACD Queue` DN) that represents the queue to which the agent is logged in.
- SIP Server `Application` object, which specifies the server-wide default.

auto-logout-timeout

Default Value: `0`
Valid Values: `0`, or any positive integer up to `35791`
Changes Take Effect: Immediately
Related Feature: “Automatic Inactive Agent Logout” on page 114

Enables automatic agent logout and specifies the length of time, in minutes, after which the logout occurs. To enable this feature, enter a value of `1` or greater; the agent is allowed to remain inactive for this length of time before having to be automatically logged out. To disable this feature, enter a value of `0` (default).

You can configure this option in the `TServer` section of the following objects (listed in order of precedence):

- `Agent Login` object
- DN object (`ACD Position` or `Extension` DN) that represents the device to which the agent is logged in.
- DN object (`Routing Point` or `ACD Queue` DN) that represents the queue to which the agent is logged in.
- SIP Server `Application` object (`Application Options` tab), which specifies the server-wide default.

backup-init-check

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After restart
Related Feature: See "Verifying Initialization Status in Backup SIP Servers" in the *SIP Server High-Availability Deployment Guide*

When set to `true`, SIP Server in Backup mode verifies that all internal components (T-Controller, Smart Proxy, Interaction Proxy, and Operational Information thread) are successfully initialized, and can provide the service when SIP Server switches to Primary mode. If some components fail to complete initialization, SIP Server reports the `SERVICE_UNAVAILABLE` status to LCA/SCS. The timeout for internal components to complete initialization is defined by the `backup-init-check-timeout` option.

backup-init-check-timeout

Default Value: 60
Valid Values: 15-3600
Changes Take Effect: After restart
Related Feature: See "Verifying Initialization Status in Backup SIP Servers" in the *SIP Server High-Availability Deployment Guide*

Restricted option. Specifies the timeout, in seconds, during which SIP Server verifies that all internal components are successfully initialized in a scenario described by the `backup-init-check` option.

backup-sip-port-check

Default Value: true
Valid Values: true, false
Changes Take Effect: After restart
Related Feature: See "`Verifying Initialization Status in Backup SIP Servers`" in the *SIP Server High-Availability Deployment Guide*

When set to `true`, SIP Server in Backup mode attempts to open a SIP port. If the port opens successfully, no SIP messages are processed, and SIP Server closes the port immediately. If the SIP port does not open, SIP Server reports the `SERVICE_UNAVAILABLE` status to LCA/SCS. This functionality is enabled only when `backup-init-check` is set to `true`. The functionality is disabled in the IP Address Takeover configuration, when the `control-vip-scripts` option is set to `true`.

backwds-compat-acw-behavior

Default Value: `false`
Valid Value: `true, false`
Changes Take Effect: Immediately
Related Feature: "Emulated Agents" on page 234

Specifies whether pre-7.5 behavior after-call work is enabled (value = `true`) or disabled (value = `false`), for backward compatibility.

Calls While in Emulated ACW

With value `true`, if an agent receives or makes a business call while in emulated After Call Work (ACW), SIP Server does the following:

1. Stops the ACW timer.
2. Forces the agent to the `Ready` state.
3. Restarts ACW (and the legal-guard timer) after the new business call is released.

If an agent makes or receives a work-related call while in ACW, SIP Server does the following:

1. Suspends the ACW, but leaves the agent in the ACW state.
2. Resumes the ACW timer once the work-related call is released.

SIP Server categorizes as a *work-related call* any call that an agent makes while in the `NotReady` state with `workmode` set to `AfterCallWork` or `AuxWork`.

After the ACW and any configured legal-guard time have been completed, the agent is forced to the `Ready` state. If an agent makes or receives a private call during ACW, no action is taken and the ACW timer keeps running.

With value `false`, if an agent receives or makes a business call while in emulated ACW, SIP Server does the following:

1. Stops the ACW timer and adds the remaining amount of ACW to the ACW period for the new call. If either of the ACW periods is untimed, the resulting ACW will also be untimed.
2. Forces the agent to the `NotReady (ManualIn)` state.
3. Restarts ACW after the business call is released.

If an agent receives a work-related or private call while in ACW, SIP Server does the following:

1. Suspends the ACW timer.
2. Forces the agent to the NotReady (`ManualIn`) state.
3. Returns the agent to the ACW state and resumes the ACW timer once the call is released.

After the ACW and any configured legal-guard time have been completed, the agent is forced to the `Ready` state.

If an agent makes a work-related call or private call while in ACW, SIP Server does the following:

1. Continues running the ACW timer.
2. Does not change the agent state, which remains in ACW.
3. Returns the agent to the `Ready` state when the ACW timer expires.

Business Call While Not Ready

With value `true`, if an agent receives a business call while in an emulated `NotReady` state, except for ACW or legal-guard time, SIP Server sets the agent state to `Ready` for the duration of the business call.

With value `false`, if an agent receives a business call while in emulated `NotReady` state, except for ACW or legal-guard time, SIP Server will maintain the current agent state for the duration of the business call. After the call and any associated wrap-up are completed, SIP Server will return to the previous `NotReady` state. Note that no legal-guard time is applied, because the agent does not go into the `Ready` state.

blind-transfer-enabled

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether SIP Server processes certain transfer requests while a consultation call is in the dialing state. If set to `true`, SIP Server processes `TCompleteTransfer` requests or SIP `REFER` messages while a consultation call is in the dialing state. Otherwise, such requests are rejected.

Notes:
- This option can also be configured at the DN level. The DN-level configuration takes precedence.
- This option is for blind transfers only. Blind conference calls are not supported.

bsns-call-dev-types

Default Values: `+acdq +rp +rpq +xrp`
Valid Values: A set of space separated flags.

`+/-acdq`	Turns on/off the classification of the call type as business on an ACD Queue.
`+/-rp`	Turns on/off the classification of the call type as business on a Routing Point.
`+/-rpq`	Turns on/off the classification of the call type as business on a Routing Queue.
+/-`xrp`	Turns on/off the classification of the call type as business on an External Routing Point.

Changes Take Effect: Immediately

Related Feature: “Business-Call Handling” on page 235

Specifies which types of distribution devices will be exempt from default business-call handling. By default, SIP Server classifies any call arriving at a distribution device (ACD Queue, Routing Point, Routing Queue, External Routing Point) as a business call. Using this option, you can disable automatic classification for calls to a particular type of distribution device. For example, if the value for this option is set to `-rp`, calls to `Routing Point` DNs will not be automatically classified as `business`, allowing the routing strategy to use the `BusinessCallType` extension key.

busy-tone

Default Value: `music/busy_5sec`
Valid Values: Name and path of any valid audio file
Changes Take Effect: Immediately for all new calls

Specifies the audio file to be played for the `Busy` treatment.

busy-tone-duration

Default Value: `5`
Valid Values: Any integer from `1`–`3600`
Changes Take Effect: The next time the `Busy` treatment is played
Related Options: `busy-tone`, `fast-busy-tone`, `sip-busy-type`

Specifies, in seconds, the maximum duration of the busy or fast-busy tone treatment is played to a party because of a busy condition or an error.

call-monitor-acw

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: On the next call

If set to `true`, SIP Server applies emulated After Call Work (ACW) to a service observer (supervisor) after a call is released.

call-observer-with-hold

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Place: Immediately

If this option is set to `true`, SIP Server sends the initial `INVITE` with hold SDP to a supervisor in a monitoring scenario. This is done so Early Media does not affect the conversation between call participants when a supervisor is monitoring the session. After the supervisor answers the call, SIP Server sends a re-`INVITE` to add the supervisor to the conference.

cancel-monitor-on-disconnect

Default Value: `true`
Valid Values:

`true`	Call supervision subscription is canceled.
`false`	Call supervision subscription is not canceled.

Changes Take Effect: Immediately
Related Feature: “Call Supervision” on page 139

Specifies whether the call supervision subscription is canceled when the client that requested it disconnects from SIP Server.

cancel-monitor-on-unpark

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether SIP Server cancels the supervisor monitoring subscription when the nailed-up connection is dropped. If set to `true`, SIP Server cancels the supervisor subscription when the supervisor is unparked. If set to `false`, SIP Sever does not cancel the supervisor subscription in case of unpark.

capacity-sip-error-code

Default Value: `603`
Valid Values: `400–699`
Changes Take Effect: Immediately
Related Feature: “Trunk Capacity Control” on page 372

Specifies the SIP error code that SIP Server distributes in response to a rejected SIP request (incoming or outgoing) when trunk capacity is reached.

capacity-tlib-error-code

Default Value: No default value
Valid Values: Any positive integer
Changes Take Effect: Immediately
Related Feature: “Trunk Capacity Control” on page 372

Specifies the error code that SIP Server distributes in the `AttributeErrorCode` of the T-Library `EventError` message in response to a rejected T-Library request when trunk capacity is reached. Recommended value is `282`, which corresponds to the error message `No Voice Channel Available`. If the value of this option is not specified, SIP Server uses different error codes for different T-Library requests to indicate a capacity problem.

cid-enable-on-vtp

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call
Related Feature: “Caller Information Delivery Content for AT&T Trunks” on page 117

Use this option to simplify provisioning of the IVR that is configured through the Voice Treatment Port (VTP) DNs.

- If set to `true`, SIP Server passes the CID content to the VTP DN in the initial `INVITE`.
- If set to `false`, SIP Server does not pass the CID content to the VTP DN in the initial `INVITE`.

Note: CID content default encoding is in UTF-8. No content re-encoding and no URL encoding is performed; CID is passed to SIP destinations as it is received

clamp-dtmf-allowed

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call
Related Feature: "DTMF Clamping in a Conference" on page 220

When set to `true`, enables the DTMF Clamping feature. When set to `false`, disables this feature. This setting also preserves backward compatibility.

clearcall-sip-reject-code

Default Value: `603`
Valid Values: A valid SIP error response in the range of 400-669
Changes Take Effect: Immediately

Specifies the SIP code that SIP Server distributes while performing a ClearCall procedure for an inbound SIP call leg in the calling state.

collect-tone

Default Value: `music/collect`
Valid Values: Name and path of any valid audio file
Changes Take Effect: Immediately for all new calls

Specifies the audio file that SIP Server uses to produce a noncompletion tone played during DTMF digit collection.

connect-nailedup-on-login

Default Value: An empty string
Valid Values: Routing Point number, `gcti::park`
Changes Take Effect: At the next agent login session
Related Feature: "Nailed-Up Connections for Agents" on page 287

Specifies SIP Server actions when receiving a `TAgentLogin` request from a DN with the configured nailed-up connection, as follows:

- When this option is set to a DN of type `Routing Point`, SIP Server immediately establishes a nailed-up connection between an agent's endpoint and the specified Routing Point. After processing the `TRouteCall` request to the `gcti::park` device, SIP Server parks the agent on `gcti::park`, establishing the persistent SIP connection with the agent's endpoint.
- When this option is set to `gcti::park`, SIP Server parks the agent on the `gcti::park` device directly, establishing the persistent SIP connection with the agent's endpoint.
- When the value for this option is not specified (the default), SIP Server does not take any action.

Note: This option can be configured at both Application and DN levels. Setting at the DN level takes precedence over the Application level.

control-remote-vip-scripts

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: After SIP Server restart
Related Feature: See the *SIP Server 8.1 High-Availability Deployment Guide*.

If only a single SIP Server is started out of the HA pair, the `sip-vip-script-down` option might need to be executed on the host where SIP Server is not started. When set to `true`, SIP Server connects to the remote LCA and executes the Virtual IP address control scripts on the remote host. This option applies only if the value of the `control-vip-scripts` option is set to `true`.

Note: This option is reserved by Genesys Engineering. Use it only when requested by Genesys Customer Care.

control-vip-scripts

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After SIP Server restart
Related Feature: See the *SIP Server 8.1 High-Availability Deployment Guide*.

For the Hot Standby configuration. When set to `true`, SIP Server itself controls execution of Virtual IP address control scripts through the LCA component. The names of the Application objects representing scripts are configured using the `sip-vip-script-up` and `sip-vip-script-down` options. SIP Server instructs LCA to execute the `sip-vip-script-up` option when switching to the primary mode, or the `sip-vip-script-down` option when switching to the backup mode.

convert-otherdn

Default Value: `+agentid +reserveddn +fwd`
Valid Values:

`+/-agentid`	Turns on/off the conversion of the Agent ID value provided in the `OtherDN` attribute of a T-Library request to the DN where the agent was logged in. SIP Server processes requested T-Library operations with this DN.
`+/-reserveddn`	(Does not apply for use with SIP Server)
`+/-fwd`	Turns on/off conversion of `OtherDN` in request `TSetCallForward`.

Changes Take Effect: Immediately
Related Feature: “Smart OtherDN Handling” on page 363

Defines whether SIP Server has to convert (if applicable) the value provided in `AttributeOtherDN` of a T-Library request to the DN where the agent was logged in.

For example:

The Application-level option `convert-otherdn` is set to `+agentid`. The agent with Agent ID 4040 is logged in on DN 2003. When an internal call is made by

a TMakeCall request containing OtherDN =4040, SIP Server sends the call to DN 2003 where Agent 4040 was logged in.

cos

Default Value: No default value
Valid Values: Any COS Voice over IP Service DN (service-type=cos)
Changes Take Effect: Immediately
Related Features: “Class of Service” on page 173, and “Dial Plan” on page 195

Specifies the Class Of Service (COS) DN assigned globally for all calls made from any DN on the Switch (unless otherwise defined on the DN or Agent Login-level).

Class of Service (COS) is the functionality that defines telephony capabilities for a device or an agent. This option is used in both of the dial plan-related features supported by SIP Server—Class of Service and Dial Plan. For more information about how to use this option for either functionality, see the following:

- “Class of Service” on page 173
- “Dial Plan” on page 195

Class of Service can also be assigned to the device (a DN object in SIP Server Switch configuration) or to the agent (an Agent Login object in SIP Server Switch configuration).

COS assigned to the agent takes precedence over the COS assigned to the device. That is, when different COSs are assigned to the device and to the agent, SIP Server will use the COS assigned to the agent.

If this option is not specified for both the device and logged-in agent, COS functionality is disabled for the calls to and from the device.

cpd-info-timeout

Default Value: 3
Valid Values: Any valid integer
Changes Take Effect: Immediately
Related Feature: “Outbound IP Solution Integration” on page 306

Specifies the length of time, in seconds, that SIP Server will wait for the INFO message with the CPD result. The timer starts at the moment that SIP Server receives the 200 Ok from the media gateway. If the timer expires before the CPD result is available, SIP Server generates an EventEstablished with a CallState of 0 (voice is detected). You can specify this option at both the Application and DN level (DN-level takes priority). By default, the DN-level option is initialized with the value of the Application-level option. The default value of the application-level option is 3 seconds. If the call_timeguard_timeout key is included in the Extensions attribute in the TMakePredictiveCall request, the value of this key will override the value of the cpd-info-timeout option.

To increase the probability that the timer expires after the CPD result is available, you can reduce the value of the post-connect CPD timer sent to the Genesys Media Server by using the `timeguard-reduction` option. See "CPD Performed by Genesys Media Server" on page 308 for more information about using this option.

If the post-connect CPD timeout is explicitly set to 0 by the `cpd-info-timeout` option or by the `call_timeguard_timeout` key in the `Extensions` attribute of the `TMakePredictiveCall` request, then SIP Server does not start its own post-connect CPD timer and does not send a proposed timer value to the Genesys Media Server. In this case, the Outbound solution relies on the value of the post-connect CPD timeout configured in the Genesys Media Server. This configuration is not recommended, because in case of missing post-connect CPD result notification from Genesys Media Server, SIP Server will not be able to process the call and the call can only be released by the customer.

default-dn

Default Value: `NULL`
Valid Value: Any valid DN
Changes Take Effect: On the next call

Specifies the DN to which calls are sent when URS is nonoperational, or when the timeout specified in the `router-timeout` option expires.

Notes:
- You can also use this option for emergency ACD routing.
- You can define this option at both the Application and the DN level. The DN-level option takes precedence.

default-monitor-mode

Default Value: `mute`
Valid Values:

`mute` (or `normal`)	Silent monitoring is used (supervisor connection is mute)
`coach`	Whisper coaching is used (only the monitored agent can hear the supervisor)
`connect`	The open supervisor presence is used

Changes Take Effect: Immediately
Related Feature: "Call Supervision" on page 139

Initializes a new call supervision subscription monitor mode if the `MonitorMode` extension is not provided (or if its value is specified incorrectly) in the `TMonitorNextCall` request.

default-monitor-scope

Default Value: `call`
Valid Values:

`call`	The supervisor remains on the call until it is finished.
`agent`	SIP Server disconnects the supervisor from the call automatically when the monitored agent leaves the call.

Changes Take Effect: Immediately
Related Feature: “Call Supervision” on page 139

Initializes a new call supervision subscription monitor scope if the `MonitorScope` extension is not provided (or its value is specified incorrectly) in the `TMonitorNextCall` request.

default-music

Default Value: `music/on_hold`
Valid Value: Name and path of any valid audio file
Changes Take Effect: Immediately for all new calls
Related Feature: “Customizing Music on Hold and in Queue” on page 179

Specifies the name of the file that is played for the music treatment if none is specified in `TApplyTreatment`, or if the specified file is missing.

Note: You can define this option at the Application, the DN level, and the Agent-Login level. The Agent-Login level option takes precedence.

default-route-point

Default Value: No default value
Valid Value: Any valid DN, or `reject=<SIP ERROR>`, where `<SIP ERROR>` is a three-digit number in the range of `400`–`699` representing the valid SIP `ERROR`—for example, `reject=404`
Changes Take Effect: On the next call

Specifies the DN to which an inbound call is sent when its destination number is external. For this call, SIP Server will report the call’s original destination number taken from the `Request-URI` of the `INVITE` message in the `DNIS` attribute of the `EventRouteRequest` message.

If set to `reject`, SIP Server, while applying a `default-route-point` rule, does not direct a call to a particular route point, rejecting the call instead with the configured SIP error code.

If `default-route-point` is set to `reject=<SIP ERROR>` and `default-route-point-order` is set to `before-dial-plan`, SIP Server rejects a call without applying any dial plan.

If `default-route-point` is set to `reject=<SIP ERROR>` and `default-route-point-order` is set to `after-dial-plan`, SIP Server applies a dial plan, as follows:

- If a dial plan does not fit the original target, SIP Server rejects the call.

- If a dial plan fits the original target but the modified target, after the dial plan was applied, does not match any internal resource, SIP Server rejects the call.
- If a dial plan fits the original target and the modified target, after the dial plan was applied, matches an internal resource, SIP Server sends the call to that resource.

If the `default-route-point` option is not configured (or does not have any value), then inbound calls are handled in accordance with the regular procedure.

Note: This functionality is applicable only to inbound calls initiated by incoming `INVITE` requests. This functionality is not applicable to ISCC calls or to calls initiated by T-Library requests, such as `TMakeCall`, `TRouteCall`, `TInitiateTransfer`, and so on.

default-route-point-order

Default Value: `before-dial-plan`
Valid Values: `before-dial-plan, after-dial-plan`
Changes Take Effect: On the next call

When the option is not configured or set to a value of `before-dial-plan`, SIP Server applies a `default-route-point` rule before processing the target destination according to a dial plan.

When the option is set to a value of `after-dial-plan`, SIP Server first applies the dial plan and only after that applies the `default-route-point` rule to the dial-plan result.

default-video-file

Default Value: `NULL`
Valid Values: Any valid video file codec and path
Changes Take Effect: Immediately
Related Feature: “Video Support” on page 382

Contains the name of the video file that is played to the caller if a single-step conference to the `gcti::video` device does not contain a `VideoFile`.

dial-plan

Default Value: No default value
Valid Values: Any dial-plan `Voice over IP Service` DN
Changes Take Effect: On the next call
Related Feature: “Dial Plan” on page 195

Specifies which dial-plan DN will be applied to calls. You can define the option on any of the following locations listed in order of highest to lowest priority:

1. Agent Login—Applies to calls made by a caller logged in under this `Agent Login` ID.

2. DN-level—Applies to calls made from a DN (where `Agent Login` dial-plan is undefined) or for inbound calls if the dial-plan is assigned to the `Trunk` DN.
3. Application-level—Applies to all calls (where no `Agent Login` or DN dial-plan is defined).

disable-media-before-greeting

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: “Disabling Media Before Greeting” on page 324

Specifies whether SIP Server establishes a call in hold state if greetings are configured to be played for a caller and an agent. If set to `true`, SIP Server establishes a call in hold state (an SDP to the caller and the agent is placed on hold/inactive state). If the recording is enabled, the SDP to a recorder is also placed on hold before the greeting is played. If set to `false`, SIP Server establishes the call in active state and the media is played before the greeting.

Note: This option can be configured at both Application and DN levels. Setting at the DN level takes precedence over the Application level. If this option is set at an Application level and if a particular DN does not support this functionality, this option must be explicitly set to `false` for that DN. For a DN-level activation of this feature, this option must be set for both origination and destination DNs.

disconnect-nailedup-timeout

Default Value: `0`
Valid Values: Any positive integer
Changes Take Effect: At the next nailed-up connection
Related Feature: “Nailed-Up Connections for Agents” on page 287

Specifies whether SIP Server terminates an agent’s nailed-up connection because of the agent’s inactivity. When set to a non-zero value, SIP Server waits this time interval, in seconds, before terminating the agent’s nailed-up connection. When set to `0` (the default), SIP Server does not terminate the agent connection.

Note: This option can be configured at both Application and DN levels. Setting at the DN level takes precedence over the Application level.

divert-on-ringing

Default Value: `true`
Valid Values:

`true`	SIP Server generates `EventRouteUsed` and `EventDiverted` messages when any SIP `18x` response (`180 Ringing` or `183 Session Progress`) arrives for the `INVITE` request at the routing destination.
`false`	SIP Server postpones `EventRouteUsed` and `EventDiverted` messages until the call is answered by the routing destination with a SIP `200 OK` message. If the call is not answered within the value specified by the `after-routing-timeout` option, the destination SIP dialog is canceled and an `EventError` message is generated.

Changes Take Effect: Immediately

Determines SIP Server behavior when routing calls.

drop-nailedup-on-logout

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: “Disconnecting the Nailed-Up Connection” on page 290

If enabled, on receiving a `TAgentLogout` from an agent with a nailed-up connection, SIP Sever will end the nailed-up connection to the agent DN. If the agent is currently on the call when the `TAgentLogout` is issued, SIP Server does not reconnect the agent DN to the `gcti::park` device when the current call ends, but instead ends the connection.

dr-forward

Default Value: `off`
Valid Values

`off`	Disaster Recovery (DR) peer forwarding to the peer switch is turned off. SIP Server works in the traditional single mode and always tries to deliver the call to the requested destination on the local switch.
`no-agent`	SIP Server delivers a call to a DN on the local switch when there is an agent logged in on this DN; if there is no agent logged in, SIP Server forwards the call to the peer switch. This setting applies only to DNs on “dual-registered” SIP endpoints that simultaneously register to both peers, or for DNs without any registered endpoints, such as for “remote agents.”
`oos`	SIP Server delivers a call to a DN on the local switch when the DN is in service; if the DN is out of service, SIP Server forwards the call to the peer switch. This setting applies only to DNs on “single-registered” SIP endpoints that register only to the preferred peer unless there is an error, in which case they register to the alternate peer.

Changes Take Effect: Immediately

Defines a system-wide mode of forwarding inbound and internal calls when SIP Server is operating in Business Continuity mode. This option can also be set at the DN level, in which case will override the Application level.

For both the `no-agent` and `oos` settings, SIP Server only forwards calls targeting an Extension DN, and it will only forward each call a single time; if the other peer is unable to deliver the call, an error is generated.

Notes:
- The registration timing of endpoints must be carefully considered when using the `oos` setting. For maximum responsiveness in a disaster scenario, a short registration interval must be used so the phone can quickly detect when a peer is unavailable. The deployment should be properly planned to account for the corresponding load of `REGISTER` messages.
- With the `oos` setting, if a desktop is unable to connect to the site where a SIP phone is registered, it might result in a phone registering a DN on one peer while the agent desktop connects to the other peer. Calls would be delivered to the phone, but the agent desktop would be unaware of these calls.

dr-peer-location

Default Value: `NULL`
Valid Values: A valid name of the DR peer Switch
Changes Take Effect: on the next target detection

Specifies the location of the other SIP Server in the DR pair. If set to `NULL` (the default), SIP Server is unable to support the Dial Plan feature.

dr-peer-trunk

Default Value: `NULL`
Valid Values: A valid name of a `Trunk` DN that points to the DR peer site.
Changes Take Effect: Immediately

Specifies that this SIP Server is a part of a DR pair and identifies the `Trunk` DN that points to the other SIP Server in the DR pair. If set to `NULL` (the default), SIP Server operates in the traditional single mode.

emergency-recording-cleanup-enabled

Default value: `false`
Valid values:

`true`	SIP Server automatically terminates emergency recording.
`false`	SIP Server does not terminate emergency recording.

Changes Take Effect: Immediately

Specifies whether SIP Server automatically terminates emergency recording when no internal parties remain on a call.

emergency-recording-filename

Default Value: NULL
Valid Values: Any valid file name using the variables specified below
Changes Take Effect: When the next emergency call recording is initiated

Specifies the recorded file name when emergency call recording is initiated by an agent. When this option contains a value, the generated emergency call recording file name is added as UserData to the call with the GSIP_EMRGREC_FN key. When this option does not contain a value, the recorded file name will be the UUID of the call.

The following variables are used when creating the file:

ANI:	The calling number.
$DNIS$:	The called number.
$DATE$:	The current date (GMT) in the Y-M-D format.
$TIME$:	The current time (GMT) in the H-M-S format.
$CONNID$:	The Connection ID of the call.
$UUID$:	The UUID of the call.
$AGENTID$:	The Agent Login ID, if the agent is logged in on the device where the emergency call recording is initiated.
$AGENTDN$:	The DN where the emergency call recording is initiated.

emulated-login-state

Default Value: ready
Valid Values:

ready	SIP Server distributes EventAgentReady after EventAgentLogin.
not-ready	SIP Server distributes EventAgentNotReady after EventAgentLogin.

Changes Take Effect: Immediately
Related Feature: “Emulated Agents” on page 234

When SIP Server performs an emulated agent login and the client specifies an agent work mode other than ManualIn or AutoIn, SIP Server uses this option to determine which event to distribute.

When the client specifies the agent work mode ManualIn, SIP Server distributes EventAgentNotReady after EventAgentLogin, and places the agent in the NotReady state.

When the client specifies the agent work mode AutoIn, SIP Server distributes EventAgentReady after EventAgentLogin, and places the agent in the Ready state.

This option can be set in a number of places, and SIP Server processes it in the following order of precedence, highest first. If the value is not present at the higher level, SIP Server checks the next level, and so on.

1. Agent Login object
2. DN object which represents the device
3. DN object which represents the Agent Group, such as an ACD Queue
4. SIP Server Application object

enable-busy-on-routed-calls

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: At next call

Specifies whether SIP Server plays a busy tone when routing multi-site calls to an agent through `direct-uui`. If this option is set to `false`, the destination SIP Server does not play a busy tone for calls routed to a busy agent in cases where the transaction type is `direct-uui`. SIP Server sends a negative response to the route origination site, and the call can then be routed to an available agent. If this option is set to `true`, SIP Server plays a busy tone for a routed call and then ends the call.

enable-enhanced-dialplan-handling

Setting: `TServer` section, the SIP Server Application (standalone SIP Server) or the VOIP Service DN with `service-type=sip-cluster-nodes`
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After restart
Related Feature: “Enhanced Handling of XS Requests” on page 214

When set to `true`, enables enhanced handling of Dial Plan extended service (XS) requests by SIP Server. This includes:

- Handling of various error codes sent as responses from Feature Server.
- Resending XS requests once on recoverable error responses from Feature Server.
- Setting a timeout (`xs-request-timeout`) for each dial plan request.
- Setting a timeout specific to the heartbeat requests (`xs-heartbeat-timeout`).
- Marking the URL as out of service on heartbeat failures based on the threshold set by `xs-missed-heartbeat-threshold`.
- Rejecting any XS request or switching SIP Server to backup mode when no active Feature Server URLs are available (the `switchover-on-xs-oos` option).

enable-ims

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies how SIP Server will handle `REGISTER` requests and populate SIP headers for this DN.

- `false`—For non-IMS endpoints (local to SIP Server) that are registered to SIP Server. SIP Server will not include IMS-related headers in SIP messages sent to all DNs (unless enabled on the DN).

- `true`—For IMS endpoints registered to the IMS-CN (using `REGISTER` requests through the third-party IMS registration). SIP Server communicates with its DNs through the S-CSCF, and adds IMS-specific headers to all SIP messages (applies globally—you cannot disable on a per-DN basis).

SIP Server considers this option enabled if set to `true` on either the Application or DN-level. If enabled on the Application, you cannot disable locally on a per-DN basis. Similarly, if enabled on a particular DN, it stays enabled despite the Application-level setting.

For IMS environments, you must either enable globally for all DNs, or set to `true` for all IMS endpoints, as well as for the `Trunk` DN used for routing to IMS.

enable-iscc-dial-plan

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: On the next call
Related Feature: “Dial Plan” on page 195

Specifies whether SIP Server applies the dial plan to the agent destination of multi-site (ISCC) calls that are routed through an External Routing Point (`cast-type=route-notoken`), as follows:

- If set to `true`, the dial plan (full, including the digit translation and forwarding rules) is applied.
- If set to `false`, the dial plan is not applied.

This option must be configured on the remote (destination) site. SIP Server applies the dial plan when a call is routed from an External Routing Point to a DN at the destination site.

Note: SIP Server will still apply the dial plan to the External Routing Point destination of multi-site (ISCC) calls, and this will take priority over the agent DN destination dial-plan rule regardless of the setting of `enable-iscc-dial-plan`.

enable-legacy-reporting

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

Starting with version 8.1.102.13, in single-site routing scenarios, SIP Server generates `EventReleased` with `AttributeCallState=7 (NoAnswer)` for an unanswered party when the `after-routing-timeout` expires. Prior to version 8.1.102.13, SIP Server generated `EventReleased` with `AttributeCallState=22 (Redirected)`.

This option, `enable-legacy-reporting`, enables backward compatibility for reporting `AttributeCallState` that SIP Server distributes in `EventReleased` for an unanswered routing target party in single-site routing scenarios.

If set to `true`, SIP Server distributes `EventReleased` with `AttributeCallState=22 (Redirected)`.

If set to `false`, SIP Server distributes `EventReleased` with `AttributeCallState=7 (NoAnswer)`.

enable-outbound-ext-dial-plan

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: For the next outbound scenario

Enables use of the external dial plan for outbound calls triggered by a `TMakePredictiveCall` request on a Trunk Group DN or Routing Point.

enable-retransmit-on-oos-transport

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

When this option is set to `true`, and SIP Server detects that a Trunk DN for an inbound call is out of service, SIP Server continues retransmission of the SIP request using the transport associated with the dialog, even though the DN is detected as out of service. When this option is set to `false`, SIP Server does not retransmit the request and sends the timeout to the application layer when the DN is detected as out of service.

This option is applicable only when the UDP transport is used.

Note: This option can be configured at both Application and DN levels. Setting at the DN level takes precedence over the Application level.

enable-strict-location-match

Default Value: No default value (empty string)
Valid Values: `msml (or true), softswitch, trunk, all`
Changes Take Effect: On the next call
Related Feature: “Geo-Location for MSML-Based Services: Strict Matching” on page 393

Controls the SIP Server behavior in cases where an MSML service that matches a call by geo-location or overflow-location is not available, or, if during an attempt to apply a treatment, the matching service responds to the `INVITE` message with a SIP error, as follows:

- If this option is not present or not configured, SIP Server tries other available services for a call.

- If this option is set to `msml` (or `true`), SIP Server tries other available services that match a call by geo-location or overflow-location. If there is no match, SIP Server does not apply a service to the call with a different geo-location. (A value of `true` is supported for compatibility with previous releases of this feature.)
- If this option is set to `trunk` or `softswitch`, SIP Server tries other available trunks or softswitches that match a call by geo-location. This applies to calls directed to an external destination or DNs located behind the softswitch. If there is no match, SIP Server does not send a call to a device with a different geo-location.
- If this option is set to `all`, SIP Server applies the `msml` setting for calls to GVP and the `trunk/softswitch` setting to other cases.

Note: If the `enable-strict-location-match` option is set to `msml` or `true`, it is possible to specify an alternative geo-location using the Application level option `overflow-location-map`, or using the `overflow-location` key in `AttributeExtensions` of `TRouteCall` and `TApplyTreatment` client requests.

enable-unknown-gateway

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether SIP Server will accept or reject calls arriving from a gateway that is not represented as a `Trunk` DN in the Configuration Layer. To accept calls from un-represented gateways, set this value to `true`. To reject calls from un-represented gateways, set this option to the default `false`.

encoding

Default Value: No default value
Valid Values: See the `ICU Home > Converter Explorer` pages for values (http://demo.icu-project.org/icu-bin/convexp)
Changes Take Effect: After SIP Server restart
Related Feature: “Mapping SIP Headers and SDP Messages” on page 261

Provides Unicode support for the SIP-to-TLib and TLib-to-SIP mapping functionality. If the option value is not specified, a default local character set is used for conversion. If the option value is specified, the character set that is specified by this option is used for conversion. To change the default, set this option to the name of a converter that can translate UTF-8 data to the local character set. The converter suitable for a particular deployment can be found using the ICU Converter Explorer.

If enabled, SIP Server can convert UTF-8 encoded data received in SIP messages to a local character set. Reverse conversion (from a local character set to UTF-8) is performed by sending data from T-Library messages encoded in a local character set to the remote destination using SIP messages.

encoding-area

Default Value: No default value
Valid Values: A list of areas separated by a comma (,) where encoding will be applied. Supported areas:

- `tlibsip`—For the "Mapping SIP Headers and SDP Messages" on page 261 feature
- `chat`—For the "Instant Messaging" on page 250 feature

Changes Take Effect: Immediately

Specifies the list of areas where encoding applies.

enforce-1pcc-inbound

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call
Related Feature: "Treating Incoming Calls As Inbound Calls" on page 369

When set to `true`, SIP Server treats 1pcc/incoming calls from external callers as inbound calls. A call is considered internal if both conditions are met:

- A username in the `From` header matches the Extension DN configured in the SIP Switch.
- A network address of the caller in the first `Via` header matches the IPv4 CIDR blocks or FQDN listed in the `internal-call-domains` option.

If the `internal-call-domains` option is empty, all incoming calls are treated as inbound calls.

enforce-external-domains

Default Value: `NULL`
Valid Values: A list of computer names or IP addresses that are external to SIP Server. The list can be separated by semicolons (;).
Changes Take Effect: Immediately

When a value is configured, SIP Server checks the list of computer names or IP addresses against the computer names or IP addresses specified in the URI of the `From` header. If there is a match, then the DN is considered external.

When a value is not configured, SIP Server uses the user part of the URI only to find the device.

Note: You must include in the value of this option the computer names or IP addresses of SIP gateways or hosts associated with other T-Servers so that SIP Server may communicate with over ISCC. In certain configurations, you may also have to configure the option `override-domain-from`.

enforce-trusted

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: On the next call
Related Feature: "Network Asserted Identity" on page 292

Specifies the default trust-level for all DNs on the SIP Server switch. For backward compatibility with IMS deployments (where all entities within the IMS deployment are trusted by default), all DNs on the SIP Server switch will be considered trusted by default, unless otherwise specified at the DN-level. If you set this option to `false`, then all DNs on the switch will be considered non-trusted, unless otherwise specified at the DN-level

This option is also used to enable SIP Server to pass following private headers:

- `P-Early-Media`—See "Early Media Private Header" on page 193.
- `P-Access-Network-Info`—See "P-Access-Network-Info Private Header" on page 319.

Note: To establish the trust-level for individual DNs, see the DN-level option `enforce-trusted` on page 581.

enhanced-pending-acw

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

When set to `true` and when SIP Server receives `RequestAgentNotReady` with `AttributeExtensions` containing `WrapUpTime` set to `untimed` during a consultation call, SIP Server distributes `EventAgentNotReady` when releasing the call.

event-ringing-on-100trying

Default Value: `false`
Valid Values:

`true`	SIP Server generates `EventRinging`.
`false`	SIP Server does not generate `EventRinging`.

Changes Take Effect: Immediately

Specifies whether SIP Server generates an `EventRinging` message for a DN when it receives a `100 Trying` SIP message. Normally, the `EventRinging` message is generated on a `180 Ringing` SIP message, but this option allows for GVP integration when the IVR Server is configured in Behind-the-Switch mode.

Note: This option must be set at both the `Application` and at the `DN` level because it is used for proper synchronization with the I-Server `Application`.

external-registrar

Default Value: NULL
Valid Values: String conforming to the SIP-URI syntax of RFC 3261, defined as: sip:[userinfo]hostport uri-parameters[headers]
Changes Take Effect: Immediately

Specifies the location of an external registrar service. SIP Server implements very limited registrar functionality to support clients that can only register dynamically (for example, Microsoft Messenger 4.7–5.1). Such clients must be configured in the Configuration Layer as DNs. Depending on the state of the internal SIP Server registrar, registration subscriptions from either all, or not configured clients are forwarded to the external registrar.

For example: sip:192.168.8.100:5090;transport=tcp

If no external registrar is specified, a 503 Service Unavailable error is returned for the REGISTER method.

extn-no-answer-overflow

Default Value: none
Valid Values:

none	SIP Server does not attempt to overflow a call on an extension when the time specified in the extn-no-answer-timeout option expires.
recall	SIP Server returns the call to the last distribution device (the device reported in the ThisQueue attribute of the call) when the time specified in the extn-no-answer-timeout option expires.
release	SIP Server releases the call.
Any valid overflow destination	SIP Server returns the call to the specified destination when the time specified in the extn-no-answer-timeout option expires.

Changes Take Effect: Immediately
Related Feature: “No-Answer Supervision” on page 302

Specifies a sequence of overflow destinations (separated by a comma) that SIP Server attempts to overflow to when the time specified in option extn-no-answer-timeout has expired. SIP Server attempts to overflow in the order specified in the list.

When all overflow attempts fail, SIP Server abandons overflow. See also the NO_ANSWER_OVERFLOW extension in section “Using the Extensions Attribute” on page 414 for more information about how this option is used.

When the list of overflow destinations contains the recall value and the call was not distributed, SIP Server skips to the next destination in the list.

SIP Server obtains the value for this option in the following order:

1. no-answer-overflow if defined at a DN level of type Extension and applies when agents logged out.
2. extn-no-answer-overflow if defined at a SIP Server Application level.

extn-no-answer-timeout

Default Value: 15
Valid Value: Any integer from 0–600
Changes Take Effect: Immediately
Related Feature: "No-Answer Supervision" on page 302

Defines the default no-answer timeout (in seconds) that SIP Server applies to any device of type Extension. When the timeout ends, SIP Server executes the actions defined in option extn-no-answer-overflow.

If set to 0, the No Answer Supervision feature for DNs of type Extension is disabled. See the NO_ANSWER_TIMEOUT extension in section "Using the Extensions Attribute" on page 414 for more information about how this option is used.

SIP Server obtains the value for this option in the following order:

1. no-answer-timeout if defined at a DN level of type Extension and applies when agents logged out.
2. extn-no-answer-timeout if defined at a SIP Server Application level.

fast-busy-tone

Default Value: music/atb_5sec
Valid Values: Name and path of any valid audio file
Changes Take Effect: Immediately for all new calls

Specifies the audio file to be played for the FastBusy treatment.

fax-detected

Default Value: drop
Valid Values:

drop	The call is released.
connect	The connected call remains connected.

Changes Take Effect: Immediately
Related Feature: "Outbound IP Solution Integration" on page 306

Specifies the behavior of SIP Server where CPD is operating and a fax machine is detected on an outbound call. SIP Server provides the CPD result in UserData attached to the call as a key-value pair with key AnswerClass containing the value Fax. This UserData in EventRouteRequest provides extra information to the strategy, so that the strategy can decide to drop the Fax call if required.

feature-code-park

Default Value: `10`
Valid Values: A two-digit integer
Changes Take Effect: Immediately
Related Feature: “Call Park/Retrieve” on page 119

Specifies the number part of the star code used to initiate a transfer to the internal `gcti::pbxpark` device, for PBX Call Park functionality. For example, if you set this option to the default value of `10`, the agent will dial `*10` to park the current call at the internal `gcti::pbxpark` device, in order to retrieve it later.

feature-code-pickup

Default Value: `12`
Valid Values: A two-digit integer
Changes Take Effect: Immediately
Related Feature: “Call Pickup” on page 120

Specifies the number part of the star code to be dialed to pick up a call ringing at another device, for PBX Call Pickup functionality.

feature-code-retrieve

Default Value: `11`
Valid Values: A two-digit integer
Changes Take Effect: Immediately
Related Feature: “Call Park/Retrieve” on page 119

Specifies the number part of the star code used to retrieve a parked call from the internal `gcti::pbxpark` device. For example, if you set this option to the default value of `11`, and the DN from which the call was parked is `1001`, the agent will dial `*11`, plus the DN number `1001` to retrieve the call: `*11 1001`.

find-outbound-msml-by-location

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When set to `true`, SIP Server selects an MSML service for outbound calls based strictly on a call's geo-location. This option applies only to a call initiated by the `TMakePredictiveCall` request on behalf of the Routing Point DN. When set to `false`, this feature is disabled.

find-trunk-by-location

Default Value: `false`
Valid Values:

`true` SIP Server considers the `geo-location` option setting when prioritizing the selection of an outbound gateway or trunk.

When making a selection, SIP Server first narrows the pool of available in-service gateways or trunks based on the prefix match, then further narrows the pool by matching the value of the `geo-location` option of the DN to the value of the `geo-location` option for the `Trunk` device. If more than one matching trunk is found, SIP Server can further narrow the selection by considering the value of the `priority` option (if it is configured). If no matching trunk is found, SIP Server selects a trunk as if `find-trunk-by-location` is set to `false`.

`false` SIP Server does not include the `geo-location` option setting when prioritizing the selection of an outbound gateway or trunk.

SIP Server selects a gateway or trunk from the pool of all configured trunks that are in service based on the prefix match, disregarding the `geo-location` option (if the `priority` option is configured, it will still be considered). If there is more than one trunk in the pool, SIP Server chooses the trunk in a round-robin algorithm that provides equal gateway load. However, if an external party is transferred to an outbound destination, the same gateway that connected the external party to the call is used for the outbound transfer.

Changes Take Effect: On the next call

Determines SIP Server behavior for choosing a gateway or trunk for the outbound call.

fmfm-confirmation-digit

Default Value: `0`
Valid Values: `0-9`
Changes Take Effect: On the next call
Related Feature: “Find Me Follow Me” on page 243

Specifies the digit that a caller must enter for call confirmation. This digit could be included in the prompt to be used for human recognition. If used, this digit must match the digit in the recorded prompt file. To use a different digit, you must record a new prompt and place the file in the `MCP folder/users` folder on the Media Control Platform server host.

fmfm-confirmation-timeout

Default Value: `10`
Valid Values: A positive number
Changes Take Effect: On the next call
Related Feature: "Find Me Follow Me" on page 243

Specifies the timeout value, in seconds, that SIP Server waits for a confirmation digit to be entered. Enter a number that includes playing time of the confirmation prompt and time for the confirmation digit to be entered.

Note: A call is considered abandoned when: the caller hangs up, the entered digit does not match the value of the `fmfm-confirmation-digit` option, or the call times out with no input at all.

fmfm-prompt-file

Default Value: Any empty string
Valid Values: A valid filename
Changes Take Effect: On the next call
Related Feature: "Find Me Follow Me" on page 243

Specifies the filename of the confirmation prompt. Must match the path and filename in the MCP folder/users folder on the Media Control Platform server host. For example: for the file `users/fmfm-confirmation-prompt-0.wav`, set `fmfm-prompt-file` to `fmfm-confirmation-prompt`.

fmfm-trunk-group

Default Value: An empty string
Valid Values: A valid Trunk Group DN name
Changes Take Effect: On the next call
Related Feature: "Find Me Follow Me" on page 243

Specifies the Trunk Group DN where events are generated, when each destination leg connects to Media Server. Enter a Trunk Group DN name that represents Media Server. SIP Server uses that DN to play ringback, and for all outbound calls to Find Me Follow Me destinations.

forced-notready

Default Value: `true`
Valid Values:

`true`	The desktop is forced into the `Not Ready` state.
`false`	The desktop is not forced into the `Not Ready` state.

Changes Take Effect: Immediately for all future calls

Determines whether the desktop is forced into a `Not Ready` state when it does not respond after a `Preview Interaction` dialog box has been displayed on the desktop.

Note: This option works with the `preview-interaction` and `preview-expired` options to determine what action to take when a desktop does not respond to a preview interaction before the time expires.

force-p-early-media

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether SIP Server adds the `P-Early-Media` header into requests and responses, even if the `P-Early-Media` header with the supported value is not present in the initial `INVITE` message.

Note: SIP Server supports the `P-Early-Media` header in the following messages: `18x`, `200`, `INVITE`, `PRACK`, and `UPDATE`.

graceful-shutdown-sip-timeout

Default Value: `4`
Valid Values: `0`–`32`
Changes Take Effect: Immediately

Specifies the timeout, in seconds, during which SIP Server re-transmits the `BYE` requests that were not confirmed with `200 OK` responses. The timeout starts as soon as the last call is ended. If set to `0` (zero), no `BYE` requests are re-transmitted. The timeout applies only when SIP Server processes the graceful shutdown.

greeting-after-merge

Default Value: `false`
Valid Values: `false, true`
Changes Take Effect: Immediately
Related Feature: “Personal Greetings” on page 319

If this option is set to `false`, a greeting will not be played to an agent or customer when a transfer or conference is completed (8.0.2 backward compatibility support). If the option is set to `true`, a greeting will be played to an agent or customer when a transfer or conference is completed.

greeting-call-type-filter

Default Value: No default value
Valid Values: `internal, consult, outbound`
Changes Take Effect: Immediately
Related Feature: "Personal Greetings" on page 319

Specifies—using a space-, comma-, or semicolon-separated list—the types of calls to which a greeting will not be played. By default (the option has no value), a greeting will be played to all calls (for 8.0.2 backward compatibility). If the option is set to `internal, consult,` and/or `outbound`, a greeting will not be played to internal, consultation, and/or outbound calls, respectively.

Note: You can define this option at both the Application and the Agent-Login level. The Agent-Login level option takes precedence.

greeting-delay-events

Default Value: `false`
Valid Values: `false, true`
Changes Take Effect: Immediately
Related Feature: "Personal Greetings" on page 319

Controls the distribution order of `EventOffHook` and `EventEstablished` messages when an agent or customer greeting is played. If the option is set to `false`, SIP Server sends `EventOffHook` and `EventEstablished` before the agent or customer greeting starts playing (8.0.2 backward compatibility support). If the option is set to `true`, SIP Server sends `EventOffHook` and `EventEstablished` when the agent or customer greeting ends. It might cause SIP Server to release the call on an agent DN in the middle of the greeting if `greeting-delay-events` is set to `true`. See `greeting-stops-no-answer-timeout` for more information.

greeting-notification

Default Value: An empty string
Valid Values: `started`, `complete` (one or both, in any order)
Changes Take Effect: Immediately
Related Feature: "Personal Greetings" on page 319

Specifies—using a space-, comma- or semicolon-separated list—whether SIP Server sends notifications when a greeting starts or ends. The default value (an empty string) means that SIP Server will not send any notification when a greeting starts or ends. If the option is set to `started`, SIP Server will send an `EventPrivateInfo` message with `AttributePrivateMsgID` set to `4012` when the greeting starts. If the option is set to `complete`, SIP Server will send `EventPrivateInfo` with `AttributePrivateMsgID` set to `4013` when the greeting ends.

greeting-repeat-once-party

Default Value: `agent`
Valid Values: `agent, customer`
Changes Take Effect: Immediately
Related Feature: "Personal Greetings" on page 319

If this option is set to `agent`, a customer greeting is played continuously until the agent greeting finishes playing (8.0.2 backward compatibility support). If the option is set to `customer`, an agent greeting is played continuously until the customer greeting finishes playing.

greeting-stops-no-answer-timeout

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call
Related Feature: "Personal Greetings" on page 319

(Introduced in SIP Server 8.1.101.86) Set this option to `true` in environments where both No-Answer Supervision and Personal Greeting functionality are configured. In this case, SIP Server stops the no-answer timer as soon as a `200 OK` SIP response is received, indicating that the destination party has answered the call. SIP Server does not apply the no-answer action and no-answer overflow to the call even if they are configured in the corresponding options.

The default value of `false` is required to preserve the original SIP Server behavior and provide backward compatibility. In this case, SIP Server does not stop the no-answer timer until `EventEstablished` is generated for the destination party. It might cause SIP Server to release the call on an agent DN in the middle of the greeting if `greeting-delay-events` is set to `true`. `EventEstablished` on the agent DN is generated only when the greeting is finished.

ha-max-calls-sync-at-once

Default Value: `500`
Valid Values: `200-1000`
Changes Take Effect: When the HA connection is established
Related Feature: "Enhanced Procedure for Upgrading of SIP Server HA Pair" in the *SIP Server High-Availability Deployment Guide*

Specifies the maximum number of calls that can be synchronized at once between the primary SIP Server and the backup SIP Server after the HA link connection is established, before waiting for 1 second to continue with synchronization. Only calls that are missing on the backup SIP Server are synchronized.

hide-msml-location

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When set to `true`, SIP Server does not include `X-Genesys-geo-location` and `X-Genesys-strict-location` headers in `INVITE` messages that it sends to GVP. Removal of these headers allows SIP Server to stay in control of geo-location selection. This configuration option can be used when all MCPs controlled by Resource Manager are deployed at the same location and Resource Manager does not need to consider geo-location in the MCP selection process.

When set to `false`, this feature is disabled.

http-port

Default Value: `0`
Valid Values: `0, 1024-65535`
Changes Take Effect: After SIP Server restart
Related Feature: “HTTP Monitoring Interface” on page 245

Specifies the HTTP interface port number. When set to `0`, the HTTP server is disabled. The port numbers in the range of `1` through `1023` are the system ports and must not be used.

ignore-presence-after-nas

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: On the next call

Specifies whether SIP Server processes or ignores presence SIP messages to change an agent state to `Ready` if the no-answer action is set to `notready` for an agent.

If set to `true`, SIP Server ignores presence SIP messages if the no-answer action is set to `notready` for an agent.

If set to `false`, SIP Server processes presence SIP messages if the no-answer action is set to `notready` for an agent.

Note: You can define this option at both the Application and the DN levels. The DN-level option takes precedence.

---

internal-call-domains

Default Value: An empty string
Valid Values: A list of IPv4 CIDR blocks or FQDN separated by semicolons (;). The IP address without a wildcard means the host address—for example, "1.2.3.0" means "1.2.3.0/32".
Changes Take Effect: On the next call
Related Feature: “Treating Incoming Calls As Inbound Calls” on page 369

If the enforce-1pcc-inbound option is set to true and the internal-call-domains option is set to a list of IP addresses, SIP Server does the following for the incoming calls:

1. SIP Server verifies the Via header of the INVITE message against the value of the internal-call-domains option:
   - If a match is found, SIP Server proceeds to Step 2.
   - If a match is not found, SIP Server treats the call as inbound.
2. SIP Server verifies only the username part in the From header in the INVITE message against the internal DNs and classifies the calls as follows:
   - If the username matches an Extension or ACD Position DN, SIP Server treats the call as internal.
   - If the username matches a Routing Point or Trunk Group DN, SIP Server rejects the call.
   - If a match is not found, SIP Server treats the call as inbound.

All other 1pcc/incoming calls are treated as inbound calls. If the option is empty, all 1pcc/incoming calls are treated as inbound calls.

Note: IPv6 addresses are not supported in the list of the internal-call-domains option.

ims-3pcc-prefix

Default Value: No default value
Valid Values: A string
Changes Take Effect: Immediately

Notes:
- In order for this option to take effect, the following additional options must also be configured:
  - enable-ims (set to true)
  - override-domain
- For alternative deployment with Mediation Proxy, the value of this option should be 3pcc. (note the period at the end of this value).

If both ims-3pcc-prefix and override-domain are configured, then for all 3pcc INVITE requests (INVITE requests resulting from 3pcc T-Library operations), SIP Server will add the value of the ims-3pcc-prefix option after the user-part of the URI, but before the override-domain.

For example, if override-domain is set to genesys.com, and ims-3pcc-prefix is set to 3pcc, then 1pcc calls will use the URI: sip+1234@genesys.com

while 3pcc calls will use the URI: sip:+1234@3pcc.genesys.com

ims-default-orig-ioi
ims-default-icid-prefix
ims-default-icid-suffix

Default Value: No default value (empty string)
Valid Value: Any sub-string valid for this part of the P-Charging-Vector
Changes Take Effect: Immediately

Specifies the values that SIP Server uses when generating the P-Charging-Vector header for calls originated by SIP Server.

SIP Server uses these option settings to generate the value of the header as follows:
icid-value="<ims-default-icid-prefix><uniq-value-generated-by-sip-server><ims-default-icid-suffix>"; orig-ioi=<ims-default-orig-ioi>

For example, the settings

ims-default-orig-ioi=genesyslab.com

ims-default-icid-prefix=prefix-

ims-default-icid-suffix=-suffix

result in the following ICID value

icid-value="prefix-23AAFD4901493123-suffix";orig-ioi=genesyslab.com

ims-propagate-pcvector

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

Enables propagation of the P-Charging-Vector value. Set this option to true and SIP Server includes the value of the P-Charging-Vector from the main call in the private header of the INVITE that it sends to the consultation call. This option setting is used to provide a single consistent P-Charging-Vector for all legs involved in the consultation call.

For RFC 3455-compliant behavior (a unique ICID for each SIP dialog), set this option to false. SIP Server will create a unique P-Charging-Vector value for the new dialog for the consultation call. This option setting is used for outbound calls, as well as inbound calls to IMS-registered agents. In addition to setting this option to false, you must also configure the following additional options, which are used to generate the P-Charging-Vector value for calls originated by SIP Server:

- ims-default-orig-ioi
- ims-default-icid-suffix
- ims-default-icid-prefix

ims-puid-domain

Default Value: No default value
Valid Values: A string containing a valid domain
Changes Take Effect: After restart

Specifies the domain for the IMS-enabled device that SIP Server uses to generate the PUID for the `P-Asserted-Identity` SIP header, in the case of 3pcc calls initiated on behalf of devices without 3rd party registration. The PUID for the `P-Asserted-Identity` header is formed using one of the following methods (listed in order of highest to lowest priority):

- For 1pcc calls, PUID is taken from the corresponding header of the incoming `INVITE`. Since this message contains the PUID of the caller, it is preserved regardless of the destination changes.
- The option `p-asserted-identity` is configured on the caller DN.
- Address of Record (AOR) of the caller is taken from 3rd party registration (not applicable to ItalTel).
- Combination of caller's DN digits and the value of the `override-domain` option. Included for backward compatibility. Not to be used for IMS deployments.
- Combination of caller's DN digits and the value of the `ims-puid-domain` option.

ims-route

Default Value: No default value
Valid Value: Any valid SIP URI
Changes Take Effect: Immediately

Specifies the value of the `Route` header for calls to IMS destinations that are not registered with SIP Server.

ims-sip-domain

Default Value: No default value
Valid Values: A string containing a valid domain
Changes Take Effect: After restart

Specifies the domain for the IMS-enabled device that SIP Server uses in the `Request-URI` and standard SIP headers (`From` and `To`), when this information is unavailable in the initial `INVITE` or 3rd party registration. In addition, when IMS changes the `Request-URI` in re-`INVITE` requests (changes upper-case letter in the URI to lower-case letters), SIP Server uses this modified re-`INVITE`.

The URI for standard SIP headers is formed using one of the following methods (listed in order of highest to lowest priority):

- For 1pcc calls, URI is taken from `Request-URI` of the initial `INVITE` arriving at SIP Server (with user part optionally modified by the Dial Plan, if specified). Not applicable to calls routed or redirected to another device.

- Combination of DN digits and the value set for the `override-domain` option (for the `Request-URI` and `To` fields) or `override-domain-from` option (for the `From` field). Included for backward compatibility. Not to be used for IMS deployments.
- Address of Record (AOR) taken from 3rd party registration (not applicable to ItalTel).
- Combination of DN digits and the value of the `ims-sip-domain` option.

ims-sip-params

Default Value: none
Valid Values: A string containing valid URI parameters
Changes Take Effect: After restart

Specifies a valid string of SIP URI parameters separated by a semicolon. Parameters are added to the SIP URIs formed by the SIP Server as explained in the description of the `ims-sip-domain` option. Parameters specified in the `ims-sip-params` option are only used for the devices where IMS is enabled.

ims-skip-ifc

Default Value: No default value
Valid Value: Any alphanumeric string
Changes Take Effect: Immediately

Specifies the value of the `X-Genesys-SkipFC` private header used to prevent "double-dipping" for internal calls in an IMS environment. The value must match the condition configured in the Initial Filter Criteria (IFC) for Public IDs served by a particular SIP Server.

ims-use-term-legs-for-routing

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Routed Calls as Originating or Terminating" on page 248

For use in IMS environments only. When set to `true`, SIP Server uses a call-terminating leg to route calls on behalf of the Routing Point after a treatment is applied. When set to `false`, SIP Server uses a call-originating leg to route calls after a treatment is applied.

inbound-bsns-calls

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Business-Call Handling" on page 235

Specifies whether SIP Server considers all established inbound calls on an agent as business calls.

inbound-trunk-hint-sip-field

Default Value: No default value
Valid Values: A string
Changes Take Effect: On the next call

Defines which SIP header (for example, `X-CarrierID`) in a SIP `INVITE` message SIP Server uses for matching against the value configured in the DN-level `inbound-trunk-hint` option set on the `Trunk` DN.

- If the `inbound-trunk-hint-sip-field` option is not configured, SIP Server works in backward compatibility mode without applying any additional rules for an inbound trunk selection.
- If the `inbound-trunk-hint-sip-field` option is configured, SIP Server looks for a particular SIP header in an incoming `INVITE` message to find the best suitable trunk among inbound trunks.
- If the `inbound-trunk-hint-sip-field` option is set to an asterisk (*) as a wildcard, SIP Server does not look for a particular SIP header in the incoming `INVITE` message but still gives preference for selecting trunks configured with the `inbound-trunk-hint` option.

info-pass-through

Default Value: No default value
Valid Values:

`*`	Allows all `INFO` messages to be sent to the peer connection.
`-`	Disables all `INFO` messages from being sent to the peer connection.
`<list>`	Allows only those `INFO` messages specified in a comma-separated list of `Content-Type` values (used to define `INFO` messages) to be sent to the peer connection.
`*,<list>`	Allows all `INFO` messages to be sent to the peer connection, except those included in a comma-separated list of `Content-Type` values.

Changes Take Effect: Immediately

Specifies which SIP `INFO` messages SIP Server will pass to a remote device. You can use this option to allow all `INFO` messages pass through to the peer connection, to disable all `INFO` messages from being sent to the peer connection, or to specify only those `INFO` messages, as defined by the `Content-Type` header, that SIP Server will allow. By default, this option is left undefined. In this case, SIP Server passes all `INFO` messages to the peer connection, except for the following:

- `application/vnd.radisys.msml+xml`
- `application/x-www-form-urlencoded`
- `application/x-detect` (not sent if a predictive call is in progress, or if no AudioCodes CPD result header is found)
- `application/dtmf-relay`

Note: You can define this option at both the Application and the DN levels. The DN-level option takes precedence.

inherit-bsns-type

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Business-Call Handling" on page 235

Determines whether a consultation call that is made from a business primary call contains the `business call` attribute.

init-dnis-by-ruri

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

If this option is set to `true`, SIP Server determines what value must be reported as the `DNIS` attribute in T-Library messages for inbound or 1pcc calls, in the following order:

1. A username from the Request-URI—if the extracted value is not empty or "Anonymous" (case insensitive), then this value is used for the `DNIS` attribute. If the value is empty or "Anonymous", SIP Server goes to Step 2.
2. A username from the `To` header—if the extracted value is not empty or "Anonymous" (case insensitive), then this value is used for the `DNIS` attribute. If the value is empty or "Anonymous", SIP Server goes to Step 3.
3. A DN name—the destination device that will be used by SIP Server to send an `INVITE`. It can be a DN name of the `Extension` if the destination device is resolved to an extension, or a DN name of the `Trunk`, if the destination device is behind the trunk.

If this option is set to `false` (the default), SIP Server takes the username from the `To` header as a value for the `DNIS` attribute.

internal-bsns-calls

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Business-Call Handling" on page 235

Determines whether SIP Server considers internal calls made to any agent as business calls.

internal-registrar-domains

Default Value: `NULL`
Valid Values: Any valid computer names separated by a semicolon (;)
Changes Take Effect: Immediately. Existing subscriptions remain valid.

Specifies the list of logical computer names, registration subscriptions from the endpoints of which are handled by the internal registrar versus an external registrar. For example, if DN `4813` is configured, DN `4814` is not configured,

the internal registrar is enabled, and `internal-registrar-domains` is set to `world`, then:

- `REGISTER` from `4813@world` is accepted.
- `REGISTER` from `4813@galaxy` is forwarded to the external registrar.
- `REGISTER` from `4814@world` is rejected with `404 Not Found`.

internal-registrar-enabled

Default Value: `true`
Valid Values:

`true`	SIP Server's internal registrar is enabled.
`false`	All registration subscriptions are proxied to external registrar (see the `external-registrar` option).

Changes Take Effect: Immediately. Existing subscriptions remain valid.

Specifies whether the internal registrar is enabled. When this option is set to `false`, a `503 Service Unavailable` error is returned for the `REGISTER` method.

internal-registrar-persistent

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

Enables SIP Server to update the DN attribute `contact` in the configuration database. When an endpoint registers, SIP Server takes the `contact` information from the `REGISTER` request and updates or creates a key called `contact` in the `Options/Annex` tab of the corresponding DN.

Set this option to `true` only if Hot Standby HA is not used in your environment.

Note: SIP Server must have `Full Control` permission for the `DN` objects in order to update a configuration object. By default, it does not have this permission. You need to grant `Full Control` permission for the `System` account for the all DNs on the corresponding Switch. It is done for all DNs at once by changing the permissions for the `System` account on the `DN` folder in the `Switch` object. Or, you can start SIP Server under another account that has `Change` permission on the necessary DNs.

intrusion-enabled

Default Value: `true`
Valid Values:

`true`	SIP Server invites the supervisor to the current call.
`false`	SIP Server does not invite the supervisor to the current call. Instead, SIP Server will wait for the next call on the monitored agent's DN to invite the supervisor.

Changes Take Effect: Immediately
Related Feature: "Call Supervision" on page 139

Determines SIP Server behavior when a `TMonitorNextCall` request is submitted while the monitored agent is on a call.

keep-mute-after-conference

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When this option is set to `true` and a third member of the conference is released, the muted party of the call must use the `TSetMuteOff` request to restore the voice path with a caller. When this option is set to `false`, the voice path with a caller is restored automatically when a third member of the conference is released.

legal-guard-time

Default Value: `0`
Valid Value: Any integer from `0`–`30`
Changes Take Effect: Immediately
Related Feature: "Emulated Agents" on page 234

Specifies a legal-guard time (in seconds) for agents to postpone the transition to the `Ready` state after a business call or after timed ACW. SIP Server always considers a routed call as a business call. The default value of `0` (zero) disables the functionality of this option.

logout-on-disconnect

Default Value: `true`
Valid Values:

`true`	The `EventLogout` message is distributed as soon as the client that requested the login disconnects from SIP Server or unregisters the DN in question. The `EventLogout` message is distributed when SIP Server distributes `EventOutOfService`.
`false`	The `EventLogout` message is not distributed.

Changes Take Effect: Immediately

Specifies how the `EventLogout` message is distributed.

Note: This option is not applicable if the `agent-logout-on-unreg` option is set to `true`.

---

logout-on-out-of-service

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether SIP Server sends `EventAgentLogout` when the agent device goes out of service. If set to `true`, SIP Server sends an `EventAgentLogout` if the device on which the agent is logged in goes out of service. If set to `false`, SIP Server does not send the `EventAgentLogout`.

Note: When SIP Server is operating in Business Continuity mode, if a DN registration expires, SIP Server sends `EventAgentLogout` regardless of the `logout-on-out-of-service` option.

make-call-alert-info

Default Value: `NULL`
Valid Value: Any string
Changes Take Effect: Immediately

The contents of this field are passed in the `Alert-Info` header of the `INVITE` message sent to the origination party in response to any of the following requests:

- `TMakeCall`
- `TInitiateTransfer`
- `TInitiateConference`

This is used to enable a distinctive ringtone or auto-answer on the originating party's endpoint.

For example, setting this field to `<file://Bellcore-dr3>` turns on a triple ring on Cisco 7940 endpoints.

map-sip-errors

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Control of SIP Response Code from within Routing Strategy" on page 177

Specifies whether a T-Library or SIP error code is reported in the `ErrorCode` attribute of an `EventError` message in response to a `TRouteCall` request. If set to `false`, the SIP status code is reported instead of the T-Library error code. For example, if routing is made to a busy destination, SIP Server will report the `ErrorCode` attribute of the `EventError` message as `486`.

max-parking-time

Default Value: `0`
Valid Values: Any positive integer
Changes Take Effect: Immediately
Related Feature: "Call Park/Retrieve" on page 119

Specifies the timeout, in seconds, after which a call parked on the `gcti::pbxpark` device will return to the original DN. If you set this option to `0`, no timers are started and calls will remain parked indefinitely.

monitor-consult-calls

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Monitoring Consultation Calls" on page 141

Specifies whether SIP Server will monitor consultation calls. If you set this option to `false`, SIP Server does not monitor consultation calls. This is default behavior. To enable monitoring of consultation calls, set this option to `true`. SIP Server will monitor all calls made to or from an agent under supervision.

monitor-internal-calls

Default Value: `true`
Valid Values:

`true`	SIP Server starts monitoring sessions for all calls on the DN where call supervision subscription is active.
`false`	SIP Server starts monitoring sessions only if external parties participate in the call.

Changes Take Effect: Immediately
Related Feature: "Call Supervision" on page 139

Specifies SIP Server behavior to start monitoring sessions.

monitor-party-on-hold

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: After SIP Server restart
Related Feature: "Customer-on-Hold Privacy" on page 143

When this option is set to `true` (the default), the supervisor in the Whisper or in the Silent mode might be able to hear the customer if the agent has put the call on hold and there are no other active participants in the call.

When this option is set to `false`, the supervisor in the Whisper or Silent mode is not be able to hear the customer if the customer is an external party, the agent has put the call on hold, and there are no other active participants in the call.

msml-enable-record-extensions

Default Value: `true`
Valid Value: `true, false`
Changes Take Effect: Immediately

When this option is set to `true`, SIP Server sends the recording parameters in the `INFO` message to Media Server while restarting recording for a particular DN for which a `TPrivateService` request with recording parameters was issued earlier. When this option is set to `false`, SIP Server does not send recording parameters when it restarts recording.

msml-location-alarm-timeout

Default Value: `0`
Valid Values: `0-65535`
Changes Take Effect: Immediately
Related Feature: “Geo-Location for MSML-Based Services: Strict Matching” on page 393

Enables a configurable alarm when a connection that involves an MSML DN and is restricted by geo-location, cannot be established. SIP Server maintains an alarm log of failed attempts and will display a `52052` message that lists those failures. The value of this option is the number of seconds between displays.

When the value is `0` or this option is not configured, no alarms are raised.

msml-mute-type

Default Value: `1`
Valid Values: `1, 2`
Changes Take Effect: Immediately
Related Feature: “Muting/Unmuting a Party in a Conference” on page 167

Specifies the type for muting/unmuting a party in a conference. Type 1 is required to support remote mute functionality in SIP Server. Type 2 is for backward compatibility.

Warning! Use this option only when requested by Genesys Customer Care.

msml-oos-recover-enabled

Default Value: `false`
Valid Value: `true, false`
Changes Take Effect: Immediately

Determines how SIP Server handles the scenario in which it detects a `Voice over IP Service` DN (`service-type=msml`) as out of service. If set to `true`, SIP Server re-connects the applied treatment, conference services, or music services—music on hold or music in queue—with an alternate `Voice over IP Service` DN (`service-type=msml`). If set to `false`, SIP Server does not re-connect the treatment, conference, or music services. This option takes effect only if Active Out-Of-Service Detection is enabled.

msml-record-support

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Call Recording—MSML-Based" on page 125

Set this option to `true` to enable MSML-based call recording. If using NETANN service, set this option to `false`.

Note: Earlier versions of Genesys Media Server (prior to release 8.1.4) do not support MSML-based call recording. If `msml-support` is set to `true`, but NETANN is required for call recording services only, set `msml-record-support` to `false`. This allows NETANN for call recording and MSML for other media services.

msml-record-metadata-support

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether SIP Server, while starting call recording, sends additional metadata in the `INFO` message to Genesys Media Server. If set to `false` (the default), SIP Server does not include additional metadata in the `INFO` message. If set to `true`, SIP Server sends additional metadata to the Genesys Media Server for use in Genesys Media Server file-based call recording.

msml-support

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Features: "Outbound IP Solution Integration" on page 306, "Genesys Media Server Integration" on page 94

Enables SIP Server to engage Genesys Voice Platform (GVP) as a media server through the MSML protocol. Set this parameter to `true` when integrating SIP Server into the Outbound IP Solution and/or when integrating with Genesys Media Server.

Outbound IP Integration. Set this option to `true` when SIP Server is used as part of the Outbound IP Solution. In this mode SIP Server processes `TMakePredictiveCall` requests by engaging GVP configured as a `Voice over IP Service` DN with `service-type` set to `msml`, or as a `Trunk Group DN`.

Genesys Media Server Integration. Set this option to `true` to enable MSML service for all media services operations (treatments, music, greetings and conferences) using the Genesys Media Server. If set to `true`, SIP Server engages the Genesys Media Server configured as a `Voice over IP Service` DN with `service-type` set to `msml`.

music-in-conference-file

Default Value: The value is taken from the `default-music` option
Valid Values: A string containing the valid name of the music file
Changes Take Effect: Available for next 3pcc or 1pcc hold operation
Related Feature: “Silence Treatment in Conference” on page 165

Specifies the silent audio file to be played in applicable conferences (more than two active participants). If the conference has only two active participants, then the music file defined in the `default-music` option will be played for the other party when the call is placed on hold. For conferences with more than two active participants, the `music-in-conference-file` option is used for a silent MOH treatment instead. Recommended value is `music/silence`.

For example, if a supervisor in silent monitoring mode listens to a call between a customer and an agent, the supervisor is not considered an active participant. If the agent or customer places the call on hold, the remaining participants will hear the default-music MOH treatment. However, if the supervisor places the call on hold, the music/silence file is played instead, so the customer and agent can continue their conversation undisturbed.

music-in-queue-file

Default Value: No default value
Valid Value: `<default_music_directory>/<file_name>`
Changes Take Effect: Immediately for all new calls
Related Feature: “Playing Music to Calls in Queue” on page 183

Specifies the file name of the music file to be played when a call is queued on a particular ACD Queue.

Notes: This option is set at the SIP Server Application level. At the Switch/DN level, the option `default-music` is used instead to specify the file for music-in-queue. This `default-music` setting at the DN-level takes precedence over the Application-level `music-in-queue-file` setting.

If there is no value specified for either option, the value of the Media Server DN `request-uri` option is used instead.

---

music-listen-disconnect

Default Value: `music/on_hold`
Valid Values: The path to any valid audio file
Changes Take Effect: For the next `TListenDisconnect` request
Related Feature: “Private Conversations During Conference” on page 166

Specifies the path to an audio file to be played to the temporary disconnected party from the conference.

music-on-pbxpark

Default Value: No default value
Valid Values: Path to a valid audio file
Changes Take Effect: Immediately
Related Feature: "Call Park/Retrieve" on page 119

Specifies the music file to be played to the remote party when a call is parked on the `gcti::pbxpark` device.

mwi-agent-enable

Default Value: `false`
Valid Values:

`true`	MWI for the agent's voice mail box is enabled.
`false`	MWI for the agent's voice mail box is disabled.

Changes Take Effect: Immediately

Enables or disables MWI for the agent's voice mail box.

Note: This option is obsolete. It is not required for integration with SIP Feature Server.

mwi-domain

Default Value: No default value
Valid Value: Any computer name
Changes Take Effect: During the next attempt to register for MWI

Specifies the computer name in the URI of the `REGISTER` request. SIP Server sends this information to Asterisk in order to initiate MWI. The value of this option must be a computer name that is recognized by Asterisk.

Note: This option is obsolete. It is not required for integration with SIP Feature Server.

mwi-extension-enable

Default Value: `false`
Valid Values:

`true`	MWI for the extension's voice mail box is enabled.
`false`	MWI for the extension's voice mail box is disabled.

Changes Take Effect: Immediately

Enables or disables MWI for the extension's voice mail box.

Note: This option is obsolete. It is not required for integration with SIP Feature Server.

mwi-group-enable

Default Value: `false`
Valid Values:

`true`	MWI for the agent group's voice mail box is enabled
`false`	MWI for the agent group's voice mail box is disabled.

Changes Take Effect: Immediately

Enables or disables MWI for the agent group's voice mail box.

Note: This option is obsolete. It is not required for integration with SIP Feature Server.

mwi-host

Default Value: No default value
Valid Values: Any host name or IP address
Changes Take Effect: During the next attempt to register for MWI

Specifies the host name of the Voice Mail system to get MWI notification from the host where Asterisk is running. SIP Server will send a `REGISTER` request to `mwi-host:mwi-port` to initiate MWI.

Note: This option is obsolete. It is not required for integration with SIP Feature Server.

mwi-implicit-notify

Default Value: No default value
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether SIP `NOTIFY` requests are sent to an endpoint regardless of subscription, for integration with Genesys SIP Feature Server.

If set to `true`, SIP Server sends the `NOTIFY` even if it has not received a `SUBSCRIBE` from the phone. If set to `false`, SIP Server sends the `NOTIFY` to the phone only if it has subscribed for the event `message-summary`.

If this option is not defined, SIP Server does not send any `NOTIFY` to the phone.

Note: Genesys recommends that, if you require MWI but your phone does not send `SUBSCRIBE`, set this option to `true`.

For more information about integrating with SIP Feature Server, see *SIP Feature Server 8.1 Deployment Guide*.

mwi-mode

Default Value: SUBSCRIBE
Valid Values: REGISTER, SUBSCRIBE
Changes Take Effect: After SIP Server restart

When this option is set to SUBSCRIBE, SIP Server activates SIP subscriptions for all voice mail box owners as configured by other mwi-<> options. If set to REGISTER, the MWI functionality is enabled using the REGISTER SIP request method. For backward compatibility with the previous SIP Server releases, set this option to a value of REGISTER.

Note: This option is obsolete. It is not required for integration with SIP Feature Server.

mwi-notify-unregistered-dn

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

Specifies whether SIP NOTIFY requests are sent to an endpoint regardless of SIP registration. This setting is required for integration with Genesys SIP Feature Server:

- If set to true, SIP Server sends the MWI NOTIFY to the phone even if the phone has not registered.
- If set to false, SIP Server sends the MWI NOTIFY to the phone only if the phone has registered with SIP Server.

Note: The above functionality works with the option mwi-implicit-notify. The mwi-notify-unregistered-dn option must to be set false to support the above functionality.

mwi-port

Default Value: No default value
Valid Value: Any available port
Changes Take Effect: During the next attempt to register for MWI

Specifies the port of the Voice Mail system to get MWI notification from the port where Asterisk is running. SIP Server will send a REGISTER request to mwi-host:mwi-port to initiate MWI.

Note: This option is obsolete. It is not required for integration with SIP Feature Server.

mwi-subscribe-vmb

Default Value: `false`
Valid Values: `false, true`
Changes Take Effect: Immediately
Related Options: `mwi-implicit-notify`, `mwi-notify-unregistered-dn`

Specifies whether SIP Server accepts MWI subscriptions that are submitted for a voice mailbox. If set to `true`, SIP Server accepts MWI subscriptions for voice mailbox numbers and for DNs. If set to `false` (the default), SIP Server accepts MWI subscriptions for only DNs.

In addition, the `mwi-implicit-notify` option must be set to `false`, and the `mwi-notify-unregistered-dn` option must be set to `true` to support this feature.

Note: This feature requires SIP Feature Server version 8.1.2 or later.

nas-private

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "No-Answer Supervision" on page 302

Specifies whether No-Answer Supervision is enabled for private calls.

You can set this option at the Application and Switch/Agent Login or Switch/DN level (DN of type `Extension`). When set at the Application level, the option value is applied globally to all private calls. When set at the Switch level, the option value is applied to a particular DN or Agent Login.

Note: The option setting at the Switch level takes precedence over the Application level setting.

network-monitoring-timeout

Default Value: `1`
Valid Values: `1-30`
Changes Take Effect: Immediately
Dependent Options: `sip-nic-address`, `tlib-nic-monitoring`, `sip-iptakeover-monitoring`
Related Feature: See "Network Status Monitoring" in the *SIP Server 8.1 High-Availability Deployment Guide*.

Defines the time interval (in seconds) for which SIP Server checks the network status of:

- The SIP NIC, if a dedicated NIC is used and the `sip-nic-address` option is configured.
- The T-Library NIC, if the value of the `tlib-nic-monitoring` option is set to `true`.
- The Virtual IP address for the IP Address Takeover configuration, if the value of the `sip-iptakeover-monitoring` option is set to `true`.

no-login-on-presence

Default Value: `false`
Valid Values: `false, true`
Changes Take Effect: Immediately

When set to `true` and, after receiving a SIP `NOTIFY` presence message, SIP Server does not log an agent into a desktop. A `Ready/Not Ready` state can be changed by a `NOTIFY` message. If using this option, Genesys recommends setting it before a SIP Server application is started.

observing-routing-point

Default Value: No default value
Valid Values: A `Routing Point` DN
Changes Take Effect: On the next call
Related Feature: “Remote Supervision” on page 153

Specifies the service observing Routing Point used for Multi-Site Supervision of the agents, whose endpoints are controlled by this SIP Server. This option must contain a number of a valid `Routing Point` DN in order for the Multi-Site Supervision feature to work.

operational-stat-timeout

Default Value: `10`
Valid Values: `3-65535`
Changes Take Effect: Immediately
Related Feature: “Call Recording—MSML-Based” on page 125, “HTTP Monitoring Interface” on page 245

Specifies how often, in seconds, a local LCA is queried for system information such as CPU and memory usage. This information is then written into the SIP Server Operational Information log as defined in the SIP Server configuration.

Note: This functionality requires Management Framework 8.1.2 or later.

outbound-bsns-calls

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: “Business-Call Handling” on page 235

Specifies whether SIP Server considers all established outbound calls on an agent as business calls.

overflow-location-map

Default Value: No default value (empty string)
Valid Values: Any valid string with comma-separated elements
Changes Take Effect: On the next call

Related Feature: "Geo-Location for MSML-Based Services: Strict Matching" on page 393

This option creates an association between geo-location labels and overflow-location labels, to support strict geo-location matching.

The format of the option is: `geo-location label=overflow-location label`.

For example, in `labelA=labelB,labelC=labelD`...

`labelA` and `labelC` are geo-location labels; `labelB` and `labelD` are overflow-location labels.

If services or resources that match the call by `geo-location=labelA` are not available, SIP Server will try services or resources that matches the call by `overflow-location=labelB`.

overload-ctrl-call-rate-capacity

Default Value: `200`
Valid Values: `0-10000`
Changes Take Effect: Immediately
Related Feature: "Overload Control" on page 313

Specifies the Call Rate (calls/sec) that SIP Server is able to maintain without performance degradation. For example, SIP Server instances deployed on host machines with a slower CPU would use a lesser option value. This capacity must be set to a value at least 2 times higher than the `overload-ctrl-threshold` (enforced by SIP Server). If you set this option to `0`, SIP Server does not monitor this load factor.

overload-ctrl-call-tapplytreatment-requests-rate

Default Value: `0`
Valid Values: `0-1000`
Changes Take Effect: Immediately
Related Feature: "Overload Control" on page 313

Specifies the `TApplyTreatment` request rate (T-Requests/sec) allowed for each call. If the T-Request rate for any particular call exceeds the configured value, SIP Server first sends a warning and, if T-Requests continues to increase, SIP Server rejects excessive `TApplyTreatment` requests for that particular call. Setting the value of the option to `0` (zero) disables this functionality.

overload-ctrl-call-trequests-rate

Default Value: `0`
Valid Values: `0-1000`
Changes Take Effect: Immediately
Related Feature: "Overload Control" on page 313

Specifies the T-Request rate (T-Requests/sec) that is allowed for each call. This prevents performance degradation if particular clients issue too many requests. If the T-Request rate for any particular call exceeds the configured value, SIP Server first sends a warning and, if T-Requests continues to increase, SIP Server rejects excessive T-Requests for that call. Setting the value of the option to `0` (zero) disables this functionality.

overload-ctrl-call-tupdateuserdata-requests-rate

Default Value: `0`
Valid Values: `0-1000`
Changes Take Effect: Immediately
Related Feature: "Overload Control" on page 313

Specifies the User Data update T-Request (`TAttachUserData, TUpdateUserData, TDeleteUserData, TDeletePair`) rate (T-Requests/sec) that is allowed for each call. If the T-Request rate for any particular call exceeds the configured value, SIP Server first sends a warning and, if T-Requests continues to increase, SIP Server rejects excessive user data T-Requests for a particular call. Setting the value of the option to `0` (zero) disables this functionality.

overload-ctrl-dialog-rate-capacity

Default Value: `400`
Valid Values: `0-10000`
Changes Take Effect: Immediately
Related Feature: "Overload Control" on page 313

Specifies the SIP Dialog Rate (dialogs/sec) that SIP Server is able to maintain without performance degradation. For example, SIP Server instances deployed on host machines with a slower CPU would use a lesser option value. This capacity must be set to a value at least 4 times higher than the `overload-ctrl-threshold` (enforced by SIP Server). If you set this option to `0`, SIP Server does not monitor this load factor.

overload-ctrl-threshold

Default Value: `0`
Valid Values: `0-65535`
Changes Take Effect: Immediately
Related Feature: "Overload Control" on page 313

Specifies the call rate (calls/second) after which the Dialog Rate and Call Rate overload control mechanism becomes active. To disable the Dialog Rate and Call Rate mechanism, set this option to the default (`0`).

overload-ctrl-trequests-rate

Default Value: `0`
Valid Values: `0-10000`
Changes Take Effect: Immediately
Related Feature: "Overload Control" on page 313

Specifies the T-Request rate (T-Requests/sec) that SIP Server is able to maintain without performance degradation. For example, if SIP Server is deployed on host machines with a slow CPU, then set this option to a lesser value. If the T-Request rate exceeds a configured value, SIP Server first sends a warning and, if T-Requests continue to increase, SIP Server rejects excessive T-Requests. Setting the value of the option to `0` (zero) disables this functionality.

override-to-on-divert

Default Value: `false`
Valid Values:

`true`	The username is equal to the destination DN.
`false`	The username is equal to the Routing Point or ACD Queue number.

Changes Take Effect: Immediately

Controls the username part of the `To` header URI for outgoing `INVITE` messages when a call is diverted from a Routing Point or an ACD Queue. This option setting also applies to 1pcc transfers (using the `REFER` method). If set to `true`, SIP Server takes the value of the `REFER`-to DN as the username part of the `To` header. If set to `false`, SIP Sever takes the value of the `From` DN (originator of the `REFER` message) for the username part of the `To` header.

Note: This option can also be configured at the DN level. The DN-level setting takes precedence over the Application-level setting.

p-asserted-identity

Default Value: No default value (empty string)
Valid Values: Any string in accordance with RFC 3325
Changes Take Effect: On the next call
Related Feature: "Network Asserted Identity" on page 292
Related Option: "privacy" on page 600

Specifies the preferred SIP URI or telephone number that SIP Server inserts in the `P-asserted-identity` header of `INVITE` messages, when required according to the scenarios described in "Network Asserted Identity" on page 292. If enabled, SIP Server adds this value to the content of the `P-asserted-identity` header for all dialogs—unless otherwise specified at the DN-level.

parking-music

Default value: `music/silence`
Valid Values:
Changes Takes Effect: For the next parked call
Related Feature: “Remote Supervision” on page 153

Specifies the music file, which is played to the remote party parked on the `gcti::park` DN.

partition-id

Default Value: SIP Server default partition
Valid Values: Any string (the name of one partition)
Changes Take Effect: Immediately

Application-level: Specifies the default partition for the particular SIP Server application.

DN-level: Specifies the partition to which this DN belongs. If you leave the option undefined, SIP Server considers the DN as belonging to the default partition. You can only define this option on `Trunk` and `Voice over IP Service`-type DNs.

SIP Server assigns a partition to each call based on the call origination device. If the call origination device does not have a defined `partition-id` parameter, the call is assigned to the default partition.

SIP Server uses information about partition-id in two ways:

1. To select a `Voice over IP Service` DN for a call.
2. To select a `Trunk` DN for the outbound call.

If multiple resources (`Voice over IP Service` DNs or `Trunk` DNs) are available for call processing, SIP Server selects one that belongs to a call partition.

posn-no-answer-overflow

Default Value: No default value
Valid Values:

`none`	SIP Server does not attempt to overflow a call on a position when the time specified in the `posn-no-answer-timeout` option expires.
`recall`	SIP Server returns the call to the last distribution device (the device reported in the `ThisQueue` attribute of the call) when the time specified in the `posn-no-answer-timeout` option expires.
`release`	SIP Server releases the call.
Any valid overflow destination	SIP Server returns the call to the specified destination when the time specified in the `posn-no-answer-timeout` option expires.

Changes Take Effect: Immediately
Related Feature: “No-Answer Supervision” on page 302

Specifies a sequence of overflow destinations (separated by a comma) that SIP Server attempts to overflow to when the time specified in option `posn-no-answer-timeout` expires. SIP Server attempts to overflow in the order specified in the list.

When all overflow attempts fail, SIP Server abandons overflow. See also extension `NO_ANSWER_OVERFLOW` in section "Using the Extensions Attribute" on page 414 for more information about how this option is used.

When the list of overflow destinations contains the value `recall` and the call was not distributed, SIP Server skips to the next destination in the list.

SIP Server obtains the value for this option in the following order of precedence:

1. `no-answer-overflow` if defined at a DN level of type `ACD Position` and applies when agents logged out.
2. `posn-no-answer-overflow` if defined at a SIP Server Application level.

posn-no-answer-timeout

Default Value: `15`
Valid Value: Any integer from `0`–`600`
Changes Take Effect: Immediately
Related Feature: "No-Answer Supervision" on page 302

Defines the default no-answer timeout (in seconds) that SIP Server applies to any device of type `position`. When the timeout ends, SIP Server executes the actions defined in option `posn-no-answer-overflow`.

If set to `0`, the No Answer Supervision feature for `DN`s of type `ACD Position` is disabled. See the `NO_ANSWER_TIMEOUT` extension in section "Using the Extensions Attribute" on page 414 for more information about how this option is used. SIP Server obtains the value for this option in the following order of precedence:

1. `no-answer-timeout` if defined at a DN level of type `ACD Position` and applies when agents logged out.
2. `posn-no-answer-timeout` if defined at a SIP Server `Application` level.

predictive-call-router-timeout

Default Value: `0`
Valid Value: Any non-negative integer
Changes Take Effect: After SIP Server restart
Related Option: `router-timeout`

Specifies the maximum time (in seconds) that an answered predictive call can wait on a Routing Point DN for either: A) any Universal Routing Server (URS) request, or B) an agent to answer the call after a successful URS request. If no request is made during this timeout period, the call is dropped. This feature is intended as a clean-up mechanism for scenarios where URS becomes non-operational or the agent does not answer the call.

SIP Server clears the timer after receiving the TApplyTreatment, or if the router-timeout is activated for any new request coming in for the call. For example, a TRouteCall request.

For Active Switching Matrix (ASM) and Proactive Notification modes, the timeout defines the maximum time that an answered predictive call will wait on a Trunk Group DN for a TMergeCall request from Outbound Contact Server (OCS). If no request arrives during this time, SIP Server drops the call. SIP Server clears the timer when the Trunk Group DN receives the ApplyTreatment message, or if the router-timeout is activated for any new request coming in for the call.

The default value of 0 (zero) disables the predictive-call-router-timeout functionality. If it is disabled, then TMakePredictiveCall requests are timed using the router-timeout option.

preview-expired

Default Value: 90
Valid Values: Any positive integer
Changes Take Effect: Immediately for future calls
Related Feature: "Preview Interactions" on page 335

Specifies the time (in seconds) that the Preview Interaction dialog box remains open on a desktop. After the time expires, the dialog box closes and the desktop changes to a Not Ready state.

The preview-expired option works with the preview-interaction and forced-notready options to determine what action to take when a desktop does not respond to a preview interaction before the time expires.

privacy

Default Value: No default value (string is empty)
Valid Values: id—as defined in RFC 3325: Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks
Changes Take Effect: On the next call
Related Feature: "Network Asserted Identity" on page 292

Specifies the level of privacy requested by the DNs in this switch, as described by the Network Asserted Identity feature. If set to id, SIP Server includes the Privacy:id header in SIP messaging. If not configured, SIP Server does not include the Privacy:id header (privacy is not requested by the DNs for this switch, unless otherwise specified at the DN level).

reason-in-extension

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether SIP Server includes the `ReasonCode` in either the `Extensions` or `Reason` Attribute in `EventAgentNotReady` messages.

If set to `true`, SIP Server includes the key-value pair `Reasoncode` with the value `private-call` in the `Extensions` Attribute of the `EventAgentNotReady` message sent in response to a SIP `NOTIFY` message stating that there is an active call on the private line DN.

If set to `false`, this key-value pair is included instead in the `Reason` Attribute of the `EventAgentNotReady` message.

record-after-merge

Default Value: `false`
Valid Values: `false, true`
Changes Take Effect: Immediately

If this option is set to `false`, SIP Server will not start recording for agents who are moved from a consultation call to a main call when a transfer or conference is completed (8.0.2 backward compatibility support). If the option is set to `true`, SIP Server will start the recording for agents who are moved from the consultation call to the main call after the transfer or conference is completed.

Note: This option is not needed in a consultation-call-recording-enabled environment (`record-consult-calls=true`) and recording must be enabled on the agent to start the recording.

record-agent-greeting

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

Specifies whether the agent greeting or the customer greeting must be recorded when both recording and greeting are enabled for the call.

If set to `true`, the agent greeting is recorded.

If set to `false`, the customer greeting is recorded.

record-consult-calls

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately (established connections remain unaffected)
Related Feature: “Call Recording—NETANN-Based” on page 121
Related Option: `record`, `recording-filename`

Specifies whether consultation calls are eligible for regular call recording.

If set to `false`, SIP Server does not allow recording for consultation calls even if one or more of the participating DNs are set for call recording (`record` option on the DN is set to `true`, or the DN is the party specified in the `record extension of a TRouteCall` request).

If set to `true`, SIP Server allows recording of consultation calls that include at least one appropriately set DN.

record-metadata-prefix

Setting: `TServer` section, the SIP Server Application (takes priority) or the VOIP Service DN with `service-type=sip-cluster-nodes`
Default Value: An empty string
Valid Values: Any valid string
Changes Take Effect: On the next call
Related Feature: “Passing Extended Recording Metadata to GVP” on page 403

Specifies the prefix that must match the initial characters of GVP parameters to be added from `AttributeExtensions` of `TRouteCall` in the `INFO` message sent to MCP. The matching KVPs are sent under recording metadata as additional GVP parameters with the prefix value stripped out. The setting at the Application level takes priority over the VOIP Service DN level setting. If this option is configured with an empty value at a VOIP Service DN, the existing recording metadata is sent without additional GVP parameters.

Those GVP parameters are added only if the following conditions are met:

- The KVP's prefix matches the `record-metadata-prefix` option value.
- The total number of matching KVPs does not exceed 5. If exceeded, no KVPs are attached to the call data (metadata storage) and no additional GVP parameters are passed to MCP in the recording `INFO` message.
- Call recording is enabled.

record-moh

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Recording Calls Without Music-on-Hold Treatment" on page 128
Related Option: `sip-enable-moh`

Specifies whether the music-on-hold treatment is recorded during call recording.

If set to `false`, SIP Server pauses the recording when the call is placed on hold and the music-on-hold treatment will not be recorded. SIP Server resumes the recording when the call is retrieved.

If set to `true`, the music-on-hold treatment is always recorded during call recording.

Note: This option is applicable only if `sip-enable-moh` is set to `true` and the MSML configuration is used for recording.

recording-failure-alarm-timeout

Default Value: `0`
Valid Values: `0`–`65535` (seconds)
Changes Take Effect: Immediately
Related Feature: "Call Recording Alarms" on page 130

Enables call recording alarm notification. When this option is set to a value other than `0` (zero), and a call recording failure is detected, SIP Server generates a `52051` alarm message and starts the timer using the interval defined by this configuration option. Each consecutive call recording failure detected during this period increments the counter.

When the timer expires, SIP Server generates an alarm message with the number of failures detected in the past interval. If the timer expires and no recording failures have been detected within the past interval, SIP Server does not generate an alarm message.

Setting this option to `0` (zero) disables the feature.

Note: This call recording alarm is designed as a persistent alarm. An administrator can clear this alarm manually or use the Clearance Timeout timer in Genesys Administrator Extension.

recording-filename

Default Value: NULL
Valid Values: Any valid file name using the variables specified below
Changes Take Effect: When the next call recording is initiated
Related Feature: "Call Recording—NETANN-Based" on page 121

Specifies the file name for call recording when call recording is initiated automatically, according to the SIP Server configuration. When this option contains a value, the generated file name is added as UserData to the call with the GSIP_REC_FN key. When this option does not contain a value, the file name is the UUID of the call.

The following variables are used when creating the file:

ANI:	The calling number.
$DNIS$:	The called number.
$DATE$:	The current date (GMT) in the Y-M-D format.
$TIME$:	The current time (GMT) in the H-M-S format.
$CONNID$:	The Connection ID of the call.
$UUID$:	The UUID of the call.
$AGENTID$:	The Agent Login ID, if the agent is logged in on the device where the call recording is initiated.
$AGENTDN$:	The DN where the call recording is initiated.

refer-enabled

Default Value: true
Valid Values: true, false
Changes Take Effect: With the next new call

Specifies whether the REFER method is sent to an endpoint. If set to true, the REFER method is sent to:

- The call party that originates a TMakeCall request.
- The call party that initiates a consultation call.
- The call party that is transferred to another destination during a single-step transfer.

If set to false, SIP Server uses the re-INVITE method instead.

For IMS deployments: When integrated with an IMS environment, you must set this option to false either globally on the Application, or individually on all IMS-enabled DNs.

registrar-default-timeout

Default Value: 1800
Valid Values: 0-4294967295
Changes Take Effect: At the next REGISTER dialog

Specifies the expiration timeout for a REGISTER request as a value (in seconds) in the 200 OK response that is sent by SIP Server to the SIP endpoint. When the option is set to 0, or is not defined, the Expires header value from the REGISTER request is used as the expiration timeout. If the option is set to any value other

than `0`, the timeout is set to the lesser of the option value and the value specified by the client.

For IMS deployments: With SIP Server integrated into an IMS environment, you can either leave this option undefined (no timeout), or set this option to the maximum value (`3600`). A smaller number can result in the expiry of DN registrations in SIP Server, because IMS does not propagate SIP Server responses back to the endpoint, so it will not refresh the registration.

Note: Genesys recommends that you do not set this option to a value less than 64 seconds. This guarantees that a new registration will not arrive within the SIP Server default interval of 32 seconds, which is the default value for keeping a non-responded SIP transaction alive.

releasing-party-report

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Call Release Tracking" on page 138

Specifies whether SIP Server reports the `Extensions` attribute key `ReleasingParty` in events `EventReleased` and `EventAbandoned` to indicate which party initiated the call release.

report-error-on-routing-end

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

If set to `true`, SIP Server generates `EventError` with `ErrorCode 453` and the `Call has been disconnected` error message. This applies to the scenario where the `divert-on-ringing` option is set to `false`, and a call routed to an agent is still in the ringing state when the caller drops the call.

reset-acw-persistent-reasons

Default Value: No default value
Valid Values: `agentlogout, agentready, all`
Changes Take Effect: Immediately
Related Options: `acw-persistent-reasons` option is set to `true`

If set to `agentlogout`, SIP Server resets `AttributeReason` on receiving `EventAgentLogout`. If set to `agentready`, SIP Server resets `AttributeReason` on receiving `EventAgentReady`. If set to `all`, SIP Server resets `AttributeReason` on receiving `EventAgentLogout` or `EventAgentReady`.

resolve-external-contact

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

Specifies whether SIP Server resolves the contact as external if the internal resolution has failed. The value will be taken from the trunk which was the source of the OOSP-causing message, which is the trunk to the transfer initiator party.

This option affects only processing of the OOSP (Out Of Signaling Path) transfer SIP operations, specifically `REFER` requests or `302` responses. It applies only to DNs of type `Trunk`.

SIP Server tries to find the destination device using the URI in the OOSP message, as follows:

- Resolving the user part—SIP Server searches among locally configured and registered DNs and tries to match trunk prefixes.
- If no matching DNs are found and if the `resolve-external-contact` option is set to `true`, SIP Server tries to find the destination trunk by matching the domain part of the received URI with the contact of the configured `Trunk` DNs.

Note: This option can also be configured at the DN level. The DN-level setting takes precedence over the Application-level setting.

resolve-internal-rp-by-host

Default Value: `false`
Valid Values: `false, true`
Changes Take Effect: For next call

In Business Continuity deployments, SIP Server can correctly resolve where to send a request when two different sites have a Routing Point configured with the same DN number. When set to `true`, SIP Server will include host information when resolving the internal Routing Point number.

When set to `false` (the default), enables the previous behavior in SIP Server.

resolve-sip-address

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After restart

If set to true, SIP Server will resolve the hostname in the `OtherDN` URI of the `RequestRouteCall` to its corresponding IPv4 address, which it then uses in the `302 Contact` message. Resolution through DNS is only done once, and will be stored in the internal server table. The corresponding IPv4 address will be extracted from this table for any similar subsequent `RequestRouteCall`. SIP Server must be restarted for a new IP address to be assigned to an FQDN.

resource-management-by-RM

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether SIP Server communicates with Genesys Media Server through the GVP Resource Manager, or with the media server directly. When integrating with Genesys Media Server, Genesys recommends that you include both Resource Manager (to handle media service distribution) and GVP Media Control Platform (to provide the media service itself) in the deployment. This is the default (`true`) behavior. For direct integrations with the media server, set this option to `false`.

restart-period

Default Value: `20`
Valid Values: `0–600`
Changes Take Effect: Immediately

Specifies the interval (in seconds) that SIP Server waits between attempts to reconnect to the switch when the link fails. A value of `0` (zero) means SIP Server does not try to reconnect unless the link configuration is changed.

reuse-tls-conn

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: On the next call

Specifies whether SIP Server reuses the existing TLS transport for sending SIP requests. If set to `false`, SIP Server opens a new TLS connection to the SIP request destination. If set to `true`, SIP Server reuses the existing TLS transport for sending SIP requests.

route-failure-alarm-high-wm

Default Value: `10`
Valid Values: Positive integer for absolute value or floating point number followed by % (percent) symbol. For example—10%, 2.25%, 5E-2%.
Changes Take Effect: Immediately
Related Feature: See “Failed Route Notifications” on page 242

Defines the high water mark which must be reached in order for a route failure alarm to be triggered, within the period configured in the `route-failure-alarm-period` configuration option.

route-failure-alarm-low-wm

Default Value: `1`
Valid Values: Positive integer for absolute value or floating point number followed by % (percent) symbol. For example—10%, 2.25%, 5E-2%.
Changes Take Effect: Immediately
Related Feature: See "Failed Route Notifications" on page 242

Defines the low water mark which must be reached, while under the route failure alarm condition, within the period configured in the `route-failure-alarm-period` configuration option.

Note: This option also specifies the minimum time between alarm setting and alarm clearing.

route-failure-alarm-period

Default Value: `0`
Valid Values: Positive integer
Changes Take Effect: Immediately
Related Feature: See "Failed Route Notifications" on page 242

Defines the interval (in seconds) in which the number of failed route requests is totalled, in order to determine either a possible route failure alarm or the cancelation of an alarm, based on the failed route counter reaching the relevant high or low water mark.

ring-tone

Default Value: `music/ring_back`
Valid Values: Name and path of any valid audio file
Changes Take Effect: Immediately for all new calls

Specifies the audio file to be played for the `RingBack` treatment.

ringing-on-route-point

Default Value: `true`
Valid Values:

`true`	SIP Server responds with `a 180 Ringing` message.
`false`	SIP Server does not respond with a `180 Ringing` message.

Changes Take Effect: Immediately

Specifies whether SIP Server responds with a `180 Ringing` message when a call arrives at a Routing Point. It enables transfers for calls waiting at Routing Points. The disadvantages are:

- Possible undesirable ringback tone.
- Multiple ringing messages delivered for the same call.

router-timeout

Default Value: `10`
Valid Value: Any non-negative integer
Changes Take Effect: Immediately

Specifies the maximum time (in seconds) that a call remains on a Routing Point without a treatment. If the timeout is triggered, the call is sent to the DN specified in `default-dn`.

rp-use-dial-plan

Default Value: `default`
Valid Values: `default, full, partial, false, agentid`
Changes Take Effect: Immediately
Related Feature: “Dial Plan” on page 195

Specifies how SIP Server applies the dial plan:

- `default`—For a SIP Server dial plan, the same as the `false` value. For a Feature Server dial plan, the same as the `partial` value.
- `full`—The dial plan is applied to the destination of `TRouteCall`, including the digit translation and forwarding rules.
- `partial`—Only the digit translation is applied to a dial-plan target. Forwarding rules, such as forwarding on no answer (`ontimeout`), forwarding on busy (`onbusy`), forwarding on DND (`ondnd`), forwarding on no response (`onunreach`), and forwarding on not SIP registered (`onnotreg`) are not applied. Valid for both SIP Server and SIP Feature Server dial plans.
- `false`—No dial plan is applied to the destination of `TRouteCall`.
- `agentid`—No dial plan is applied to the destination of `TRouteCall`; only an agent ID provided by SIP Feature Server is added to the response.

If the SIP Server dial plan is used, SIP Server selects the dial plan assigned to the caller. This is the dial plan configured for the DN/Agent Login of the DN for internal calls, or the Trunk DN for inbound calls, or the Application-level option if no DN/Agent-Login-level dial plan is configured.

send-200-on-clear-call

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

When this option is set to `true`, SIP Server, when executing a `TReleaseCall` request for a non-established call, terminates the call leg in the dialing state by sending a `200 OK` message. When this option is set to `false`, SIP Server sends a `404 Not Found`.

server-role

Default Value: `0`
Valid Value: `0, 1`
Changes Take Effect: After SIP Server restart

Specifies the role that SIP Server plays in the deployment scenario:

- `0`—SIP Server runs in a standalone deployment, where the server is not integrated into an IMS environment (DNs are registered or provisioned on SIP Server).
- `1`—SIP Server runs as a SIP Application Server (SIP-AS) in an IMS deployment.

session-refresh-enforced

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

Controls whether SIP Server activates the SIP Session timer within a SIP dialog. If set to `false`, SIP Server activates the SIP Session timer only if both an initial `INVITE` and `200 OK` response to that `INVITE` contains the `Session-Expires` header. If set to `true`, SIP Server, while activating that timer, ignores the absence of the `Session-Expires` header in the response and starts the timer based on the header presence in the request. If an endpoint does not support the session refresh mechanism, set this option to `false`. The option has an affect only when the `session-refresh-interval` option is set to a non-zero value.

You can define this option at both the Application and the DN levels. The DN-level setting takes precedence over the Application-level setting.

session-refresh-interval

Default Value: `1800`
Valid Values: `0, 90–86400`
Changes Take Effect: Immediately

Specifies (in seconds) how often active calls are checked to see if they are still active. A `0` (zero) value disables this feature (the session refresh mechanism is turned off). Values between `1` and `89` (inclusive) are treated as value `90`.

This option is used to remove stuck calls that must accumulate if endpoints terminate calls without sending the appropriate SIP message.

Note: In compliance with RFC 4028, Genesys recommends keeping the default setting of `1800` (30 minutes) for this option. Setting the `session-refresh-interval` to a considerably lower number may cause conflicts with the session timer in some switches.

set-notready-on-busy

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

With this option set to `true`, when a call is distributed to a ready agent (that is, the agent is not previously engaged in a call), and the agent endpoint responds to the `INVITE` with a `4xx`, `5xx`, or `6xx` message, SIP Server places the agent in the `Not Ready` state (an `EventAgentNotReady` message is distributed). In addition, a `ReasonCode` key with a value equal to a returned error will be reported in the `Extensions` attribute in the `EventAgentNotReady` message. If a call is distributed to an agent via an ACD queue, the agent is placed in the `Not Ready` state and the call is diverted to the same ACD queue (at the end of the queue).

shutdown-sip-reject-code

Default Value: `603`
Valid Values: `300–603`
Changes Take Effect: Immediately

Specifies the error response used for rejecting new `INVITE` messages received by the system that is in shutdown mode. If set to `300`, `301`, or `302`, SIP Server first checks to see if `dr-peer-trunk` is configured, and if so, sends the contact of that `Trunk` DN in the `302` response.

Note: If a PSTN provider has alternative paths for call delivery, such as via Genesys SIP Server located at another site or through telephony infrastructure of a third party, set this option to a value of `503`.

silence-detected

Default Value: `drop`
Valid Values:

`drop`	The call is released.
`connect`	The connected call remains connected.

Changes Take Effect: Immediately
Related Feature: “Outbound IP Solution Integration” on page 306

Specifies the behavior of SIP Server where CPD is operating and silence is detected on the destination of a predictive call. SIP Server provides the CPD result in the following ways, depending on the value of the option and the type of call flow:

- When `silence-detected` is set to `drop`, SIP Server drops the call and generates an `EventReleased` with `CallState=CallStateSilenceDetected`.
- When `silence-detected` is set to `connect`, and the `TMakePredictiveCall` is invoked for a Trunk Group-based call, SIP Server establishes the call and generates an `EventEstablished` with `CallState=CallStateSilenceDetected`.

- When `silence-detected` is set to `connect`, and the `TMakePredictiveCall` is invoked for a Routing Point-based call, SIP Server establishes the call and generates an `EventQueued` with `CallState=CallStateSilenceDetected` and UserData `AnswerClass=SILENCE` attached to the event.

silence-tone

Default Value: `music/silence`
Valid Values: Name and path of any valid audio file
Changes Take Effect: Immediately for all new calls

Specifies the audio file to be played for the `Silence` treatment.

sip-3pcc-from-pass-through

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call
Related Feature: “Dial Plan” on page 195

Specifies the value that SIP Server includes as the username part of the `From` header in the `INVITE` message sent to the origination device. If set to `true`, SIP Server includes the `AttributeOtherDN` value from a `TMakeCall`, `TInitiateTransfer`, or `TInitiateConference` 3pcc request. If set to `false`, SIP Server includes the resulting digits when the dial plan is applied.

sip-491-passthrough

Default Value: `false`
Valid Values: `true, false`
Changed Take Effect: Immediately

Specifies whether SIP Server will forward `491 Request Pending` messages sent in response to a re-`INVITE` request to the remote endpoint. This operation mode should be used in the environments where SIP message conflict resolution is preferred to be carried out by the endpoints and not by SIP Server.

sip-add-contact-early-dialog

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

Specifies whether SIP Server adds the `Contact` header to unreliable `180` SIP messages. If set to `true`, SIP Server adds the `Contact` header to unreliable `180` SIP messages. If set to `false`, SIP server does not add the `Contact` header to unreliable `1xx` SIP messages (to provide backward compatibility).

sip-add-local-contact-user

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When set to `true`, SIP Server takes the username from the URI of an incoming `INVITE` request and adds it to the local contact of the created SIP dialog. As a result, the SIP response that establishes the dialog (such as a `2xx` to `INVITE`) as well as consecutive responses and new requests within the dialog from SIP Server will have that username inside the `Contact` header.

sip-address

Default Value: `NULL`
Valid Values: Any valid IP address or host name
Changes Take Effect: After SIP Server restart

Specifies an IP address of the SIP Server interface. This option must be set when deploying SIP Server on a host with multiple network interfaces. SIP Server uses this value to build the `Via` and the `Contact` headers in SIP messages. When this option is not set, SIP Server attempts to detect the IP address automatically.

sip-address-srv

Default Value: No default value
Valid Values: Valid Fully Qualified Domain Name (FQDN)
Changes Take Effect: After SIP Server restart
Related Feature: “DNS Name Resolution” on page 217

When specified, SIP Server can use this FQDN as its own contact for the DNS name resolution procedure.

Attention! This value will be inserted in the `Contact` and `Via` header fields of all outgoing SIP messages.

---

sip-alert-info

Default Value: No default value
Valid Values: Any string
Changes Take Effect: Immediately
Related Feature: “Alternate Ringtones” on page 102

Specifies the content to be added to the `Alert-Info` header used to trigger alternate ringtones or auto-answer functionality in the destination endpoint.

If configured, SIP Server will include the `Alert-Info` header with the value of this option whenever it sends an `INVITE` to any `Extension` or `ACD Position` DN on the switch—unless a different value is configured at the DN-level, or in the `SIP_HEADERS` extension.

For example, the following value points the endpoint to a ringtone file that can be used for internal calls:

`<http://www.provider.com/tones/internal_caller.pcm>`

Note: The URI must be enclosed in angle brackets.

If alternate ringtones are also configured for external or consultation calls (`sip-alert-info-external` or `sip-alert-info-consult`), then that configuration takes precedence over `sip-alert-info`.

sip-alert-info-external

Default Value: No default value
Valid Values: Any string
Changes Take Effect: Immediately
Related Feature: "Alternate Ringtones" on page 102

Specifies the content to be added to the `Alert-Info` header for inbound external calls. If configured, SIP Server will include the value of this option in the `Alert-Info` header of the `INVITE` messages that it sends for an external call to any `Extension` or `ACD Position` DNs in the switch—unless a different value is configured at the DN-level or in the `SIP_HEADERS` extension.

For example, the following value points the endpoint to the ringtone file that will be used for external calls:

`<http://www.provider.com/tones/internal_caller.pcm>`

Note: The URI must be enclosed in angle brackets.

sip-alert-info-consult

Default Value: No default value
Valid Values: Any string
Changes Take Effect: Immediately
Related Feature: "Alternate Ringtones" on page 102

Specifies the content to be added to the `Alert-Info` header for consultation calls. If configured, SIP Server will include the value of this option in the `Alert-Info` header of the `INVITE` messages that it sends to establish a consultation call with any `Extension` or `ACD Position` DNs in the switch—unless a different value is configured at the DN-level or in the `SIP_HEADERS` extension.

For example, the following value points the endpoint to the ringtone file that will be used for external calls:

`<http://www.provider.com/tones/consultation_call.pcm>>`

Note: The URI must be enclosed in angle brackets.

sip-answer-mode

Default Value: An empty string
Valid Values: `Auto`
Changes Take Effect: Immediately

Specifies the content to be added to the `Answer-Mode` header that is used to trigger the auto-answer functionality in the destination endpoint. SIP Server sends this header regardless of whether an endpoint has advertised support for the "answermode" sip.extension in the contact of a `REGISTER` message. If this option is configured, SIP Server includes the `Answer-Mode` header with the value of this option whenever it sends an initial `INVITE` message.

Notes:
- This option can also be configured at the DN level. The DN-level setting takes precedence over the Application-level setting.
- Support of the `Answer-Mode` SIP header in `Auto` mode as described in RFC 5373; compatible with Avaya 96xx phones. Avaya phones send `INVITE` messages without a `Referred-By` header in response to `REFER` from SIP Server; therefore, the `refer-enabled` configuration option must be set to `false`. Also, for Avaya phones, the `dual-dialog-enabled` configuration option must be set to `true` and the `sip-cti-control` configuration option should not be configured.

sip-block-headers

Default Value: An empty string
Valid Values: A comma-separated list of the headers to be filtered out during `INVITE` message propagation
Changes Take Effect: Immediately

Specifies a way to filter out headers during `INVITE` message propagation. With an empty string, no headers will be filtered out.

sip-call-id-suffix

Default Value: `sip-host`
Valid Values: `sip-host`, `sip-switch`, `sip-application`, any string, or empty
Changes Take Effect: After SIP Server restart

Defines the suffix that SIP Server inserts in the `Call-ID` header after the @ (at) character when SIP Server generates the `INVITE` message, as follows:

- If this option is set to `sip-host` (the default), SIP Server inserts the SIP listener IP address.
- If this option is set to `sip-switch`, SIP Server inserts the name of the Switch object.
- If this option is set to `sip-application`, SIP Server inserts the name of the SIP Server application.
- If this option is set to any other string of characters, SIP Server inserts that string as is.
- If the value is empty, SIP Server does not insert anything after the @ character in the `Call-ID` header.

sip-continue-treatment-on-call-reject

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Controls the behavior for the continuous treatment in the following scenario:

A call is returned to the Routing Point:

- `sip-treatments-continuous` is set to `true` and `divert-on-ringing` is set to `false`, and
- a continuous treatment is applied to a call, and
- the call is routed to an agent, but the agent rejects the call with the `TReleaseCall` or `TRedirectCall` operation before answering it.

In the case of `TRedirectCall`, the call is sent to a new destination without returning to the Routing Point.

When the option is set to `true` (the recommended setting), the continuous treatment that is already applied is not interrupted in both scenarios above.

When the option is set to `false` (the default setting to ensure backward compatibility), the continuous treatment is terminated in the scenario above as soon as `TReleaseCall` or `TRedirectCall` is received.

sip-disable-via-srv

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: “SRV Address Support in Contact and Record-Route Headers” on page 365

When set to `true`, SIP Server inserts a value of the `sip-address` option in the `Via` header. This option applies when the `sip-address-srv` option is configured.

sip-dtmf-send-rtp

Default Value: `false`
Valid Values:

`true`	SIP Server instructs Media Server to send DTMF tones to all call participants using one or both of the following DTMF generation methods: RTP packets with Named Telephone Event (NTE) payload as specified by RFC 2833, and in-band audio tones according to ITU-T Recommendation Q.23.
`false`	The feature is disabled.

Changes Take Effect: Immediately
Related Feature: “DTMF Tones Generation on Media Server” on page 222

Specifies whether SIP Server instructs Media Server to send DTMF tones when a T-Library client issues a `TSendDTMF` request.

sip-elin-timeout

Default Value: `1200`
Valid Values: `0–3600`
Changes Take Effect: On the next call
Related Feature: RedSky E911 Manager integration in the *SIP Server Integration Reference Manual*

Specifies the time interval, in seconds, for SIP Server to keep in memory the association between a 911 caller and the Emergency Location Identification Number (ELIN) assigned to the caller. If a call arrives at that ELIN before the timeout expires, the call is sent to the associated 911 caller DN. If within this time interval there are several emergency calls with the same ELIN, SIP Server directs the callback to the latest caller.

sip-enable-aoc-after-established

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: For the next request
Related Feature: “Providing AoC Notifications for Established Calls” on page 102

When this option is set to `true`, it enables the mode of providing Advice of Charge (AoC) notifications for established calls. In particular, SIP Server accepts and processes `TPrivateService(3018)` AoC requests in which `AttributeThisDN` refers to a Routing Point DN that is not present in the call. At the same time, the `AOC-Destination-DN` extension key points to an existing party in the established state. To successfully process this request, the Routing Point DN referred by `AttributeThisDN` must also have the `divert-on-ringing` option set to `false`.

When this option is set to false (for backward compatibility), SIP Server rejects an AoC `TPrivateService(3018)` request if `AttributeThisDN` refers to a DN not present in the call.

sip-enable-call-info

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Features: “Providing Call Participant Info” on page 336, “Switching Between Supervision Modes” on page 142

If set to `true`, SIP Server does the following:

- Distributes information about call participants except their locations and the supervisor-related information (see the `sip-enable-call-info-extended` option) to logged-in agents by using the SIP `NOTIFY` method and `EventUserEvent` messages.
- Distributes an `EventPrivateInfo(4024)` message, with the `MonitorMode` key in `AttributeExtensions`, to a supervisor and agent DNs indicating that the monitoring mode was changed.

If set to `false`, SIP Server does not distribute an `EventPrivateInfo(4024)` message when the monitoring mode changes.

sip-enable-call-info-extended

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Features: “Providing Call Participant Info” on page 336

This option applies only when `sip-enable-call-info` is enabled. When this option is set to `true`, SIP Server generates the supervisor information (`LCTSupervisor<n>` key-value pairs) and the location of call participants (`LCTParty<n>_location`) in `EventUserEvent`.

sip-enable-gdns

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: After SIP Server restart
Related Feature: “DNS Name Resolution” on page 217

Specifies the DNS resolution mode. If you set this option to `true`, SIP Server uses its internal DNS client to connect to a DNS server available on the network to use its conversion services.

If no DNS server is available, set this option to `false`. In this case, SIP Server resolves the domain names using local operating system utilities.

If set to `false`, SIP Server is unable to perform DNS resolution for SRV records with contacts that are missing port information (indicating the need to use SRV). Instead, ‘A’ record resolution and default ports will be used. The default port for UDP/TCP is `5060`, while the default for TLS is `5061`.

sip-enable-100rel

Default Value: `true`
Valid Values:

`true`	SIP Server advertises support for `100rel`, and requires it whenever the other side indicates support.
`false`	SIP Server does not negotiate support for the reliability of provisional responses.

Changes Take Effect: Immediately

If set to `true`, SIP Server places the option tag `100Rel` inside the `Supported` header of outgoing initial `INVITE` requests. This informs SIP clients that SIP Server is able to process provisional responses reliably.

sip-enable-ivr-metadata

Default Value: No default value
Valid Values: `true, false`
Changes Take Effect: On the next call

This option is used for IVR recording call scenarios. Specifies whether SIP Server passes its Application name in the initial `INVITE` message (in the `X-Genesys-sipsAppName` header) to Media Server. If this option is set to `true`, SIP Server includes its Application name in the custom header of the `INVITE` that it sends to Media Server. It also enables the default behavior of the feature depending on the DN type, as follows:

- `Voice over IP Service (msml), Trunk Group`, and `Voice Treatment Port`—SIP Server sends the custom header.
- `Trunk`—SIP Server does not send the custom header.

If this option is set to `false`, SIP Server does not include its Application name in the initial `INVITE` sent to Media Server.

Notes:
- If the IVR recording feature is enabled, it is not required to explicitly enable recording by setting the `record` option to `true` on DNs representing GVP, such as `Trunk, Trunk Group,` or `Voice Treatment Port`. Recording is started by the VXML application running on the Media Server.
- This option can also be configured at the DN level. The DN-level setting takes precedence over the Application-level setting.

sip-enable-moh

Default Value: `false`
Valid Value:

`true`	Music-on-hold is enabled.
`false`	Music-on-hold is disabled.

Changes Take Effect: At the next `Hold/THoldCall` operation
Related Feature: “Customizing Music on Hold and in Queue” on page 179

Enables or disables music-on-hold.

Note: This option can also be configured at the DN level. The DN-level setting takes precedence over the Application-level setting.

sip-enable-rfc3263

Default Value: `false`
Valid Value: `true, false`
Changes Take Effect: After SIP Server restart
Related Feature: “DNS Name Resolution” on page 217

Specifies the DNS resolution mode.

If you set this option to `true`, SIP Server includes priority and weight information from the Returned Record Set (resolved from the `contact` option using the internal DNS) when it applies the destination selection procedure. This is in accordance with RFC 3263 and RFC 2782. If set to `true`, SIP Server ignores the values of the `contacts-backup` option as redundant.

If you set this option to `false`, SIP Server does not factor in priority or weight from the RR Set when applying the destination selection procedure (multiple active destinations are given equal ranking). The destinations in this case are taken from the URIs configured in the following DN options:

- `contact`
- `contacts-backup` (can be several URIs in a comma-separated list)

The Active Out-of-Service Detection procedure uses DNS SRV/A Resource Records resolution for composing a list of transports (protocol, IP address, port) for each contact’s URI. To use priority and weight, set this option to `true`. To treat all destinations as equal, set this option to `false`.

sip-enable-sdp-application-filter

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether SIP Server modifies the SDP message body during SIP negotiation. When set to `true`, SDP with media-type `(m=) "application"` will be filtered.

sip-enable-tcp-keep-alive

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After restart
Related Feature: “Keep Alive for TCP Connections” on page 259

When set to `true`, enables the TCP keep-alive mechanism for all SIP-related connections. Keep-alive timeouts are configured on the operating system level.

sip-enable-two-party-mute

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Muting/Unmuting a Party in a Conference" on page 167

When set to `true`, enables muting and unmuting parties in two-way calls via a T-Library request; requires MSML to be enabled.

Note: When set to `true`, two-party conferences are not be converted to direct calls.

sip-enable-x-genesys-route

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately
Related Feature: "SRV Address Support in Contact and Record-Route Headers" on page 365

Specifies if SIP Server adds the private `X-Genesys-Route` header to SIP messages when deployed with SIP Proxy. This is for backward compatibility to disable new functionality in old deployments.

sip-error-conversion

Default Value: No default value
Valid Values: A comma-separated list of value pairs: <received error code>=<converted error code>, `0`=<converted error code> (for example: `408=486`, `0=486`)
Changes Take Effect: Immediately

When this option is set to <received error code>=<converted error code>, SIP Server converts the received error response code to the configured code and sends the converted SIP response code to the origination device. This setting affects the following:

- How the SIP error code is processed by SIP Server.

  For example, SIP error code `486 (Busy Here)` means a destination is busy. SIP error code `408` (`Request Timeout`) received for DNs located behind a softswitch places a DN in out-of-service state. If this option is set to `408=486` and the softswitch responds with the `408` error code, SIP Server will not place the DN in out-of-service state.

- The `ErrorCode` that is returned in `EventError` to a routing application when a routing attempt is unsuccessful.

  For example, when a routing destination responds to the `INVITE` message with code `484` (`Address Incomplete`), SIP Server sends `ErrorCode 231` (`DN is Busy`) to a routing application. If this option is set to `484=404`, SIP Server returns `ErrorCode 71` (`Invalid Called DN`) to a routing application.

When this option is set to `0`=<converted error code>, SIP Server sends the converted SIP error code if one of the following occurs:

- The destination device fails to respond to the incoming `INVITE` message.
- No active DN is found for SIP Server to send a call.

Note: If a destination DN of type `Extension` fails to respond to the incoming `INVITE` message, SIP Server places the `Extension` DN in out-of-service state regardless of the `sip-error-conversion` setting.

The option can be configured at the following levels and in the following order of precedence:

1. DN level
2. Application level

sip-enable-sdp-codec-filter

Default Value: `false`

Valid Values:

`true`	SIP Server modifies the SDP message body during SIP renegotiation.
`false`	SIP Server does not modify the SDP message body.

Changes Take Effect: On the next call

Related Option: `audio-codecs`

Specifies whether SIP Server modifies the SDP message body during SIP renegotiation. All codecs that are not in the list of values for the `audio-codecs` option are deleted from the SDP. As a result, all call center audio traffic is established based on the codecs listed in the `audio-codecs` option.

You can also specify this option at the DN-level. If `sip-enable-sdp-codec-filter` is set to `true` in the DN configuration, SIP Server, as it propagates the SDP to and from the device represented by this DN, will use as its list of available codecs the value configured in the `audio-codecs` option on the DN rather than on the application. If `sip-enable-sdp-codec-filter` is set to `true` at both the application and the DN level, then the `audio-codecs` configured in the DN should contain a subset of the `audio-codecs` configured in the application.

Note: Currently, SIP Server does not support filtration of video codecs.

sip-enable-strict-auth

Default Value: `false`

Valid Values: `true, false`

Changes Take Effect: For the next new call or `REGISTER` request

Enables SIP Server in SIP Cluster mode to mandate authorization of internal devices on `REGISTER` and `INVITE` requests. To register or establish communication, devices must not use empty passwords or passwords equal to

the DN name. When this option is set to `false`, an internal device can register or establish communication with SIP Cluster without any authorization.

You can define the `sip-enable-strict-auth` option at the following levels listed in order of priority:

1. A DN of type `Extension` that is not behind a softswitch
2. A SIP Server Application
3. A SIP Cluster Node VOIP Service DN (`service-type=sip-cluster-nodes`)

sip-enhance-diversion

Default Value: `false`
Valid Values: `false, true`
Changes take effect: On the next call
Related Feature: "Handling Call Forwarding Loop" in the *8.1 SIP Server Integration Reference Manual*

Specifies how SIP Server processes an `INVITE` message based on the value of the `Diversion` header. If the option is set to `true` and the `Diversion` header references the call forwarding party, SIP Server rejects that `INVITE`, waits until that rejection is propagated by the PBX back to the SIP Server in the original SIP dialog, and sends a new `INVITE` message to a forwarding destination.

sip-error-codes-overflow

Default Value: An empty string (or `503` error code)
Valid Values: A list of patterns for numeric error codes separated by a comma (,). Letter X in a pattern represents any digit. A single pattern must start with a digit and contain all 3 digits, and a pattern containing X should conclude the pattern's list, if present. Examples:

- `503`
- `503,504`
- `487,50X`
- `487,5XX`
- Patterns `5X3, XXX` are invalid

Changes Take Effect: On the next call

When, on an initial `INVITE` message, SIP Server receives a negative response containing the error code that matches the option value setting, SIP Server attempts to find an alternative trunk or softswitch to initiate an outbound call.

In addition, SIP Server can attempt to connect to a DN via an alternative softswitch (found in the DN configuration) if the first attempt to connect to a DN via a softswitch resulted in a negative response from that softswitch.

sip-filter-media

Default Value: No default value
Valid Values: `video`
Changes Take Effect: Immediately
Related Feature: "Video Blocking" on page 381

When set to `video`, SIP Server blocks video media streams in calls.

Note: This option can also be configured at the DN level. The DN-level setting takes precedence over the Application-level setting.

sip-from-pass-through

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

Specifies whether SIP Server will use the content of the `From` header from the original `INVITE` to generate the content for the `From` header in the outgoing `INVITE` message.

When set to `true`, this option takes precedence over any cpn-controlling option or the `CPNDigits` key in `AttributeExtensions` of a T-Library request.

sip-fqdn-ip-version

Default Value: `4`
Valid Values: `4, 6`
Changes Take Effect: After restart
Related Feature: "IPv6 Support" on page 257

Specifies the IP protocol when the peer's address is represented as an FQDN. If, in the environment, all SIP devices use IPv4, set this option to `4`. If all SIP devices use IPv6, set this option to `6`.

sip-hold-rfc3264

Default Value: `false`
Valid Value:

`true`	RFC3264-compliant implementation.
`false`	RFC2543-compliant implementation.

Changes Take Effect: On the next call

Specifies which implementation of hold media SDP is used by SIP Server for third-party call control (3pcc) hold operations.

Note: This option can also be configured at the DN level. The DN-level setting takes precedence over the Application-level setting.

sip-invite-timeout

Default Value: `0` (in effect, 32 seconds)
Valid Values: `0–34`
Changes Take Effect: Immediately

Specifies the number of seconds SIP Server waits for a response to the `INVITE` message. The call times out if no response is received. If set to `0`, or if a value is not specified, then the default SIP call timeout of `32` seconds is used.

SIP Server uses different timeout options for regular devices and media service devices, in order to correctly process scenarios where only a provisional response is received after sending an `INVITE` to a device (without receiving a final response). SIP Server treats the expiry of either timeout setting the same way it does an expiry of SIP Timer B.

`sip-invite-timeout`—For regular devices, used to specify the length of time that a SIP transaction can remain in the Proceeding state when the only provisional responses that it receives are `100 Trying` messages. Any other provisional message removes the timer, so that the regular device can remain in a ringing state until the peer's action causes SIP Server to cancel the `INVITE` request.

`sip-invite-treatment-timeout`—For media service devices, used to specify the length of time to wait for a final or reliable provisional response. If this timeout expires, the media service device is considered to be out of service and SIP Server tries to use an alternative device to perform the required function.

sip-invite-treatment-timeout

Default Value: `0` (in effect, 32 seconds)
Valid Values: `0–34`
Changes Take Effect: Immediately

Specifies the number of seconds SIP Server waits for a response to the `INVITE` message for a treatment (such as an announcement or music-on-hold). The call times out if no response is received. If set to `0`, or if a value is not specified, then the default SIP call timeout of `32` seconds is used.

Note: Setting this value to less than the session timer (Timer D - 32 seconds) allows a `MakeCall` operation to succeed when the ringtone is enabled (`ring-tone-on-make-call` set to `true`) and the media server is unavailable. If the value is greater than or equal to the session timer (or 0), then this scenario instead results in a failed call when the media server is unavailable.

sip-iptakeover-monitoring

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After SIP Server restart
Dependent Option: `sip-address`
Related Feature: See "Network Status Monitoring" in the *SIP Server 8.1 High-Availability Deployment Guide.*

For the Hot Standby IP Address Takeover configuration. When set to `true`, this option enables the Virtual IP address status monitoring. The Virtual IP address is taken from the `sip-address` option.

sip-ip-tos

Default Value: `256`
Valid Values: Any integer from `0-256`, either in decimal format or in hexadecimal format with `0x` prefix
Changes Take Effect: Immediately
Related Feature: "Quality of Service" on page 341

Defines the value of the Type of Service (TOS) byte in the IP header of SIP messages that are sent by SIP Server (if undefined, the operating system TOS is used). The default value (`256`) disables this functionality.

Depending on the network configuration, the TOS byte is treated as one of the following:

- 3-bit IP precedence field, followed by a 4-bit type-of-service. The least significant bit (LSB) is unused and set to 0. (RFC 1349)
- 6-bit DiffServ, with the two least significant bits unused. (RFC 2474)

For example, the following values may be used to assign a higher priority to SIP packets:

- 0x10—IPTOS_LOWDELAY, low-delay type of service
- 0x20—IPTOS_PREC_PRIORITY, priority precedence
- 0x40—IPTOS_PREC_CRITICAL, critical precedence
- 0xB8—DiffServ EF (Expedited Forward)

Notes: On most operating systems, applications that are running on behalf of non-privileged user accounts are not permitted to set a non-zero TOS value, so you might have to perform additional actions to enable this functionality. In particular:

- On Linux, the application must have `CAP_NET_ADMIN` capability (that is, run from the root account).
- On Windows, the following registry setting must be set (see also http://support.microsoft.com\kb/248611):

  ```
  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\
  Parameters\DisableUserTOSSetting = (DWORD) 0
  ```

Refer to operating system documentation for additional information.

sip-legacy-invite-retr-interval

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After SIP Server restart

Enables `INVITE` retransmissions in accordance with RFC 3261 "SIP: Session Initiation Protocol". If you set this value to `true`, SIP Server sends the `INVITE` retransmissions in periods of 0.5 - 1 - 2 -4 -4 -4 - 4...seconds (legacy SIP Server behavior). If set to `false`, SIP Server sends the `INVITE` retransmissions in periods of 0.5 - 1 - 2 - 4 - 8 - 16... seconds, as per RFC 3261.

sip-link-type

Default Value: `0`
Valid Values: `0, 3, 4`
Changes Take Effect: After SIP Server restart
Related Feature: "Multi-Threaded Logging" on page 280

Specifies whether SIP Server will run in multi-threaded mode, or in single-threaded mode for backward compatibility.

Configure the valid values for this option as follows:

`0` (default) SIP Server runs in single-threaded mode, as in pre-8.0.3 releases, with the Main thread processing T-Library requests, distributing Events, managing SIP calls, and processing SIP signaling

`1, 2` Reserved for debugging

`3` Enables multi-threaded mode with the following threads:

- T-Server thread—processes T-Library requests and distributes Events
- Call Manager thread—manages SIP calls and processes SIP signaling (except `OPTIONS` messages)
- Service Checker thread— performs Active Out-of-Service Detection (`OPTIONS` messages)

`4` Enables multi-threaded mode designed for IMS double-dip deployments, with the following threads:

- T-Server thread
- 16 Call Manager threads
- Service Checker thread
- Presence Manager thread

In both single-threaded and multi-threaded modes, SIP Server runs the following threads:

- SIP transport layer thread to dispatch SIP messages
- Operational Information thread to collect and report statistics; to perform NIC monitoring
- A number of auxiliary threads

Note: For an HA configuration, this option must be configured with the same value in both primary and backup SIP Servers.

sip-max-uui-length

Default Value: `256`
Valid Value: `0-8192`
Changes Take Effect: Immediately
Related Feature: "User to User Information (UUI)" on page 379

Specifies the maximum number of characters by which SIP Server limits the length of the data included in the User-to-User Information (UUI) header. For example, with the default value of `256`, SIP Server will limit the length of UUI content, with hexadecimal encoding, to 128 bytes.

sip-max-retry-listen

Default Value: `15`
Valid Values: `0-65535`
Changes Take Effect: After SIP Server restart

Specifies the number of times SIP Server retries opening its listening port per time interval after the CTI link is disconnected. The time interval starts at 1 attempt per second, maximizing at 1 attempt every 30 seconds, after which SIP Server continues retrying every 30 seconds indefinitely.

sip-nic-address

Default Value: `NULL`
Valid Values: Any valid IP address or FQDN
Changes Take Effect: After SIP Server restart
Dependent Option: `sip-nic-monitoring`
Related Feature: See "Network Status Monitoring" in the *SIP Server 8.1 High-Availability Deployment Guide.*

This option can be set in deployments with dedicated SIP NICs (network interface cards) where the SIP traffic is separated from the T-Library network traffic. This option specifies the IP address of the NIC that belong to the host where the SIP Server runs and is used for SIP traffic. This IP address must always be present on this host regardless of the role of SIP Server (primary or backup). For the IP Address Takeover configuration, its unique IP address is associated with the SIP NIC, not the Virtual IP address.

sip-nic-monitoring

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After SIP Server restart
Dependent Option: `sip-nic-address`
Related Feature: See "Network Status Monitoring" in the *SIP Server 8.1 High-Availability Deployment Guide.*

When set to `true`, this option enables the SIP NIC IP address status monitoring. The SIP IP address is taken from the `sip-nic-address` option.

sip-outbound-proxy

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After SIP Server restart

If set to `true`, all SIP messages are sent through SIP Proxy. SIP Server looks for a VoIP Service DN with `service-type=sip-outbound-proxy`. For each initial out-of-dialog outgoing SIP request, SIP Server inserts a `Route` header with the value of the DN contact.

Note: SIP Server sends the `REGISTER` message directly to the `Trunk` DN configured with the `force-register` option, instead of SIP Proxy.

sip-pass-check

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Features: “Endpoint Service Monitoring” on page 239, “SIP Traffic Monitoring” on page 355

When set to `true`, enables tracking of SIP messages that reach the primary SIP Server, including responses from SIP devices configured for Active Out-of-Service Detection.

The primary SIP Server reports the `SERVICE_UNAVAILABLE` status to LCA/SCS when all devices configured with the Active OOS check have failed and no other SIP messages have been received for a period of time. The period of time is calculated as the maximum of the sums of the `oos-check` and `oos-force` option values configured for service DNs (if `oos-force` is less than `5`, `5` is used). When SIP Server reports the `SERVICE_UNAVAILABLE` status to LCA/SCS, SCS switches the primary SIP Server to the backup role, and SIP Server reports the `SERVICE_RUNNING` status to LCA/SCS. The backup SIP Server becomes the primary, and starts monitoring SIP traffic.

Note: If both the primary and backup servers receive no SIP traffic, a switchover would occur each time the effective out-of-service timeout expires. To prevent frequent switchovers, SIP Server detects the “double switchover” condition and doubles the effective out-of-service timeout each time the double switchover happens, up to two times, or until one of the two servers detects SIP traffic. As soon as SIP traffic is detected, the server that detected the traffic remains the primary SIP Server and continues normal operation.

sip-pass-from-parameters

Default Value: No default value
Valid Values: A comma-separated list of parameters, or *
Changes Take Effect: Immediately

Specifies which parameters in the `From` header SIP Server will pass through in the outgoing `INVITE` message—with the exception of the `tag` parameter generated by SIP Server. To pass through all parameters, use the asterisk (`*`) value.

sip-pass-refer-headers

Default Value: No default value
Valid Values: A string of SIP headers separated by commas; may contain full header names or name parts with an asterisk representing a subset of headers
Changes Take Effect: At the next established call

When specified, SIP Server will pass custom SIP headers from a `REFER` request to an outgoing `INVITE` or `REFER` request.

For example:

If the `sip-pass-refer-headers` option is set to `X-Tellme-*,X-Information` and the incoming REFER request contains the following headers:

- `X-Tellme-Session-ID`
- `X-Tellme-Header`
- `X-Information`

Then SIP Server will include all three headers to an outgoing `INVITE` or `REFER` request.

sip-port

Default Value: `5060`
Valid Value: Any valid TCP/IP port
Changes Take Effect: After SIP Server restart

Specifies the port on which SIP Server listens for incoming SIP requests. The same port number is used for both TCP and UDP transports.

sip-port-tls

Default Value: No default value
Valid Values: Any valid port number
Changes Take Effect: After SIP Server restart
Related Feature: “Transport Layer Security for SIP Traffic” on page 367

Specifies the port on which SIP Server listens for incoming requests using TLS encryption. To disable TLS transport for SIP traffic altogether, set this option to `0`.

sip-preserve-contact

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately with the exception (see the option description)

Specifies whether SIP Server preserves session information (a cookie) that is appended to the user-info part of the `Contact` header in `REGISTER` requests. If you set this option to `true`, SIP Server preserves the cookie from the `REGISTER` request, and then includes the cookie in the `Request-URI` of the outgoing `INVITE` request.

The `sip-preserve-contact` option affects DNs that contain the username in the `contact` option. If the `sip-preserve-contact` option is set to `true`, SIP Server uses the username from the configured contact in the `Request-URI` of an outgoing `INVITE` message. If you change the `sip-preserve-contact` option value, Genesys recommends restarting SIP Server for changes to take effect.

Genesys does not recommend using the `sip-preserve-contact` option at the DN level if a DN has the `contact` option containing a username.

Note: This option can also be configured at the DN level. The DN-level setting takes precedence over the Application-level setting.

sip-proxy-headers-enabled

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether SIP Server will proxy the custom SIP headers or not. If set to `true`, SIP Server will proxy the custom headers, when sending the message to other side. SIP Server will proxy the custom headers only in the following SIP messages:

- `INVITE`
- `REFER`
- `200 OK`

If set to `false`, SIP Server will not proxy custom headers, when sending the message to other side. This parameter can be defined either at the application or a DN level with the DN-level parameter having a higher priority.

Note: For integration with GVP, the Resource Manager must be configured as a Trunk for supporting the above option in the `REFER` message.

sip-recovery-allow-userdata

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Enables mapping of T-Library User Data into a SIP `INVITE` message which is sent to Media Server during recovery of the `PlayApplication` treatment.

If set to `true`, during recovery after failure of the `PlayApplication` treatment (an initial `INVITE` is rejected by a SIP error response), SIP Server does mapping of T-Library User Data into SIP INVITE message.

If set to `false`, no user data is mapped in the `INVITE` message during recovery.

sip-referred-by-support

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: “Referred-By Header Support” on page 172

If set to `true`, SIP Server sends an identity of the party which has originated the transfer, in the SIP URI of the outgoing `REFER` request’s `Referred-By` header. If the call is processed on the Routing Point and is routed using `TRouteCall`, then the Routing Point name is used as a userpart of the `Referred-By` SIP URI. In addition, SIP Server can substitute the “hostport” component of the SIP URI in `Refer-To` and `Referred-By` headers of `REFER` messages with the values configured by a user. If set to `false`, this feature is disabled.

sip-referxfer-bye-timeout

Default value: `0`
Valid Values: `0`–`65535`
Changes Take Effect: Immediately

Specifies the time interval, in milliseconds, SIP Server waits before releasing the initial dialog after a single-step transfer using the `REFER` method is completed.

sip-registrar-allowlist

Default Value: An empty string
Valid Values: A string
Changes Take Effect: At the next registration request
Related Feature: “Strict SIP Endpoint Registration” on page 366

This option contains a list of IP addresses, separated by a semicolon (;). An empty value means that this functionality is disabled.

Each entry in the list can be in one of the following forms:

- FQDN
- IP address
- IPv4 CIDR block (in the form of *a.b.c.d/n*)

For the FQDN and IP address entries, SIP Server makes an exact match of the entry to the address extracted from the `REGISTER` request. For a CIDR block, SIP Server takes *n* bits starting from the left of the address and matches them against *n* left bits of the entry. For example, to accept the range of 255 addresses from 192.0.2.0 to 129.0.2.255, the entry in the list must be as follows: 192.0.2.0/24.

sip-registrar-allowlist-origin

Default Value: `via`
Valid Values: `via, contact`
Changes Take Effect: At the next registration request
Related Feature: “Strict SIP Endpoint Registration” on page 366

Defines a `REGISTER` message header from which SIP Server takes an IP address to match against a list of IP addresses defined by the `sip-registrar-allowlist` option.

sip-registrar-reject-code

Default Value: `403`
Valid Values: A valid SIP error code in the range of `400-599`
Changes Take Effect: At the next registration request
Related Feature: “Strict SIP Endpoint Registration” on page 366

Defines an error response that SIP Server sends if a SIP endpoint's IP address in its `REGISTER` request does not match the one defined in the trusted IP addresses list (in the `sip-registrar-allowlist` option).

sip-rel-200-retransmit

Default Value: `false`
Valid Values: `false, true`
Changes Take Effect: On the next call

Specifies if SIP Server retransmits `200 OK` in response to an `INVITE` message on reliable transports if `ACK` is not received for `200 OK`. (The default value of this option (`false`) enables the previous behavior in SIP Server.)

Note: This option can also be configured at the DN level. The DN-level setting takes precedence over the Application-level setting.

sip-release-call-on-disable-dn

Default Values: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

Specifies whether SIP Server releases all calls for a DN that was disabled in the configuration environment. If set to `true`, SIP Server releases both call’s dialogs (T-Library and SIP) for the disabled DN.

sip-remote-del-from-conf

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Deleting Party From Conference in Multi-site Deployments" on page 165
Related Option: `sip-enable-call-info`

In multi-site deployments, when this option is set to `true`, SIP Server processes a `TDeleteFromConference` request to remove a remote party (specified in `OtherDN`) from a conference. The `OtherDN` attribute of the `TDeleteFromConference` request must contain the party ID received in the `LCTParty` list. When this option is set to `false`, this feature is disabled.

sip-replaces-mode

Default Value: `0`
Valid Values: `0, 1, 2, 3`
Changes Take Effect: On the next call
Related Feature: "TCompleteTransfer using REFER or REFER with Replaces" on page 171

Specifies the SIP method used by SIP Server to complete a two-step transfer.

- With a value of `0`, SIP Server uses the `REFER` method if the `transfer-complete-by-refer` option is set to `true`.
- With a value of `1`, SIP Server uses the `REFER` method with `Replaces` if the `Allow` header contains `REFER` as a supported method and the `Supported` header contains `Replaces`. If `REFER` with `Replaces` is not supported by a device, then `TCompleteTransfer` will be performed using the `REFER` method. If a device does not support the `REFER` method, then the transfer will be completed using the re-`INVITE` method.
- With a value of `2`, SIP Server uses the `REFER` method with `Replaces` to process `TCompleteTransfer`. The `Allow` and `Supported` headers will not be analyzed.
- With a value of `3`, when the DN-level `sip-server-inter-trunk` option is set to `true`, SIP Server uses the re-`INVITE` method instead of the `REFER` method for transfers and call routing.

Note: For this functionality to work, the `refer-enabled` option must be set to `true` in the DN from which a call party is transferred to another destination during a two-step transfer.

sip-respect-privacy

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

Specifies the content SIP Server will report in `AttributeANI` when an inbound `INVITE` message contains `P-Asserted-Identity` and `Privacy:id` headers. If this option is set to `false` (the default), the content of the `P-Asserted-Identity` header is ignored when determining `AttributeANI`. If this option is set to `true`, the content for `AttributeANI` is taken from the `P-Asserted-Identity` header. If the `P-Asserted-Identity` header is not present, then the content for `AttributeANI` is determined as follows:

- From the user part of the `From` header, if present and not anonymous.
- From the user part of the `Contact` header, if present and not anonymous.
- From the matched origination device DN.
- Set to anonymous in all other cases.

sip-retry-after

Default Value: `0`
Valid Values: Any integer from `0-30`
Changes Take Effect: On the next call

Specifies the value of the `Retry-After` header, in seconds, that SIP Server inserts in the error response to an incoming re-`INVITE` or `REFER` message, which is received while a dialog with Media Server is in progress. If set to `0`, SIP Server does not insert the `Retry-After` header.

sip-retry-timeout

Default Value: `30`
Valid Values: `1-3600`
Changes Take Effect: Immediately

Specifies the time interval, in seconds, after which SIP Server initiates a new subscription if the previous `SUBSCRIBE` dialog is terminated.

sip-ring-tone-mode

Default Value: `0`
Valid Values: `0, 1, 2`
Changes Take Effect: Immediately

- When the option is set to `0`, SIP Server connects Media Server to a call to play an audio ringtone.
- When the option is set to `1`, SIP Server waits for a response from the called device, then connects Media Server to a call to play an audio ringtone, but *only* when the returned response cannot be used as the offer to a calling device.

- When set to 2, SIP Server plays an audio ring tone only to an inbound external call, by connecting Media Server, before the call is placed to an agent.

Notes:
- SIP Server does not support internal ringtones in conference scenarios where the sip-ring-tone-mode option is set to 1. In this case, SIP Server provides a ringtone only if the endpoint returns an SDP in the provisional message.
- This option can also be configured at the DN level. The DN-level setting takes precedence over the Application-level setting.
- To enable a ringback to be played to an external caller, set the following parameters in the Trunk DN:
  - ring-tone-on-make-call=true
  - sip-ring-tone-mode=1
  - refer-enabled=false

  Note: For an inbound call to a Routing Point, a ringback is played to an external caller only if a treatment was applied to the call at the Routing Point, and then the call is delivered to an available agent. If an agent was available immediately and the treatment was not applied to the call on the Routing Point, the ringback is not played to the caller on 180 Ringing from the agent's DN.
- When sip-treatments-continuous is set to true and sip-ring-tone-mode is set as 0 or 1, the ringtone is not played. For the ringtone to be played, set sip-treatments-continuous to false.

sip-resubscribe-on-nonotify

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

Specifies whether SIP Server, at startup or after a switchover, re-sends a SUBSCRIBE request to Resource Manager if it does not receive a NOTIFY message within a two-second timeout after a successful subscription. If set to true and SIP Server does not receive NOTIFY, it terminates the subscription and sends a new SUBSCRIBE request. SIP Server continues to re-subscribe until it receives NOTIFY. If set to false (the default), SIP Server does not take any additional action.

Note: For SIP Server deployments using F5 Networks BIG-IP LTM, set sip-resubscribe-on-nonotify to true.

sip-server-info

Default Value: No default value
Valid Values: A valid string or the special character `*`
Changes Take Effect: On the next call
Related Feature: “Enabling Server and User-Agent Headers” on page 185

Specifies the value of the `Server` header that SIP Server includes in all reply messages that it sends. The value for this option can contain the following placeholders:

- `$VERSION$` = will be replaced with the current SIP Server build
- `$APP-NAME$` = will be replaced with the name of the application in the environment

You can also use the special value *, which is equivalent to `Genesys SIP Server $VERSION$ ($APP-NAME$)`.

sip-timer-c-support

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After restart

Controls Timer C functionality as described in RFC 3261. If enabled, SIP Server applies the following logic to processing of the provisional responses it generates:

1. Start timer C when the first `1xx` (>`100`) response is sent out.
2. If timer activated, resend the last `1xx` response.

This functionality is disabled by default.

sip-tls-cert

Default Value: No default value
Valid Values: certificate `thumbprint`, or valid `path and filename`
Changes Take Effect: After SIP Sever restart
Related Feature: “Transport Layer Security for SIP Traffic” on page 367

For Windows, set this to the thumbprint obtained from the user certificate generated for the host.

For Solaris, Linux, or AIX, set this option to the path and filename of the `.pem` encoded file that contains the host certificate.

sip-tls-cert-key

Default Value: No default value
Valid Values: A valid path and filename
Changes Take Effect: After SIP Sever restart
Related Feature: “Transport Layer Security for SIP Traffic” on page 367

For Solaris, Linux, or AIX, deployments only. Specifies the path and filename of the `.pem` encoded file that contains the host private key.

Note: This option is only used when creating the initial connection. Changes to this option do not affect open connections (open connections are not closed).

sip-tls-cipher-list

Default Value: No default value
Valid Values: A colon-separated list of cipher suites, or cipher aliases, which includes the mandatory cipher suite `TLS_RSA_WITH_AES_128_CBC_SHA`
Changes Take Effect: After SIP Server restart
Related Feature: “Transport Layer Security for SIP Traffic” on page 367

Specifies the list of preferred cipher suites to be used in TLS for SIP traffic. SIP Server transfers the value of this option to the third-party software library that provides TLS. Configure the cipher list as a string of cipher operations, where each operation consists of an operator character (optional), followed by a name.

When creating the cipher list, you must follow these rules:

- Use valid cipher names or cipher aliases. Valid names can contain the characters `a-z`, `A-Z`, `0-9`, and a dash (`-`).
- Separate names and aliases in the list with a colon (`:`).
- Join multi-part names with the plus sign (`+`).
- Use the exclamation character (`!`) immediately after a separating colon to indicate a kill operation. The cipher following this exclamation mark becomes unavailable.
- Use the plus sign (`+`) immediately after the separating colon (`:`) to indicate an order operation. This moves the active cipher to the current position in the list of ciphers.
- Use the minus sign (`-`) immediately after the separating colon (`:`) to indicate a delete operation. The cipher following a minus sign becomes inactive (though it remains available for further operations).
- A non-operator character appearing immediately after a separating colon (`:`) indicates an add operation. If the cipher following the character is not currently active, the cipher is added as an active cipher to the end of the list of available ciphers.

Table 110 lists the primary cipher aliases.

Table 110: Primary Cipher Aliases

Alias	Description
kRSA, kDHr, kDHd and kEDH	Key exchange types
aRSA, aDSS, aNULL and aDH	Authentication
DES, 3DES, RC4, RC2 and eNULL	Ciphers
MD5 and SHA1	Message digests

Table 111 lists available cipher group aliases.

Table 111: Group Aliases

Alias	Description
SSLv2	All SSLv2 ciphers
SSLv3	All SSLv3 ciphers
EXP	All export ciphers
LOW	All low strength ciphers (no export ciphers, normally single DES)
MEDIUM	128-bit encryption
HIGH	Triple DES

Example: An example of a configured cipher list is as follows:

```
!ADH:RC4+RSA:HIGH:MEDIUM:LOW:EXP:+SSLv2:+EXP
```

In this example, the operator character and placement in the list instructs SIP Server to interpret the cipher string using the following sequence:

1. Does not consider any ciphers that do not authenticate.
2. Uses ciphers RC4 and RSA
3. Includes HIGH, MEDIUM, and LOW security ciphers.
4. Adds all export ciphers.
5. Places all SSLv2 and export ciphers to the end of the list.

sip-tls-crl

Default Value: No default value
Valid Values: Valid file name
Changes Take Effect: After SIP Server restart
Related Feature: “Transport Layer Security for SIP Traffic” on page 367

Specifies the name of the file that contains one or more certificates in `PEM` format, defining the Certificate Revocation List. As part of the authentication process, the system checks whether a presented certificate is included in this list of revoked certificates before completing authentication. This option applies only to UNIX operating systems.

sip-tls-sec-protocol

Default Value: `SSLv23`
Valid Values: `SSLv23, SSLv3, TLSv1, TLSv11, TLSv12`
Changes Take Effect: After SIP Server restart
Related Feature: “Transport Layer Security for SIP Traffic” on page 367

If configured, this option specifies the lowest version of TLS that SIP Server will use to send and accept secure connection requests with SIP devices.

This option can be used only on UNIX operating systems with Genesys Security Pack on UNIX 8.5.100.09 or later. The option has no effect on Windows. TLS versions are as follows:

- `SSLv3`—SSL version 3.0.
- `TLSv1`—TLS version 1.0.
- `TLSv11`—TLS version 1.1.
- `TLSv12`—TLS version 1.2.

If the option is not configured or set to `SSLv23` (the default, for backward compatibility), SIP Server uses the highest TLS version supported by Genesys Security Pack 8.5.1. Currently, it is TLS 1.2.

Refer to the *Genesys Security Deployment Guide* for details.

Notes:
- The `TLSv12` value is supported by SIP Server 8.1.102.25 or later and with Genesys Security Pack version 8.5.100.09 or later.
- Some of the older protocols might not be supported by latest Security Pack versions. Refer to the *Genesys Security Deployment Guide* for more details.

sip-tls-mutual

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After SIP Server restart
Related Feature: "Transport Layer Security for SIP Traffic" on page 367

Specifies whether SIP Server will request the client certificate and initiates a mutual TLS connection when SIP Server is in the TLS-Server role during authentication. To enable mutual TLS, set this option to `true`.

sip-tls-target-name-check

Default Value: `no`
Valid Values: `no`, `host`
Changes Take Effect: After SIP Server restart
Related Feature: "Transport Layer Security for SIP Traffic" on page 367

Specifies whether the `host` field in the server certificate will be compared to the target host name during the authentication process. If you set the value for this option to `host`, SIP Server tries to match the host field in the certificate with the target host name. If no match is found, the connection fails.

sip-tls-trusted-ca

Default Value: No default value
Valid Values: Valid path and filename
Changes Take Effect: After SIP Server restart
Related Feature: "Transport Layer Security for SIP Traffic" on page 367

For Solaris, Linux, or AIX, deployments only. Specifies the path and filename to a `.pem` encoded Certificate Authority (CA) file containing one or more certificates in PEM format.

Note: This option is only used when creating the initial connection. Changes to this option do not affect open connections (open connections are not closed).

---

sip-transfer-complete-timeout

Default Value: `0` (unlimited wait, for backward compatibility)
Valid Values: `0-34`
Changes Take Effect: Immediately
Related Option: `sip-transfer-complete-message`

Specifies how many seconds SIP Server waits for a `NOTIFY` message before considering the Out Of Signaling Path transfer as failed.

Note: While completing the Out Of Signaling Path transfer by the `REFER` method with the `Replaces` header, SIP Server waits for a `NOTIFY` message with the final response event, even if the transfer destination terminates its dialog (according to RFC 5589). To work around this situation, set the `sip-transfer-complete-message` configuration option to `200` on the Trunk DN representing the referred party.

sip-treatments-continuous

Default Value: `false`
Valid Values:

`true`	A routing strategy treatment is continuously played until the routing destination has answered the call.
`false`	A routing strategy treatment is not played continuously.

Changes Take Effect: Immediately for all new calls

Enables or disables a routing strategy treatment to be continuously played until the routing destination has answered the call.

Note: When `sip-treatments-continuous` is set to `true` and `sip-ring-tone-mode` is set as `0` or `1`, the ringtone is not played. In order for the ringtone to be played, set `sip-treatments-continuous` to `false`.

sip-treatment-dtmf-interruptable

Default Value: `false`
Valid Values: `true, false`

If set to `true`, SIP Server stops playing all prompt elements while processing `RequestPlayAnnouncementAndDigits` treatments, as soon as the first DTMF digit is collected. If set to `false`, SIP Server stops playing the current prompt, but then immediately after digit collection, starts playing the next prompt.

sip-user-agent

Default Value: No default value
Valid Values: A valid string or the special character `*`
Changes Take Effect: On the next call
Related Feature: “Enabling Server and User-Agent Headers” on page 185

Specifies whether SIP Server includes the `User-Agent` header in all request messages that it sends. The value for this option can contain the following placeholders:

- `$VERSION$` = will be replaced with the current SIP Server build
- `$APP-NAME$` = will be replaced with the name of the application in the environment

You can also use the special value *, which is equivalent to `Genesys SIP Server $VERSION$ ($APP-NAME$)`.

sip-vip-script-down

Default Value: `NULL`
Valid Values: Valid name of the Application object
Changes Take Effect: After SIP Server restart
Dependent Option: `control-vip-scripts`
Related Feature: See the *SIP Server 8.1 High-Availability Deployment Guide.*

For the Hot Standby configuration, if the `control-vip-scripts` option is set to `true`. It specifies the name of the Application object representing the scripts that is used to disable the Virtual IP address (or the port for Windows NLB Cluster) when SIP Server is switching to backup mode. The script must be configured as an Application object of type `Third Party Server`.

For example, for a primary SIP Server, you will set the value of this option to `SIP_SERVER_PRIMARY_VIP_DOWN`, and for a backup SIP Server, you will set the value of this option to `SIP_SERVER_BACKUP_VIP_DOWN`.

sip-vip-script-up

Default Value: NULL
Valid Values: Valid name of the Application object
Changes Take Effect: After SIP Server restart
Dependent Option: `control-vip-scripts`
Related Feature: See the *SIP Server 8.1 High-Availability Deployment Guide.*

For the Hot Standby configuration, if the `control-vip-scripts` option is set to `true`. It specifies the name of the Application object representing the script that is used to enable the Virtual IP address (or the port for Windows NLB Cluster) when SIP Server is switching to primary mode. The script must be configured as an Application object of type `Third Party Server`.

For example, for a primary SIP Server, you will set the value of this option to `SIP_SERVER_PRIMARY_VIP_UP`, and for a backup SIP Server, you will set the value of this option to `SIP_SERVER_BACKUP_VIP_UP`.

sip-wait-ack-timeout

Default Value: `2 sec`
Valid Value: Any positive integer, sec/msec (example: 3 sec, 250 msec)
Changes Take Effect: Immediately

When SIP Server processes an incoming re-`INVITE` request, it starts a timer to wait for the `ACK` message to be received for this transaction. Once the `ACK` message arrives, SIP Server sends the re-`INVITE` message to perform the requested operation (greeting, treatment, and so on). If the option is set to `0` (zero), this functionality is disabled.

stranded-calls-overflow

Default Value: No default value
Valid Values: `<destination_number>`, `default` or `<empty string>`, `recall`, `release`, `none`
Changes Take Effect: Immediately
Related Feature: “Alternate Routing for Stranded Calls” on page 106
Related Option: `stranded-call-redirection-limit`

Specifies a list of actions that SIP Server attempts to take for calls stranded on ACD queues. You can configure these actions globally for all queues (at the Application-level) or individually for a particular `ACD Queue` DN. Configure the actions using a comma-separated list of valid values; SIP Server tries to process each item in the list sequentially, moving to the next item if any action fails, and stopping after the first successful action begins (subsequent failure of the successful action does not restart the list).

Note: For a description of the valid values and their related SIP Server actions, see “Stranded Calls Overflow Valid Values” on page 107.

stranded-on-arrival-calls-overflow

Default Value: No default value
Valid Values: `<destination_number>`, `default` or `<empty string>`, `recall`, `release`, `none`
Changes Take Effect: Immediately
Related Feature: “Alternate Routing for Stranded Calls” on page 106
Related Option: `stranded-call-redirection-limit`

Specifies a list of actions that SIP Server attempts to take for calls arriving on ACD Queues with no logged-in agents. You can configure these actions globally for all queues (at the Application-level) or individually for a particular `ACD Queue` DN. Configure the actions using a comma-separated list of valid values; SIP Server tries to process each item in the list sequentially, moving to the next item if any action fails, and stopping after the first successful action begins (subsequent failure of the successful action does not restart the list).

Note: For a description of the valid values and their related SIP Server actions, see “Stranded Calls Overflow Valid Values” on page 107.

stranded-call-redirection-limit

Default Value: `4`
Valid Values: `0`–`15`
Changes Take Effect: Immediately
Related Feature: “Alternate Routing for Stranded Calls” on page 106
Related Options: `stranded-calls-overflow`, `stranded-on-arrival-calls-overflow`

Limits the number of times that SIP Server tries to redirect a stranded call. Use this option to prevent infinite loops during stranded call redirection. SIP Server stops trying to redirect the call after the configured number of attempts. The call remains waiting on the last attempted queue, regardless of its stranded state. As soon as a stranded call is successfully re-routed, the redirection-limit counter is reset.

subscription-delay

Default Value: `0`
Valid Values: `0`–`10000`
Changes Take Effect: Immediately

Specifies the time interval (in milliseconds) between the new individual `SUBSCRIBE` requests used to create new `SUBSCRIBE` dialogs that SIP Server sends if several `Voice over IP Service` DNs are configured with `service-type` set to `blf`.

Note: Genesys recommends setting the option to a value in a range of `20`–`200`.

subscription-event-allowed

Default Value: No default value
Valid Values: String (‘*’, or name of package allowed), `reg`
Changes Take Effect: Immediately

Defines the Event packages allowed by SIP Server for integration with Genesys SIP Feature Server. SIP Server rejects `SUBSCRIBE` messages for unsupported packages. The value asterisk (`*`) allows all subscriptions.

Notes: Genesys recommends the setting `*` for this option. In this case, MWI uses the event `message-summary`.

For more information about integrating with SIP Feature Server, see *SIP Feature Server 8.1 Deployment Guide*.

subscription-max-body-size

Default Value: `14336`
Valid Values: `0–500000`
Changes Take Effect: Immediately

Defines the maximum size of the `NOTIFY` XML body (in bytes) within the `SUBSCRIBE` dialog. If this option is set to `0` (zero), the message body can be any size. The zero value can be used for TCP transport but is not recommended for UDP. For bulk notification, SIP Server sends more than one `NOTIFY`, so adjust the size accordingly.

subscription-timeout

Default Value: `180`
Valid Values: `1–3600`
Changes Take Effect: Immediately

Specifies the time interval (in seconds) in the `Expires` header of the `200 OK` response to a `SUBSCRIBE` request only if the `Expires` header is missing in the `SUBSCRIBE` request.

summary-stat-timeout

Default Value: `60`
Valid Values: Integer value `1-65535`
Changes Take Effect: After SIP Server restart
Related Feature: “HTTP Monitoring Interface” on page 245

Specifies how often, in minutes, the summary statistics are calculated.

switchover-on-msml-oos

Setting: `TServer` section, the SIP Server Application (in standalone mode) or the VOIP Service DN with `service-type=sip-cluster-nodes` (in SIP Cluster mode)
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call
Related Feature: See “Enhanced HA Resilience for Network Disruptions” in the *SIP Server High-Availability Deployment Guide*

Specifies the SIP Server action in case of losing connectivity with MSML VOIP Service DNs. When set to `true`, in the case of strict matching only, VOIP Service DNs with the same or alternative geo-location are considered. After detecting that those DNs are out of service, SIP Server checks one more time that MSML VOIP Service DNs are unresponsive, before reporting the `SERVICE_UNAVAILABLE` status to LCA/SCS in order to trigger a switchover.

switchover-on-trunks-oos

Setting: `TServer` section, the SIP Server Application (in standalone mode) or the VOIP Service DN with `service-type=sip-cluster-nodes` (in SIP Cluster mode)
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call
Related Feature: See "Enhanced HA Resilience for Network Disruptions" in the *SIP Server High-Availability Deployment Guide*

Specifies the SIP Server action in case of losing connectivity with Trunk DNs. When set to `true`, in the case of strict matching only, Trunk DNs with the same or alternative geo-location are considered. After detecting that those DNs are out of service, SIP Server checks one more time that Trunk DNs are unresponsive, before reporting the `SERVICE_UNAVAILABLE` status to LCA/SCS in order to trigger a switchover.

switchover-on-xs-oos

Setting: `TServer` section, the SIP Server Application (standalone SIP Server) or the VOIP Service DN with `service-type=sip-cluster-nodes`
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Enhanced Handling of XS Requests" on page 214

Specifies the SIP Server action in case of losing connectivity with all SIP Feature Server URLs. SIP Server marks a URL as out of service when the threshold of failed heartbeat requests set by the `xs-missed-heartbeat-threshold` option is reached. When set to `true` and all configured SIP Feature Server URLs become out of service, SIP Server reports the `SERVICE_UNAVAILABLE` status to LCA/SCS to switch over to backup mode. When set to `false`, SIP Server responds with a 503 Service Unavailable message to all calls, until one of the SIP Feature Server URLs becomes available.

t-library-stats-enabled

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After SIP Server restart
Related Feature: "HTTP Monitoring Interface" on page 245

When set to `true`, SIP Server collects T-Library client statistics for SIP Server threads and embeds them in HTTP monitoring statistics. When set to `false` (the default), this feature is disabled.

Warning! The `t-library-stats-enabled` option can be used only in the deployment with the persistent and limited number of T-Library clients. Using this option with the number of clients more than 100, will negatively impact SIP Server performance.

time-before-switchover-on-xs-oos

Setting: `TServer` section, the SIP Server Application (standalone SIP Server) or the VOIP Service DN with `service-type=sip-cluster-nodes`
Default Value: `1`
Valid Values: `0-60`
Changes Take Effect: Immediately
Related Feature: "Enhanced Handling of XS Requests" on page 214

Specifies the timeout, in seconds, that SIP Server waits before reporting the `SERVICE_UNAVAILABLE` status in a scenario described in the `switchover-on-xs-oos` option. When set to `0` (zero), SIP Server reports the `SERVICE_UNAVAILABLE` status immediately after the SIP Feature Server VOIP Service DN is detected as out of service.

timed-acw-in-idle

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Emulated Agents" on page 234

Specifies whether SIP Server applies the automatic wrap-up timer (using the `wrap-up-time` parameter) when an agent sends the `TAgentNotReady` request while in idle state.

If set to `false`, SIP Server does not automatically end manual wrap-up—the agent must return manually from ACW.

Note: For compatibility with the previous SIP Server releases, you can use the name `timed-cwk-in-idle` for this option as an alias.

timeguard-reduction

Default Value: `0`
Valid Values: `0–30000 ms`
Changes Take Effect: Immediately
Related Feature: "Outbound IP Solution Integration" on page 306

Calculates the timer duration that SIP Server sends to Media Server to limit the time of the post-connect CPD detection. If the original post-connect CPD timeout value specified by the `cpd-info-timeout` option (in seconds) or `call_timeguard_timeout` key in `AttributeExtensions` is greater than zero (0), then the timeout value sent to Media Server is calculated as follows:

`<original CPD post-connect timeout> - 'timeguard-reduction'`

If the calculated value of the post-connect timeout to be sent to the Media Server is less than 200 ms, then the `timeguard-reduction` option is ignored and the original post-connect CPD timeout value is distributed.

This parameter can be used to improve the reliability of silence detection in the Outbound Solution. A reduced post-connect CPD timeout in the Media Server should ensure the CPD result of silence is received by SIP Server before its

own timer expires. A practical value of the `timeguard-reduction` option can be slightly more than the round-trip time between SIP Server and Media Server.

An increased value of the `timeguard-reduction` option improves the reliability of silence detection, but at the same time it shortens the time taken for the CPD post-connect detection for all scenarios. To avoid this, the value of the original CPD post-connect timeout must also be increased when `timeguard-timeout` is defined. If millisecond precision is required for the definition of the post-connect CPD timeout in the Media Server, then the `call_timeguard_timeout` key in `AttributeExtensions` of `TMakePredictiveCall` must be used to define the original post-connect CPD timeout.

tlib-map-replace-dn

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Enables dynamic replacement of the `[dn]` pattern in SIP headers mapped from T-Library attributes. If you set this option to `true`, SIP Server replaces the [dn] pattern in mapped SIP messages with the digits of the DN where the SIP message is being sent. This applies to both `AttributeExtensions` mapping in `TRouteCall`, and `UserData` mapping as configured on a particular DN.

Note: This `[dn]` pattern replacement functionality applies to SIP header mapping only, not to Request-URI parameters mapping.

tlib-nic-monitoring

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After SIP Server restart
Related Feature: See “Network Status Monitoring” in the *SIP Server 8.1 High-Availability Deployment Guide.*

When set to `true`, this option enables T-Library NIC IP status monitoring. The T-Library IP address is taken from the `Host` object associated with the SIP Server application. The Host object name is used to resolve the T-Library NIC IP address.

trunk-stats-enabled

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately (see Notes below)

When set to `true`, this option enables calculation of trunk statistics and capacity group statistics. When set to `false`, this features is disabled,

Notes:
- Setting this option to `false` does not reset trunk and capacity group statistics; it only stops SIP Server from continuing to calculate them.
- Setting this option to `true` without restarting SIP Server might result in incorrect call statistics and peak call statistics for trunks and capacity groups.

unknown-gateway-reject-code

Default Value: `0`
Valid Values: `0–699`
Changes Take Effect: Immediately

When the `enable-unknown-gateway` option is set to `false`, the `unknown-gateway-reject-code` defines which SIP error code SIP Server returns when an incoming `INVITE` message cannot be associated with an internal device or trunk. If the value of this option is less than `400`, SIP Server uses the `404 Not Found` error code. If the value of this option is `400–699`, SIP Server returns the corresponding error code.

unknown-bsns-calls

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: “Business-Call Handling” on page 235

Determines whether SIP Server considers unknown call types made from or to any agent, as business calls.

untimed-wrap-up-value

Default Value: `1000`
Valid Value: Any nonzero positive integer
Changes Take Effect: Immediately
Related Feature: “Emulated Agents” on page 234

Specifies the threshold (in seconds) at which the timing of ACW stops and manual intervention is required (*untimed ACW*).

update-ctrl-party

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

For call supervision scenarios, when set to `true`, SIP Server sets `AttributeCtrlParty` in `EventCallDeleted` to the party that has released a call.

use-propagated-call-type

Default Value: `never`
Valid Values: `never, monitor`
Changes Take Effect: Immediately

Specifies whether SIP Server uses the call type as defined on the originating site for a multi-site consultation call. If this option is set to `monitor`, SIP Server uses the call type defined at the origination site to identify whether to start monitoring. Genesys recommends using this option in environments where Switch Partitioning functionality is enabled.

userdata-map-all-calls

Default Value: `false`
Valid Values: `true, false`
Changes take effect: Immediately

Specifies whether SIP headers are mapped for all calls. If you set this option to true, SIP Server maps the SIP headers to UserData (and/or Extensions) for all incoming calls, not just for the calls to a Routing Point. This functionality is required to extend the Network Asserted Identity mechanism from SIP messages to T-Library events. For more details, see "Mapping SIP Headers and SDP Messages" on page 261 and "Network Asserted Identity" on page 292.

userdata-map-filter-mode

Default Value: `allow`
Valid Values: `allow, block`
Changes Take Effect: Immediately
Related option: `userdata-map-filter`
Related Feature: "Mapping SIP Headers and SDP Messages" on page 261

Specifies whether the patterns, provided in the `userdata-map-filter` option, are allowed or blocked for mapping the matching T-Library `UserData` to SIP headers.

- If set to `block`, and:
  - `userdata-map-filter=*` - no `UserData` is mapped to SIP headers
  - `userdata-map-filter=<pattern>` - `UserData` matching the pattern is blocked and all others are mapped to SIP headers
- If set to `allow`, and:
  - `userdata-map-filter=*` - all `UserData` is mapped to SIP headers
  - `userdata-map-filter=<pattern>` - only `UserData` matching the pattern is mapped to SIP headers

- If `userdata-map-filter` is not set or empty, no `UserData` is mapped regardless of the userdata-map-filter-mode option setting.

userdata-map-invite-after-refer

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

If this option is set to `true`, SIP Server executes SIP-to-TLib mapping from the SIP `INVITE` message received in response to a `REFER` request that SIP Server sent to an endpoint to transfer the request to a Routing Point. If this option is set to `false` (the default), no mapping is performed from that `INVITE`.

Note: If SIP-to-TLib mapping is configured for both `INVITE` and `REFER` requests and the `userdata-map-invite-after-refer` option is set to `true`, then in cases where an unattended transfer is triggered by a 1pcc `REFER`, SIP Server maps data twice. First, SIP Server maps data from the received `REFER`, and then it maps data from the `INVITE`. If the same keys must be mapped from both `REFER` and `INVITE` (for example, `Call-ID`), the keys from the `INVITE` take precedence.

---

userdata-map-trans-prefix

Default Value: No default value
Valid Values: A string
Changes Take Effect: Immediately
Related Feature: “Mapping SIP Headers and SDP Messages” on page 261

Contains a transport prefix to indicate what headers in the SIP message carry the mapped `UserData`. SIP Server adds this prefix to all data mapped to the outgoing `INVITE` message. SIP Server scans incoming `INVITE` or `REFER` messages used to place a call on the Routing Point for headers that start with this prefix, in addition to performing the normal mapping procedure. Also, SIP Server scans mid-call messages `INFO`, `BYE`, and `UPDATE` for headers that start with this prefix and maps these to `UserData`.

If this option is not specified, no prefix is added to the transmitted data.

verify-sip-names

Default value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Enables forced verification of the dialed number, to ensure it contains only the following syntax:

- characters
- numbers
- symbols: `-_.!~*'()&=+$,;?/`

If set to `true`, SIP Server analyses the user part and if it contains any unsupported syntax, SIP Server considers it invalid. For example, the space symbol is not supported—SIP Server will consider the user part to be invalid.

If set to `false`, SIP Server does not enforce verification.

vip-state-change-timeout

Default Value: `10`
Valid Values: `3-60`
Changes Take Effect: Immediately
Related Feature: See the *SIP Server 8.1 High-Availability Deployment Guide.*

Defines the maximum time allotted (in seconds) for the Virtual IP control script to execute. If the script fails to change the Virtual IP state during this timeout, SIP Server executes the script again. After several unsuccessful attempts, SIP Server declares that the Virtual IP script failed. The same script is not executed after the timeout expires.

wrap-up-time

Default Value: `0`
Valid Value: Any positive integer, `untimed`

`0`	ACW is disabled.
Value greater than 0 but less than `untimed-wrap-up-value`	The number of seconds of timed ACW, after which SIP Server returns the agent to the `Ready` state.
Value equal to `untimed-wrap-up-value`	ACW is untimed and the agent must manually return to the `Ready` state.
Value greater than `untimed-wrap-up-value`	ACW is disabled.
`untimed`	ACW is untimed and the agent must manually return to the `Ready` state.

Changes Take Effect: Immediately
Related Feature: “Emulated Agents” on page 234
Specifies the amount of ACW wrap-up time allocated to emulated agents at the end of a business call.

This option can be set in a number of places, and SIP Server processes it in the following order of precedence, highest first. If the value is not present at the higher level, SIP Server checks the next level, and so on.

SIP Server option priority processing:

1. In the call, in user data `WrapUpTime`, if user data `WrapUpTime` is attached to a call before the call is answered by an agent.
2. In a `DN` configuration object of type `Routing Point`, in the `TServer` section.
3. In a `DN` configuration object of type `ACD Queue`, in the `TServer` section.

4. In the `TAgentLogin` request, in attribute extension `WrapUpTime` (applies to this agent only).
5. In an `Agent Login` configuration object, in the `TServer` section.
6. In a `DN` configuration object of type `Extension`, in the `TServer` section.
7. In a `DN` configuration object of type `ACD Queue` or `Routing Point` that represents logged-in agents (`Agent Group`), in the `TServer` section.
8. In the SIP Server `Application` object, on the `Application Options` tab in the `TServer` section.
9. While in ACW, in the `TAgentNotReady` request with `WorkMode=ACW` (Extending ACW), in attribute extension `WrapUpTime` (applies to this agent only).

xs-heartbeat-interval

Setting: `TServer` section, the SIP Server Application (standalone SIP Server) or the VOIP Service DN with `service-type=extended`
Default Value: `10`
Valid Values: `0-65535`
Changes Take Effect: For the next XS request
Related Feature: “Enhanced Handling of XS Requests” on page 214

Specifies the heartbeat messages interval, in seconds. Value of `0` (zero) disables heartbeats. The setting at a DN level takes priority.

xs-pool-size

Setting: `TServer` section, the SIP Server Application (standalone SIP Server) or the VOIP Service DN with `service-type=extended`
Default Value: `10`
Valid Values: Any number of connections that is possible for the system
Changes Take Effect: For the next XS request
Related Feature: “Enhanced Handling of XS Requests” on page 214

Specifies the maximum number of connections to one SIP Feature Server URL. The setting at a DN level takes priority.

UPDATE, INVITE, INFO, and REFER Sections

The option names in this section are a combination of the `TEvent` attribute name (`Extensions` or `UserData`), a dash, then a numeric value.

extensions-<n>

Default Value: No default value
Valid Values:

- **For SIP-to-T-Lib mapping:** A string containing any character allowed in the header field name of a SIP message (according to RFC 3261) plus the colon character to address the parameter name of a header
- **For T-Lib-to-SIP mapping:** A string containing the SIP header name to be mapped from `AttributeExtensions` of the `TRouteCall` request to the SIP header of the SIP message

Changes Take Effect: Immediately
Related Feature: “Mapping SIP Headers and SDP Messages” on page 261

For SIP-to-T-Lib mapping, the `extensions-<n>` option value indicates which SIP header or SIP header with its parameter is mapped to the `Extensions` attribute. A SIP header name is mapped as a key of the `Extensions` key-value pair, and a SIP header value is mapped as a value of this key-value pair.

You can use the colon character to include the parameter name of a header. For example, `extensions-1=From:tag`. See “Mapping Examples from INVITE Messages” on page 266.

For T-Lib-to-SIP mapping, the `extensions-<n>` option value indicates which key of the `Extensions` attribute key-value pair is mapped as a new SIP header in the `INVITE` message. The value of this key-value pair is mapped as a SIP header value. See the mapping example in “Using the extensions-<n> Option” on page 272.

userdata-<n>

Default Value: No default value
Valid Values: A string containing any character allowed in the header field name of a SIP message
Changes Take Effect: Immediately
Related Feature: “Mapping SIP Headers and SDP Messages” on page 261

The `userdata` option value indicates which SIP header is mapped in the `UserData` attribute. A SIP header name is mapped as a key of the `UserData` key-value pair, and a SIP header value is mapped as a value of this key-value pair. See “Mapping Examples from INVITE Messages” on page 266.

Log Section

There is one SIP Server-specific option available for the `Log` section. For the common `Log` options, see Chapter 10 on page 715.

x-sip-log

Default Value: No default value
Valid Values: The file name
Changes Take Effect: After SIP Server restart
Related Feature: "Multi-Threaded Logging" on page 280
Related Option: `sip-link-type` on page page 526

Specifies whether SIP Server creates a single log file for T-Library messages, or separate log files for other threads when operating in multi-threaded mode. If this option is specified, SIP Server creates separate log files according to the configuration option `sip-link-type`. To configure this option, enter a path and file name where the log files will be created.

If this option does not exist in the configuration, SIP Server generates log files for all running threads. If this option is configured as an empty string, SIP Server generates a single log file for only the main thread.

Note: The SIP processing log file inherits only the following settings from the common log options: `expire, segment, verbose`.

x-sip-mask-sensitive-data

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Masking Sensitive Data in SIP Messages" on page 276

Specifies whether SIP Server masks sensitive data in SIP messages contained in SIP Server logs.

If set to `true`, SIP Server masks all private SIP header values and SIP message body content of all types, except for `application/sdp` and `application/vnd.radisys.msml+xml`. If the message contains `application/vnd.radisys.msml+xml`, SIP Server masks it only when it contains user data.

If set to `false`, SIP Server does not mask sensitive data in SIP messages contained in SIP Server logs.

x-sip-unmask-headers

Default Value: No default value
Valid Values: A list of comma-separated SIP headers
Changes Take Effect: Immediately
Related Feature: "Masking Sensitive Data in SIP Messages" on page 276

Specifies a list of private SIP headers that SIP Server does not mask in SIP messages contained in SIP Server logs. These headers are unmasked in addition to the headers specified in the `x-sip-unmask-headers-default` option. If the value of this option is not configured or empty, headers specified in the `x-sip-unmask-headers-default` are unmasked.

Example: `X-Genesys-UUID, X-ISCC-Id`

x-sip-unmask-headers-default

Default Value: `X-Genesys-strict-location, X-Genesys-peer-proxy-contact, X-Genesys-CallUUID, X-Genesys-PartyInfo, X-Genesys-GVP-Session-ID, X-Genesys-CallInfo, X-Genesys-Route, X-Genesys-geo-location, X-Genesys-bypass-resource-list, X-ISCC-Id, X-ISCC-CofId, X-Detect, Event, presence, Answer-Mode`
Valid Values: A list of comma-separated SIP headers
Changes Take Effect: Immediately
Related Feature: "Masking Sensitive Data in SIP Messages" on page 276

Specifies a list of private SIP headers that SIP Server does not mask in SIP messages contained in SIP Server logs, by default. To unmask other SIP headers that are not included in the default value of this option, use the `x-sip-unmask-headers` option. If the value of this option is empty, the private SIP headers remain masked/unmasked based on the value of `x-sip-unmask-headers` and `x-sip-mask-sensitive-data`.

Multi-Site Support Section

This section must be called `extrouter`.

default-network-call-id-matching

Default Value: No default value
Valid Values: `sip`
Changes Take Effect: Immediately

When this option is set to `sip`, SIP Server will use the content of the `X-ISCC-CofId` header for the ISCC/COF call matching.

To activate this feature, the `cof-feature` must also be set to `true`.

overload Section

This section must be called `overload`.

log-reduce-cpu-threshold

Default Value: `0`
Valid Values: `0`, `5-100`
Changes Take Effect: Immediately
Related Feature: “CPU Usage Overload Control” on page 316

Specifies the CPU usage overload threshold in percent. When the SIP Server CPU usage increases beyond the specified value, SIP Server is considered overloaded and the log level is decremented. The default value of `0` (zero) disables the dynamic overload control feature.

SIP Error Map Section

This section must be called `SipErrorMap`.

sip-<SIP_error_code>

Default Value: No default value
Valid Values: See the Identifying Number column in Table 112
Changes Take Effect: For the next predictive call

Maps a particular 3-digit SIP error code (as defined in various SIP RFCs) with an integer that represents an `AttributeCallState` value included in the `TEvent` response to a `TMakePredictiveCall` request. For the SIP error code variable (`<SIP_error_code>` in the option name), use any integer value from `400` to `699`. For example, `sip-404`, `sip-600`, and so on.

For the value of this option, use the identifying number for the particular `TEvent` message that you want to map. See Table 112 for a list of available `TEvent` messages and their corresponding identifying number.

Table 112: TEvent Identifiers

Identifying Number	TEvent
3	CallStateGeneralError
4	CallStateSystemError
5	CallStateRemoteRelease
6	CallStateBusy
7	CallStateNoAnswer
8	CallStateSitDetected
10	CallStateAllTrunksBusy
11	CallStateSitInvalidnum

Table 112: TEvent Identifiers (Continued)

Identifying Number	TEvent
12	CallStateSitVacant
13	CallStateSitIntercept
14	CallStateSitUnknown
15	CallStateSitNocircuit
16	CallStateSitReorder
26	CallStateDropped
27	CallStateDroppednoanswer
28	CallStateUnknown

Notes:
- Make sure that configured mapping does not inadvertently affect OCS functionality. For example, the `CallStateBusy` message in `EventReleased` is only used in scenarios when the dialed endpoint is found to be busy—not in any other scenario.
- If the media gateway rejects the `INVITE` with an error code, SIP Server checks to see if the Paraxip header `CPD-Result` is present in the response. If so, the value of this header is mapped to the call state. If not, the SIP error code is converted to the `CallState` in accordance with existing mapping.

Agent Login–Level and DN-Level Options

Set configuration options described in this section in the `Options` tab in GAX (formerly, in the `Annex` tab in Genesys Administrator) of the relevant `Agent Login` or `DN` object.

Note: For individual DNs configured behind a softswitch (the `contact` option is not configured), all DN-level option values are taken from the corresponding softswitch DN (`Voice over IP Service` DN with `service-type` set to `softswitch`), and not from the settings on the individual DN. So in effect, all DNs configured behind a softswitch share identical DN-level values.

The one exception is the option `sip-alert-info`, where the value is taken from the individual DN even if it is behind a softswitch.

AuthClient Section

The option names in this section are used to properly calculate authorization parameters used to prepare responses to `401` authorization challenges.

password

Default Value: No default value
Valid Values: Any string
Changed Take Effect: Immediately
Related feature: “SIP Authentication” on page 352

Specifies the password to be included when generating the response to a Digest challenge on this outbound `Trunk` DN, or when generating a Digest Challenge by a softswitch, when an endpoint that is located behind it, receives a request to be authenticated.

username

Default Value: No default value
Valid Values: Any string
Changed Take Effect: Immediately
Related feature: “SIP Authentication” on page 352

Specifies the username to be included when generating the response to a Digest challenge on this outbound `Trunk` DN, or when generating a Digest Challenge by a softswitch, when an endpoint that is located behind it, receives a request to be authenticated.

TServer Section

after-call-divert-destination

Default Value: No default value
Valid Value: Any valid DN
Changes Take Effect: Immediately
Related feature: “Call Divert Destination” on page 116

Specifies the destination DN where SIP Server will divert the call in cases where the caller remains on the line when all other parties have left. For example, use this feature to send callers to an after-call survey.

Note: The `after-call-divert-destination` option is supported only for inbound calls in single-site deployments.

agent-greeting

Default Value: NULL
Valid Values: Any file name played to the agent
Changes Take Effect: On the next call
Related Feature: "Personal Greetings" on page 319

Agent-Login level only. Specifies the media file name that will be used as a greeting for the agent. When used with the customer-greeting option, the option values are used as follows:

- When both options contain different file name values, each file will be played to the customer and the agent as specified.
- When only one option contains a value, the same file will be played to both the customer and the agent.
- When neither option contains a value, no greeting will be played to either the customer or the agent.

Notes:
- When used in conjunction with customer-greeting, if either the customer or agent greeting, for whatever reason, cannot be played, SIP Server does not attempt to play another greeting, but immediately connects the customer and agent. No greetings are played.
- Both agent-greeting and customer-greeting options are configured at the Agent-Login level. Greetings can also be enabled by specifying agent-greeting and customer-greeting keys in AttributeExtensions of the TRouteCall request. These key-value pairs take precedence over the options specified in the Agent Login object.

agent-reject-route-point

Default Value: No default value
Valid Values: Any valid Routing Point
Changes Take Effect: Immediately
Related Feature: "VXML Support for Agent Greetings" on page 321

Specifies the Routing Point where a call is queued if an agent rejects the call. This is used only in multi-site VXML greeting scenarios with the ISCC transaction type route for determining if an agent is willing to accept the call. URS can route the call from this Routing Point to the origination Routing Point at the origination SIP Server.

audio-codecs

Default Value: `telephone-event, PCMU, PCMA, G723, G729, GSM`
Valid Values: Any from the list of `telephone-event, PCMU, PCMA, G723, G729,` and `GSM` words, delimited by commas. Unrecognized words are ignored.
Changes Take Effect: On the next call
Related Option: `sip-enable-sdp-codec-filter`

For a description of this option, see the Application-level option "audio-codecs" on page 444.

Note: The `audio-codecs` option takes effect for SDP renegotiation if the DN-level `sip-enable-sdp-codec-filter` option is set to `true`.

authenticate-requests

Default Value: No default value
Valid Values: `register`, `invite`
Changes Take Effect: Immediately

Determines if incoming SIP requests (`REGISTER` or `INVITE`) are treated with an authentication procedure when the following conditions are true:

- The name of the incoming SIP message exits in the list of the `authenticate-requests` parameter.
- The option `password` is configured on the same `DN` object.

Both `authenticate-requests` and `password` configuration options must be configured on the DN, otherwise no requests will be authenticated.

auto-answer-after

Default Value: No default value
Valid Values: Any valid number
Changes Take Effect: On the next call

Specifies the value that SIP Server adds to the `answer-after` parameter in the `Call-Info` header of the `INVITE` message that it sends to a SIP Endpoint.

auto-logout-ready

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: Immediately
Related Feature: "Automatic Inactive Agent Logout" on page 114

Enables a stricter enforcement of the automatic agent-logout policy (as set in the related `auto-logout-timeout` option). If this option is set to `true`, SIP Server will log out the agent regardless of agent state. If it is set to `false`, SIP Server will not log out agents in the following agent states: `Ready`, `NotReady/ACW`, `NotReady/AuxWork`, `NotReady/LegalGuard`.

You can configure this option in the TServer section of the following objects (listed in order of precedence):

- Agent Login object
- DN object (ACD Position or Extension DN) that represents the device to which the agent is logged in.
- DN object (Routing Point or ACD Queue DN) that represents the queue to which the agent is logged in.
- SIP Server Application object, which specifies the server-wide default.

auto-logout-timeout

Default Value: 0
Valid Values: 0, or any positive integer up to 35791
Changes Take Effect: Immediately
Related Feature: "Automatic Inactive Agent Logout" on page 114

Enables automatic agent logout and specifies the length of time after which the logout occurs (in minutes). To enable this feature, enter a value of 1 or greater; the agent is allowed to remain inactive for this length of time before having to be automatically logged out. To disable this feature, enter a value of 0 (default).

You can configure this option in the TServer section of the following objects (listed in order of precedence):

- Agent Login object
- DN object (ACD Position or Extension DN) that represents the device to which the agent is logged in.
- DN object (Routing Point or ACD Queue DN) that represents the queue to which the agent is logged in.
- SIP Server Application object, which specifies the server-wide default.

auto-redirect-enabled

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

Enables direct routing to the target URI from a SIP 3xx redirect response to an INVITE.

The following events describe the effect of this option:

- SIP Server sends an INVITE request to another SIP device.
- SIP Server receives a 3xx response containing a new target Uniform Resource Identifier (URI).
- SIP Server generates a new INVITE as follows:
  - If auto-redirect=true, SIP Server copies the new target URI to the request URI of the INVITE and sends the INVITE to the location specified in the host portion of the target URI.

- If auto-redirect=false, SIP Server determines the redirect target by processing the username portion of the returned target URI. The routing of the INVITE is determined by SIP Server configuration.

This option must be set in the TServer section on the Trunk DN.

beep-duration

Default Value: 2000 (milliseconds)
Valid Value: Up to a maximum of 10000 (milliseconds)
Changes Take Effect: Immediately

Specifies the length of time, in milliseconds, GVP Media Control Platform (MCP) will play the beep tone if one is requested for the call. Configure this option on the Resource Manager Trunk Group DN. If the beep-duration timer expires before SIP Server receives a notification from GVP that the beep tone is finished, SIP Server proceeds with connecting the agent with the called customer.

blind-transfer-enabled

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

Specifies whether SIP Server processes certain transfer requests while a consultation call is in the dialing state. If set to true, SIP Server processes TCompleteTransfer requests or SIP REFER messages while a consultation call is in the dialing state. Otherwise, such requests are rejected. This option is configured on the transfer target DN.

Notes:
- This option can also be configured at the Application level. The option setting at the DN level takes precedence over the Application level setting.
- This option is for blind transfers only. Blind conference calls are not supported.

capacity

Default Value: 0
Valid Values: Any positive integer
Changes Take Effect: Immediately
Related Feature: “Trunk Capacity Control” on page 372

Specifies how many calls can be handled by a specific Voice over IP device represented in the SIP Server configuration as either a Trunk DN, or a Voice over IP Service DN with service-type set to softswitch.

capacity-group

Default Value: `<DN name>`
Valid Values: Any non-empty string
Changes Take Effect: Immediately
Related Feature: "Trunk Capacity Control" on page 372

Specifies the name of the group of DN objects of type `Trunk` or `Voice over IP Service` (`service-type` set to `softswitch`) with shared capacity. All DNs configured with the same `capacity-group` share the device capacity defined in the `capacity` option.

Note: The value of the `capacity` option must be defined in only one `Trunk` or `Voice over IP Service` DN in any particular `capacity-group`.

capacity-limit-inbound

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Trunk Capacity Control" on page 372

When set to `true`, enables rejection of incoming calls if a limit on the total number of calls for a trunk (or trunks) specified by the `capacity` option is reached. This option must be specified on the same `Trunk` DN where the `capacity` option is defined.

charge-type

Default Value: `0`, Effect varies according to DN type

DN Type	Default Effect
Voice over IP Service	free
Trunk Group	charged
Extension	charged

Valid Values:

`0` Default. Charge type is not set. Charging is based on DN type.

`1` Free. Charge type is free (including an agent).

`2` Charged. Charge type is charged (including IVR).

Changes Take Effect: Immediately (in some cases the change is not considered if the device is currently in a call)
Related Feature: "Early Media for Inbound Calls" on page 232
Related Trunk Options: "sip-early-dialog-mode" on page 615 and "sip-server-inter-trunk" on page 625

Specifies whether a charge will be incurred for services that are supplied by the DN.

Early media audio treatments can be applied to a call by doing one of the following:

- Setting `charge-type` to `0` for `Voice over IP Service` DNs, when Media Server is used.
- Setting `charge-type` to `1` for `Extension` or `Trunk Group` DNs when an external media server or IVR is used (for example, `Trunk Group` DNs that represent access numbers to connect the call to an external IVR/GVP by using its own IVR T-Server).

Notes:
- This option is currently supported on `Trunk Group`, `Extension`, and `Voice over IP Service` DNs only.
- Once the call is established (`200 Ok` is sent), no further early toll-free services are possible.
- In addition to the `sip-early-dialog-mode` and `charge-type` options, Genesys recommends that you also set the `ringing-on-route-point` option on the SIP Server `Application` object to `false`.

clamp-dtmf-enabled

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call
Related Feature: “DTMF Clamping in a Conference” on page 220

When set to `true` on a `Trunk` or `Trunk Group` DN that is added to a conference, enables DTMF clamping for all parties except the DN where this option is configured.

When set to `false`, disables DTMF clamping.

This option applies only to `Trunk` and `Trunk Group` DNs.

connect-nailedup-on-login

Default Value: An empty string
Valid Values: Routing Point number, `gcti::park`
Changes Take Effect: At the next agent login session
Related Feature: “Nailed-Up Connections for Agents” on page 287

Specifies SIP Server actions when receiving a `TAgentLogin` request from a DN with the configured nailed-up connection, as follows:

- When this option is set to a DN of type `Routing Point`, SIP Server immediately establishes a nailed-up connection between an agent’s endpoint and the specified Routing Point. After processing the `TRouteCall` request to the `gcti::park` device, SIP Server parks the agent on `gcti::park`, establishing the persistent SIP connection with the agent’s endpoint.

- When this option is set to `gcti::park`, SIP Server parks the agent on the `gcti::park` device directly, establishing the persistent SIP connection with the agent's endpoint.
- When the value for this option is not specified (the default), SIP Server does not take any action.

At a DN level, this option must be set on an agent `Extension` DN, or, if this DN is located behind the softswitch on the respective softswitch DN.

Note: This option can be configured at both Application and DN levels. Setting at the DN level takes precedence over the Application level.

contact

Default Value: No default value
Valid Values Any alphanumeric string
Changes Take Effect: On the next call, except in Active-Active RM deployments in which the change takes effect only after SIP Server restart.

Contains the contact URI, specifying the device's IP address, if this address is fixed. This option is necessary only for standalone configurations, and only if the configured device does not register itself in the SIP Server registrar. It is part of the persistent registrar feature.

For example, if the SIP device sends a `REGISTER` request to SIP Server and this request is accepted, SIP Server uses the contact information from the `REGISTER` request, and updates (or creates) in the Configuration Layer the option `contact` in the `TServer` section of the corresponding `DN` object.

For a self-registered SIP endpoint, configure the option `contact` with a value of `*` (asterisk).

For a DN behind the softswitch, do not configure the option `contact` (or keep it empty, the default).

The URI format is:

```
[sip:][number@]hostport[;transport={tcp/udp}]
```

OR

```
[sip:][number@]srvFQDN[;transport={tcp/udp}]
```

Where:

- `sip:` is an optional prefix.
- `number` is the DN number. This is currently ignored.
- `hostport` is a `host:port` pair, where `host` is either a dotted IP address or a DNS-resolvable hostname for the endpoint.

Note: If the port number is not included (only the hostname is included), then SIP Server will try to resolve the hostname using DNS SRV records. For more information, see "DNS Name Resolution" on page 217.

- srvFQDN is the SRV FQDN.
- transport=tcp or transport=udp is used to select the network transport. The default value is udp.

Transport Layer Security (TLS). To enable TLS for SIP traffic sent to this device, append the value with the following parameter: transport=tls.

In this case, the URI format is:

```
[sip:][number@]hostport;transport=tls
```

In this case, the same security certificate used for the SIP listening port as configured will be used for SIP traffic to this DN.

Note: For SIP signaling over UDP, the size limit for an individual SIP message is 16 kilobytes. This allows the entire SIP message to fit within a single UDP packet.

contact-list

Default Value: No default value
Valid Values: A comma-separated list of SIP URIs in the format given below
Changes Take Effect: After SIP Server restart

Specifies a list of SIP URIs to support multiple IP address features without DNS. All URIs in the list must use the same transport (UDP/TCP/TLS). Configure each URI using the following format:

```
[sip/sips:][number@]hostport[;transport=(tcp/udp/tls)]
```

where:

- sip/sip is an optional prefix
- number is the DN number (currently ignored)
- hostport is the host:port pair, where the host is a dotted IP address for the endpoint

Note: The contact-list option is only applicable for Active-Active RM integration (and thus it only applies to a Voice over IP Service DN).

contacts-backup

Default Value: No default value
Valid Values: A comma-separated list of any valid SIP URI
Changes Take Effect: On the next call

Specifies a list of SIP URI addresses that supplement the SIP URI specified in the contact option. All URIs in the complete list (contact + contacts-backup) are considered with the same priority and must use the same transport (udp/tcp/tls). You can apply this option to Trunk and Voice over IP Service DNs only. SIP Server uses the Active Out-of-Service Detection feature (oos-check, oos-force, and recovery-timeout options) to determine which node in

the cluster is currently available to handle SIP requests. Configure each URI using the following format:

`[sip:][number@]hostport[;transport=(tcp/udp/tls)]`

Notes:
- The same combination of IP/hostname, port, and protocol must not be used in more than one DN if Active Out-of-Service Detection is enabled. If an incorrect configuration is applied, SIP Server may incorrectly match SIP requests to DNs, or it may inadvertently switch a working DN to Out-of-Service, along with other possible errors.
- The value of the `contacts-backup` option is never used by SIP Server to match an incoming `INVITE` request to a Trunk DN object representing an external party.

For integration with Cisco Unified Communications Manager (UCM), you must configure this option on the `Trunk` DN used to control presence subscription, in cases where more than one Cisco SIP trunk is deployed.

cos

Default Value: No default value
Valid Values: Any COS `Voice over IP Service` DN (`service-type=cos`)
Changes Take Effect: On the next call
Related Feature: “Class of Service” on page 173, and “Dial Plan” on page 195

Specifies the Class Of Service (COS) DN assigned to this DN/Agent Login.

Class of Service (COS) is the functionality that defines telephony capabilities for a device or an agent. This option is used in both of the dial plan-related features supported by SIP Server: Class of Service and Dial Plan. For more information about how to use this option for either functionality, see the following:

- “Class of Service” on page 173
- “Dial Plan” on page 195

cpn

Default Value: No default value
Valid Values: A name that will be used as the user part of the SIP URI
Changes Take Effect: On the next call

Specifies the user part to be included in the SIP URI. SIP Server handles this option differently depending on the following scenarios:

- `cpn` is configured on an `ACD Position` or `Extension` DN.
- `cpn` is configured on a `Trunk` DN.

If configured on an `ACD Position` or `Extension` DN, SIP Server uses the value of this option as the user part of the SIP URI in the `From` header of the `INVITE` message that it sends from this DN to the destination DN. Since this option is

used to provide customized caller-ID information to the destination, this option must be configured in the originating DN.

If configured on a `Trunk` DN, SIP Server uses the value as the user part of the SIP URI in the `From` header of the `INVITE` message it sends to the DN. This special case is used to provide customized caller-ID information for all calls routed through this Trunk (similar to the `replace-prefix` option).

Note: If the option is configured on both the `Extension` and `Trunk` DNs (or `ACD Position` and `Trunk` DNs), when a call is made from `Extension` to `Trunk` (or `ACD Position` to `Trunk`), the value from the option on the `Trunk` takes priority.

---

cpn-digits-to-both-legs

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call
Related Feature: "Modifying the From Header in SIP INVITE" on page 279

This option applies to a `TMakeCall` request containing the `CPNDigits` key-value pair in `AttributeExtensions`. If set to `true`, SIP Server replaces the `User-Name` in the `From` header of the `INVITE` message with the value of the `CPNDigits`, when sending the `INVITE` messages to a call originator and a call destination.

cpn-dnis

Default Value: An empty string
Valid Values: Any string containing the name of the VoIP Service DN with `service-type=dial-plan`
Changes Take Effect: On the next call
Related Feature: "Modifying the From Header in SIP INVITE" on page 279

If configured, SIP Server replaces the `User-Name` in the `From` header of the `INVITE` message with the value produced by applying dial-plan rules to the call DNIS, when sending the `INVITE` to the device/DN where this option is specified. This option applies only to inbound calls.

cpn-self

Default Value: An empty string
Valid Values: Any string
Changes Take Effect: On the next call
Related Feature: "Modifying the From Header in SIP INVITE" on page 279

If configured, SIP Server replaces the `User-Name` in the `From` header of the `INVITE` message with the value of this option, when sending the `INVITE` to the device/DN where this option is specified. This option takes precedence over any other `cpn`-controlling option and the `CPNDigits` key in `AttributeExtensions` of a T-Library request.

cpd-capability

Default Value: No default value
Valid Values: `audiocodes`, `paraxip`, `mediaserver`
Changes Take Effect: Next `TMakePredictiveCall` request

Identifies a particular device as capable of performing call progress detection (CPD). You can configure this option on `Trunk` DNs for media gateways, or on `Trunk Group` DNs for GVP Media Server functionality. For media gateways, you can set the value of this option to either of two supported CPD-capable gateways: `paraxip` or `audiocodes`. For CPD on the media server, you must set the `cpd-capability` option on the Resource Manager `Trunk Group` DN to `mediaserver`.

When making an outbound predictive call, SIP Server narrows the pool of available outbound gateways to those configured for cpd-capability. If no `Trunk` DN with cpd-capability is found, SIP Server will try to perform CPD on GVP instead, using the media server capability of the GVP Media Control Platform (MCP).

customer-greeting

Default Value: `NULL`
Valid Values: Any file name played to the customer
Changes Take Effect: On the next call
Related Feature: “Personal Greetings” on page 319

Agent-Login level only. Specifies the media file name that will be used as a greeting for the customer. The customer greeting plays continuously until the agent greeting finishes playing. The `agent-greeting` and `customer-greeting` option values are used as follows:

- When both options contain different file name values, each file will be played to the customer and the agent as specified.
- When only one option contains a value, the same file will be played to both the customer and the agent.
- When neither option contains a value, no greeting will be played to either the customer or the agent.

Notes:
- When used with `agent-greeting`, if either the customer or agent greeting, for whatever reason, cannot be played, SIP Server does not attempt to play the other greeting but immediately connects the customer and agent. No greetings are played.
- Both `agent-greeting` and `customer-greeting` options are configured at the Agent-Login level. Greetings can also be enabled by specifying `agent-greeting` and `customer-greeting` keys in `AttributeExtensions` of the `TRouteCall` request. These key-value pairs take precedence over the options specified in the Agent Login object.

default-dn

Default Value: NULL
Valid Values: Any valid DN
Changes Take Effect: On the next call

This option can be configured only on DNs of type Routing Point. Specifies the DN to which calls are sent when URS is nonoperational, or when the timeout specified in the router-timeout option expires. This option does not apply to calls that are delivered to an ACD Queue associated with the Routing Point.

Note: This option can also be configured at the Application level. The DN-level setting takes precedence over the Application-level setting.

default-music

Default Value: No default value
Valid Value: The subdirectory and name of the audio file in the MCP root directory, using the following format: <subdirectory>/<music file name>; for example: music/in_queue_welcome.wav
Changes Take Effect: Immediately for all new calls
Related Feature: “Customizing Music on Hold and in Queue” on page 179

This option can be configured on an agent’s Extension DN, an Agent Login, and an ACD Queue DN. Specifies the name of the file that is played for the music-on-hold treatment to a caller when a respective agent places the call on hold or when the call is waiting on an ACD queue. The option applies to calls distributed to this agent, unless a call is passed through a Routing Point with the music-on-hold option, or a call is distributed with the TRouteCall request that contains the music-on-hold key in AttributeExtensions.

Note: This option can also be configured at the Application level, the DN level, and the Agent Login level. The Agent-Login level setting takes precedence over the Application or DN level settings.

dial-plan

Default Value: No default value
Valid Values: Any dial-plan Voice over IP Service DN
Changes Take Effect: For next INVITE or 3pcc operation
Related Feature: “Dial Plan” on page 195

Specifies which dial-plan DN will be applied to calls. You can define the option at any of the following locations listed in order of priority:

- Agent Login level—Applies to calls made by a caller logged in under this Agent Login ID.
- DN level—Applies to calls made from this DN (where Agent Login dial-plan is undefined).
- Application level—Applies to all calls (where no Agent Login or DN dial-plan is defined).

dial-plan-rule-<n>

Default Value: No default
Valid Values: A string in the following format

```
pattern => digits;param1=value1;param2=value2 (etc...) # comment
```

Changes Take Effect: On the next call
Related Feature: "Dial Plan" on page 195

Defines the dial-plan rule. Each rule contains a pattern and an instruction. SIP Server tries to match the dialed number against the patterns defined for all dialing rules on the dial-plan DN.

Each dial-plan rule is made up of the following parts:

`pattern =>`	Specifies the pattern-matching syntax that is matched against the number that was dialed. SIP Server uses the Asterisk format (see Table 53 on page 200).
`digits;`	Specifies the digits SIP Server will use to make the call instead of the digits that were actually dialed. These digits can be any alphanumeric string ending with a semicolon (;). The specific meaning is defined by the value of the type parameter. You can also use the `{DIGITS}` variable—in the format `${DIGITS:X:Y}`—to further define how the dialed digits will be translated into the actual digits used to make the call.
parameters (`param1=value1; ...`)	Specifies a variety of parameters used to control SIP Server actions when processing the dial-plan.
comments (`[space]#`)	A string. Any data after the # is ignored. You must include a space before the #, otherwise SIP Server interprets it as a regular character.

For a detailed description of the syntax required for this option, see the following sections:

- "Pattern Matching" on page 200
- "Digit Translation" on page 200
- "Dial Plan Parameters" on page 201

disable-media-before-greeting

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Disabling Media Before Greeting" on page 324

Specifies whether SIP Server establishes a call in hold state if greetings are configured to be played for a caller and an agent. If set to `true,` SIP Server establishes a call in hold state (an SDP to the caller and the agent is placed on hold/inactive state). If the recording is enabled, the SDP to a recorder is also

placed on hold before the greeting is played. If set to `false`, SIP Server establishes the call in active state and the media is played before the greeting.

Note: This option can be configured at both Application and DN levels. Setting at the DN level takes precedence over the Application level. If this option is set at an Application level and if a particular DN does not support this functionality, this option must be explicitly set to `false` for that DN. For a DN-level activation of this feature, this option must be set for both origination and destination DNs.

disconnect-nailedup-timeout

Default Value: `0`
Valid Values: Any positive integer
Changes Take Effect: At the next nailed-up connection
Related Feature: “Nailed-Up Connections for Agents” on page 287

Specifies whether SIP Server terminates an agent’s nailed-up connection because of the agent’s inactivity. When set to a non-zero value, SIP Server waits this time interval, in seconds, before terminating the agent’s nailed-up connection. When set to `0` (the default), SIP Server does not terminate the agent connection.

Note: This option can be configured at both Application and DN levels. Setting at the DN level takes precedence over the Application level.

display-name

Default Value: No default value
Valid Values: A string
Changes Take Effect: On the next call
Related Option: `use-display-name`

Specifies the string that will be sent as a display name in the `From` header of the `INVITE` request. This option is supported for the following DN types:

- `Extension`
- `ACD Position`
- `Routing Point`
- `Trunk Group`
- `Trunk`

When this option is configured on a `Trunk` DN, it is activated by the `cpn` option. If there is no `cpn` option configured on the `Trunk` DN, SIP Server passes the `INVITE` through the trunk without modifying the display name. If the `Trunk` DN has the `cpn` option configured, SIP Server replaces the display name in the `From` header of the `INVITE` with the value of the `display-name` option, if this option is configured. If the `Trunk` DN has the `override-domain-from` option configured,

SIP Server does not populate the display name in the `From` header, even if the call origination DN has the `display-name` option configured.

SIP Server obtains the display name it populates in the `From` header of the outgoing `INVITE` from the following source, in order of precedence:

1. The extension keys, `CPNDigits` and `DisplayName`, of T-Library requests.
2. The `Trunk` DN of the call destination.
3. DNs of type `Extension`, `ACD Position`, `Routing Point`, and `Trunk Group`.

This option applies only if the `use-display-name` configuration option is set to `true` (the default).

divert-on-ringing

Default Value: `true`

Valid Values:

`true`	SIP Server generates `EventRouteUsed` and `EventDiverted` messages when any SIP `18x` response (`180 Ringing` or `183 Session Progress`) arrives for the `INVITE` request at the routing destination.
`false`	SIP Server postpones `EventRouteUsed` and `EventDiverted` messages until the call is answered by the routing destination with a SIP `200 OK` message. If the call is not answered within the value specified by the `after-routing-timeout` option, the destination SIP dialog is canceled and an `EventError` message is generated.

Changes Take Effect: On the next call

Determines SIP Server behavior when routing calls. You can configure this option on Routing Point DNs only. You can also configure the `divert-on-ringing` key in the `Extensions` Attribute for `TRouteCall` messages.

dr-forward

Default Value: `off`

Valid Values:

`off`	DR forwarding to the peer switch is disabled. SIP Server delivers calls to a DN on the local switch.
`no-agent`	SIP Server forwards the call to its DR peer when there is no agent logged into the DN.
`oos`	SIP Server forwards the call to its DR peer if an endpoint is in an Out-Of-Service (OOS) state. **Note:** Use this value if a SIP phone is configured to work in Business Continuity mode with a single SIP registration.

Changes Take Effect: Immediately

Defines a system-wide mode of forwarding inbound and internal calls when SIP Server is operating in Business Continuity mode. This option can also be set at the Application level. The setting at the DN level takes precedence.

dr-oosp-transfer-enabled

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: On the next call

In Business Continuity deployments, for special circumstances where an inbound call remains on the same site where it arrives and SIP Server puts itself Out of Signaling Path. This option is supported only for `Trunk` DNs pointing to external destinations. It must not be configured on the trunks between SIP Servers.

If set to `false` on the `Trunk` DN from where an inbound `INVITE` is received, SIP Server stays in the signaling path if the call, after being processed on the Routing Point DN, is sent to the local `Extension` DN where the DR call forwarding procedure is applied to deliver the call to the corresponding DN on the peer SIP Server. If set to true (the default), SIP Server puts itself Out Of Signaling Path.

dual-dialog-enabled

Default Value: `true`
Valid Values:

`true`	Set the option to `true` for endpoints that accept more than one SIP dialog and can provide hold/retrieve control over SIP (usually through SIP `NOTIFY`).
`false`	Set the option to `false` for endpoints that can only accept one active SIP dialog and for endpoints that can accept more than one dialog but cannot provide hold/retrieve control over SIP (usually through SIP `NOTIFY`).
`single-dialog-rtp-on-hold`. (introduced in 8.1.103.73)	Use this value for a scenario where a remote agent located behind a PSTN trunk places a call on hold, for which SIP Server connects the on-hold party to a media service with a silent treatment to prevent disconnection of the call by the trunk. SIP Server does not send an inactive SDP to the party during the hold operation. This affects only 3pcc Hold requests and cannot be applied to devices with dual dialog support.

Changes Take Effect: On the next call

Enables the SIP dialog functionality for making consultation calls according to the endpoint type. If the option is set to `false`, SIP Server, while making a 3pcc consultation call, will not allocate a new dialog but uses the existing dialog to connect the DN to the destination (while the other party from the main call is on music). All operations—alternate, reconnect, hold, and

retrieve—can be performed through the re-INVITE operation without the need for NOTIFY processing.

Note: After a regular SIP phone allows the second SIP dialog, it blocks the first dialog from the attempt to re-INVITE. Only pushing a phone button (such as hold/retrieve) may unblock it. The alternate way to unblock the phone is through the SIP NOTIFY message with a special event—hold or talk—that does the same action as the corresponding phone buttons. (See the sip-cti-control option values talk, hold.)

Setting dual-dialog-enabled=false means that either sip-cti-control is not configured or has only partial support such as sip-cti-control=talk (the phone that does allow answering the call through "talk" but does not allow to retrieve it from hold).

emergency-backup

Default Value: No default value
Valid Value: A string
Changes Take Effect: Immediately
Related Feature: "E911 Emergency Gateway" on page 226

Specifies IP addresses, in a comma-separated list, of backup devices used for the Emergency Gateway (EGW) in integrations with the 911 Enable service. The first entry in this list must be the address for the EGW; all other entries should represent any PSTN Trunks that could conduct Emergency Calls.

Note: For the callback Trunk, this option must contain the single address of the backup EGW only.

emergency-callback-plan

Default Value: No default value
Valid Value: A string
Changes Take Effect: Immediately
Related Feature: "E911 Emergency Gateway" on page 226

Specifies the name of the Voice over IP Service dial-plan DN created for integration with the 911 Enable (E911) Emergency Gateway (EGW). For deployments that support Direct Inward Dialing (DID), you must:

1. Configure the dial plan itself so that its dialing rule converts the calling DN (ANI) into a 10-digit call back number (CBN) that SIP Server will include in the P-Asserted-Identity header of INVITE requests it sends on behalf of registered DNs, when processing 911 calls.
2. On the Trunk Group DN representing the EGW, set the emergency-callback-plan option to the name of the ANI-to-CBN dial-plan DN.

Note: For deployments that do not support DID, do not configure this option. Instead, you must configure the `Trunk` DNs to represent the EGW as described in Step 3 of the Table 61 on page 227.

emergency-device

Default Value: `false`
Valid Value: `true, false`
Changes Take Effect: Immediately
Related Feature: "E911 Emergency Gateway" on page 226

If set to `true`, enables this device to conduct emergency calls.

enable-agentlogin-presence

Default Value: `false`
Valid Values:

`true` This value must be used in deployments where agent desktops are not used and all information about agent states is determined by presence subscription. In this environment, SIP Server controls the agent state based on SIP-level information. `EventAgentLogin` and `EventAgentReady` messages are generated if an endpoint registers with SIP Server using the `REGISTER` request, or if the endpoint submits the `PUBLISH` request with the presence content indicating an `open` status. If the endpoint terminates the SIP registration or submits the `PUBLISH` request indicating a `closed` status, then SIP Server generates `EventAgentLogout`. All TEvents are generated on behalf of the agent with the Agent ID set to the same value as the DN name for which all SIP messages are received.

`false` This functionality is disabled.

Changes Take Effect: After endpoint re-registers

Enables an agent login using presence notification. See "Presence from Switches and Endpoints" on page 325 for more information.

Note: You must enable the `subscribe-presence` option before enabling this option.

enable-agentlogin-subscribe

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Enables SIP Server control over the state of an agent based on SIP messages that are received from the agent endpoint. SIP Server can log in or log out an agent in response to SIP `SUBSCRIBE` requests; it can also change the availability

state for an agent in response to NOTIFY requests. To enable this functionality, set this option to true. To disable the functionality, set this option to false.

You must enable this option for any endpoint that supports ACD agent log in and log out, in deployments where Genesys Desktop is not used. When enabled, SIP Server generates EventAgentLogin and EventAgentReady messages in response to SUBSCRIBE requests from the endpoint after agent authentication. If the endpoint terminates the subscription, SIP Server generates an EventAgentLogout message. SIP Server generates EventAgentReady messages in response to NOTIFY requests that have an open presence state, and EventAgentNotReady messages in response to NOTIFY requests that have a closed presence state.

If the agent state changed as a result of TAgentReady or TAgentNotReady message, SIP Server notifies the agent endpoint by sending a NOTIFY request to update the agent status on the IP phone. All TEvents are generated on behalf of the agent with the Agent Login ID taken from the Request-URI in the SUBSCRIBE request. The phone may prompt for both User ID and password. The User ID must correspond to the actual Agent Login object configured in the Configuration Layer. The password is optional and can be left empty (user enters an empty password at the prompt). If a password is required, enter the password in the Enter Password field on the Advanced tab of the Agent Login object. This password must be used during the login.

enable-async-fqdn-resolve

Default Value: false
Valid Values: true, false
Changes Take Effect: After restart
Related Feature: "Asynchronous DNS Resolution" on page 219
Related Options: The enable-async-fqdn-resolve option applies only when the Application-level option TServer/sip-enable-gdns is set to true and common/enable-async-dns is set to 1.

Specifies whether SIP Server resolves an FQDN address contact using the asynchronous DNS resolution method. If set to false, SIP Server resolves the FQDN using the synchronous DNS resolution method. If set to true, SIP Server resolves the FQDN of a DN using the asynchronous DNS resolution method. If the FQDN is unresolvable, SIP Server places the DN out of service. In addition, when any outbound UDP/TCP connection is established, and if the address is the FQDN, SIP Server resolves it using the asynchronous DNS resolution method. If an asynchronous DNS resolution is unresolvable, SIP Server uses synchronous DNS resolution for the call. SIP Server continues applying the asynchronous DNS resolution method to the next call on that DN.

enable-extension-headers

Default Values: `predictive,routing`
Valid Values: See value descriptions below
Changes Take Effect: At the next established call

Controls which SIP headers, specified as a value in `SIP_HEADERS` of `AttributeExtensions` in `TRouteCall` and/or `TMakePredictiveCall` requests, are blocked or mapped from these T-Library requests into an outgoing `INVITE` or `REFER` request, as follows:

- `predictive,routing`—Enables mapping of SIP headers from both `TRouteCall` and `TMakePredictiveCall` requests (the default behavior);
- `none`—Blocks mapping of SIP headers from both `TRouteCall` and `TMakePredictiveCall` requests;
- `predictive`—Enables mapping of SIP headers from TMakePredictiveCall requests only;
- `routing`—Enables mapping of SIP headers from `TRouteCall` requests only.

For example:

If the `enable-extension-headers` option is set to `routing` on a `Trunk` DN and the `TMakePredictiveCall` request contains the following values in `SIP_HEADERS` of `AttributeExtensions`:

```
AttributeExtensions
   'HEADER1'     'data1'
   'HEADER2'     'data2'
   'HEADER3'     'data3'
   'SIP_HEADERS' 'HEADER1,HEADER2,HEADER3'
```

Then SIP Server will block custom SIP headers `HEADER1`, `HEADER2`, `HEADER3` when generating an outgoing `INVITE` or `REFER` request to an external gateway.

In Out Of Signaling Path (OOSP) scenarios where the call goes through several Trunk devices, in order for SIP Server to control (or filter) the mapping of custom SIP headers from `TRouteCall` and/or `TMakePredictiveCall` requests to an outgoing `REFER` request, the `enable-extension-headers` configuration option should be specified on the referred-by Trunk device.

enable-ims

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies how SIP Server will handle `REGISTER` requests and populate SIP headers for this DN.

- `false`—For non-IMS endpoints (local to SIP Server) that are registered to SIP Server. SIP Server will not include IMS-related headers in SIP messages sent to these DNs.

- `true`—For IMS endpoints registered to the IMS-CN (using `REGISTER` requests through the third-party IMS registration). SIP Server communicates with these DNs through the S-CSCF, and adds IMS-specific headers to all SIP messages.

Note: For IMS environments, this option must be set to `true` for all IMS endpoints, as well as for the `Trunk` DN used for routing to IMS. You cannot disable this option on a per-DN basis if already enabled at the Application-level.

enable-direct-pickup

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Call Pickup" on page 120

Specifies whether a direct call pickup feature is enabled for this DN.

enable-oosp-alarm

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Using SIP Feature Server Dial Plan" on page 212

When set to `true`, SIP Server generates alarms 52035 and 52056. When set to `false`, SIP Server does not generate alarms 52035 and 52056.

enable-retransmit-on-oos-transport

Default Value: No default value
Valid Values: `true, false`
Changes Take Effect: On the next call

When this option is set to `true`, SIP Server continues retransmission of the SIP request using the transport associated with the dialog, even though the DN is detected as out of service. If set to `false`, SIP Server does not retransmit the request and sends the timeout to the application layer when the DN is detected as out of service.

This option is applicable only when the UDP transport is used.

Note: This option can be configured at both Application and DN levels. Setting at the DN level takes precedence over the Application level.

enforce-privacy

Default Value: An empty string
Valid Values: `id`
Changes Take Effect: On the next call
Related Feature: "Network Asserted Identity" on page 292

Enforces privacy for an outbound `Trunk` DN or a destination DN of type `Extension`, `ACD Position`, or `Voice over IP Service`. When configured, this option provides the `Privacy` header value that SIP Server includes in `INVITE` requests. In addition, SIP Server replaces the `From` part of the URI with anonymous content. The `P-Asserted-Identity` header will only be included if the `enforce-p-asserted-identity` option is configured on the destination DN and only on a trusted destination.

enforce-p-asserted-identity

Default Value: An empty string
Valid Values: A string
Changes Take Effect: On the next call
Related Feature: "Network Asserted Identity" on page 292

Specifies the preferred SIP URI or the phone number that SIP Server inserts in the `P-Asserted-Identity` header of `INVITE` messages to the trusted destination or intrusted destination when privacy is not requested. This option can be configured on an outbound `Trunk` DN or a destination DN of type `Extension`, `ACD Position`, or `Voice over IP Service`. If this option configured on the destination DN, it takes precedence over the value of the `p-asserted-identity` option configured on the origination DN, or it takes precedence over the value included with the origination `INVITE` request.

enforce-rfc3455

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

If set to `true`, SIP Server does not propagate the `P-Called-Party-ID` header in outgoing `INVITE` messages in SIP environments where RFC 3455 is strictly enforced. If set to `false`, SIP Server propagates the `P-Called-Party-ID` header in outgoing `INVITE` messages.

enforce-trusted

Default Value: Default is applied at the Application-level
Valid Values: `true`, `false`
Changes Take Effect: On the next call
Related Feature: "Network Asserted Identity" on page 292

Specifies whether a particular DN is considered as a "trusted" entity when handling the presentation of private information in the Network Asserted Identity feature. If set to `true` on a particular DN, SIP Server considers the DN trusted, and includes the `P-Asserted-Identity` header for the Contact

associated with this DN. If set to `false`, SIP Server considers the DN non-trusted and removes the `P-Asserted-Identity` header.

This option is also used to enable SIP Server to pass following private headers:

- `P-Early-Media`—See "Early Media Private Header" on page 193.
- `P-Access-Network-Info`—See "P-Access-Network-Info Private Header" on page 319.

Note: This option can also be configured at the Application-level, for backwards compatibility with IMS deployments (where all entities are trusted by default).

force-register

Default Value: `NULL`
Valid Values: Any SIP endpoint address
Changes Take Effect: Immediately

Enables trunk registration and used as the `From` header in the `REGISTER` request.

force-register-disable-totag

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether SIP Server will suppress the `To-Tag` in the second `REGISTER` message sent by SIP Server. If set to `true`, SIP Server does not add the `To-Tag` to the `REGISTER` messages it sends.

fwd-privilege-level

Default Value: No default value
Valid Values: `X, Y, Z...` (each value must be a minimum of `1`, maximum of `10`)
Changes Take Effect: On the next call
Related Features: "Dial Plan" on page 195
Related Option: `privilege-level`

Specifies a list of privileges assigned to a Class of Service DN. This option works similarly to the `privilege-level` option, except it is applied specifically to call forwarding operations. If you do not define this option for call forwarding, then the privilege-level option is used for all call models (including call forwarding) instead.

The `fwd-privilege-level` option applies to the following operations:

- `TSingleStepTransfer`
- `TSingleStepConference`
- `TRedirectCall`
- 1pcc `302 (Moved Temporarily)` received from an endpoint
- 1pcc `SingleStepTransfer`

- `TCompleteTransfer`, `TCompleteConference`. In this case, the matching is applied at the endpoint where the transfer is completed. At least one of the calls must have been initiated by a user with this COS to the privilege number for this to be applicable.

geo-location

Default Value: No default value
Valid Value: Any alphanumeric string
Changes Take Effect: On the next call
Related Feature: "Working with Multiple Devices" on page 386

Specifies the `geo-location` value for the DN that represents a particular SIP device or service. SIP Server includes the `geo-location` attribute in the algorithm that it uses to select a particular service when multiple services are available. After narrowing the pool of currently available resources, SIP Server then matches the `geo-location` value assigned for the call to the `geo-location` option configured on the DN. If more than one match is found, SIP Server can further narrow the selection by considering the value of the `priority` option (if it is configured).

Notes:
- For gateway `Trunk` DN selection, you must set `find-trunk-by-location` to `true` to include the `geo-location` option in the selection procedure.
- Virtual resources such as Routing Points or ACD Queues must not use this option.

Table 113 describes the possible DNs that can use this option:

Table 113: DN Configuration Objects

Device Type	Genesys DN Type
Agent SIP endpoint	`Extension`
Media Gateway	`Trunk`
Media Server	`VoIP Service` with `service-type=msml`
Music-on-hold or Music-in-queue server	`VoIP Service` with `service-type=music`
Voice Treatment Server	`VoIP Service` with `service-type=treatment`
Voice Recorder	`VoIP Service` with `service-type=recorder`
Multipoint Conference Unit	`VoIP Service` with `service-type=mcu`

greeting-call-type-filter

Default Value: No default value
Valid Values: `none, internal, consult, outbound`
Changes Take Effect: Immediately
Related Feature: "Personal Greetings" on page 319

Specifies—using a space-, comma-, or semicolon-separated list—the types of calls to which a greeting will not be played. By default (the option has no value), a greeting will be played to all calls (for 8.0.2 backward compatibility).

If the option is set to `internal, consult,` and/or `outbound`, a greeting will not be played to internal, consultation, and/or outbound calls, respectively.

If the option is set to `none,` the greeting is played to the agent regardless of call type. The keyword `none` cannot be used with other values or delimiters.

Note: If this option is incorrectly configured at the Agent-Login level, SIP Server disregards this option setting, using the Application-level option setting instead (if it is configured).

hg-busy-timeout

Default Value: `0`
Valid Values: `0–600`
Changes Take Effect: On the next call
Related Feature: "Hunt Groups" on page 245

Specifies the period of time, in seconds, SIP Server waits before attempting to deliver a call to a destination that previously rejected the call distributed from this Hunt Group.

hg-members

Default Value: No default value
Valid Values: A string
Changes Take Effect: On the next call
Related Feature: "Hunt Groups" on page 245

Specifies a comma-separated list of DNs that comprise a particular Hunt Group. This list may contain internal DNs (Extensions or ACD Positions).

hg-noanswer-timeout

Default Value: `0`
Valid Values: `0–600`
Changes Take Effect: On the next call
Related Feature: "Hunt Groups" on page 245

For a parallel call distribution, this option specifies a period of time, in seconds, an unanswered call remains in a Hunt Group before SIP Server either redirects the call to the `default-dn` destination (if configured) or rejects it.

For a sequential call distribution, this option specifies a period of time, in seconds, that SIP Server allows for a Hunt Group member to answer a call before SIP Server redirects the call to another available Hunt Group member. If the call is not answered, SIP Server either redirects the call to the `default-dn` destination (if configured) or rejects it.

If set to `0`, the call remains in ringing state until answered by the destination or dropped by the caller.

hg-preferred-site

Default Value: No default value
Valid Values: Any string value
Changes Take Effect: On the next call distribution
Related Feature: See "Hunt Groups in Business Continuity" in the *SIP Server 8.1 High-Availability Deployment Guide.*

Specifies the name of the SIP Server DR Peer application corresponding to the preferred Hunt Group site. If not set or set to an invalid application name, the preferred Hunt Group site cannot be determined, and inbound Hunt Group calls are processed at the site where they are received.

hg-queue-limit

Default Value: `0`
Valid Values: `0`–`20`
Changes Take Effect: On the next call
Related Feature: "Hunt Groups" on page 245

Specifies the maximum number of calls that can be queued at the Hunt Group. When the limit is reached, a new call is either redirected to the `default-dn` destination (if configured) or rejected.

If set to `0`, the number of calls in the queue is unlimited.

hg-queue-timeout

Default Value: `30`
Valid Values: `0`–`6000`
Changes Take Effect: On the next call
Related Feature: "Hunt Groups" on page 245

Specifies the period of time, in seconds, that a call is queued on the Hunt Group waiting for processing. When the time period is reached, the call is either redirected to the `default-dn` destination (if configured) or rejected. If set to `0`, the call remains in the queue until all previous call processing is finished, or the call is dropped by the caller.

hg-type

Default Value: No default value
Valid Values: `fork`, `linear`, `circular`
Changes Take Effect: For next call distribution
Related Feature: “Hunt Groups” on page 245

Specifies the type of Hunt Group algorithm that is used to deliver calls to Hunt Group members, as follows:

- `fork`–Parallel distribution strategy (forking)
- `linear`–Sequential distribution strategy, linear hunting
- `circular`–Sequential distribution strategy, circular hunting

inbound-trunk-hint

Setting: `Trunk` DNs
Default Value: No default value
Valid Values: A string
Changes Take Effect: On the next call
Dependent option: `inbound-trunk-hint-sip-field`

Specifies the value of the SIP header that is defined in the Application-level `inbound-trunk-hint-sip-field` option. SIP Server uses this SIP header value to select the best suitable trunk among other trunks with the same contact option value. The `inbound-trunk-hint` option applies only if the Application-level `inbound-trunk-hint-sip-field` is configured.

If the `inbound-trunk-hint` option is set to an asterisk (*) as a wildcard, SIP Server gives preference to selecting trunks that contain this option as inbound, as compared to trunks that do not have this option configured.

Example:

If `inbound-trunk-hint-sip-field` = `X-CarrierID`, and `Trunk` DNs are configured as follows:

- Trunk_A DN: `contact = address1, inbound-trunk-hint = carrier1`
- Trunk_B DN: `contact = address1, inbound-trunk-hint = carrier2`

When an `INVITE` message arrives containing `Via: address1` and `X-CarrierID: carrier1`, SIP Server selects Trunk_A as inbound for performing particular business needs.

ignore-presence-after-nas

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: On the next call

Specifies whether SIP Server processes or ignores presence SIP messages to change an agent state to `Ready` if the no-answer action is set to `notready` for an agent.

If set to `true`, SIP Server ignores presence SIP messages.

If set to `false`, SIP Server processes presence SIP messages.

Note: You can define this option at both the Application and the DN levels. The DN-level option takes precedence.

include-dial-plan-<n>

Default Value: No default value
Valid Values: Any string that matches the name of another dial-plan `Voice over IP Service` DN
Changes Take Effect: On the next call
Related Feature: "Dial Plan" on page 195

SIP Server will consider the dial-plan rules specified in this option. This option allows hierarchies of dial-plans to be created, if required.

Note: SIP Server selects the dial-plan rule based purely on the number of specific digits matched. No preference is given to any rules in this dial-plan, or in the included dial-plan.

info-pass-through

Default Value: No default value
Valid Values:

`*`	Allows all `INFO` messages to be sent to the peer connection.
`-`	Disables all `INFO` messages from being sent to the peer connection.
`<list>`	Allows only those `INFO` messages specified in a comma-separated list of `Content-Type` values (used to define `INFO` messages) to be sent to the peer connection.
`*,<list>`	Allows all `INFO` messages to be sent to the peer connection, except those included in a comma-separated list of `Content-Type` values.

Changes Take Effect: On the next call

Specifies which SIP `INFO` messages SIP Server will pass to a remote device. You can use this option to allow all `INFO` messages through to the peer connection to disable all `INFO` messages from being sent to the peer connection, or to specify only those `INFO` messages, as defined by the `Content-Type` header, that SIP Server will allow. By default, this option is left undefined. In this case,

SIP Server passes all `INFO` messages to the peer connection, except for the following:

- `application/vnd.radisys.msml+xml`
- `application/x-www-form-urlencoded`
- `application/x-detect`
- `application/dtmf-relay`

Note: This option can also be configured at the Application level. The DN-level setting takes precedence over the Application-level setting.

line-type

Default Value: `0`
Valid Values: `0`, `1`
Changes Take Effect: When an agent reconnects (when set to 1)

Specifies the line type for this DN as either a regular line (`0`) or a nailed-up line (`1`). If set to `1`, when a call to this DN is released due to a 3pcc request (`TReleaseCall`, `TSingleStepTransfer`, or `TCompleteTransfer`), SIP Server does not end the SIP session with this DN. Instead, SIP Server parks the nailed-up line on the `gcti::park device`, where the SIP session is maintained and the DN is able to make 3pcc calls or receive new calls. This behavior is typically required for TDM DNs behind a media gateway, where the agent requires a dedicated connection to the contact center for the duration of a work session.

Note: Nailed-up DNs must not be configured with the `sip-cti-control` option (`talk`, `hold`). This option applies to SIP endpoints only (nailed-up lines are typically TDM lines behind the media gateway).

In addition, for each nailed-up DN you must also configure the following options:

- Set `refer-enabled` to `false`.
- Set `dual-dialog-enabled` to `false`.
- Set `reject-call-notready` to `true` (recommended, not mandatory)

make-call-cpd-merged-userdata

Setting: `TServer` section, the `Trunk Group` DN
Default Value: No default value
Valid Values: A prefix, or a comma-separated list of prefixes that must match the initial characters of the key in the `UserData` key-value pair
Changes Take Effect: For next call

Specifies a prefix (or a list of prefixes) that must match the initial characters of the key in the `UserData` key-value pair. When the initial characters match, SIP Server passes the `UserData` key-value-pair from an engaging call to an outbound call. If this option is not specified, no data is mapped to an outbound call.

For example:

If `make-call-cpd-merged-userdata=test` and `AttributeUserData` contains `'test'='value1', 'testlocal'='value2',` and `'generaltest'='value3'`, only key-value pairs `'test'='value1'` and `'testlocal'='value2'` are mapped. The `'generaltest'='value3'` is ignored, because its initial characters do not match the prefix `test`.

Note: The `make-call-cpd-merged-userdata` option is enabled only on an outbound call made by a `TMakeCall` request with the GDPR feature enabled. To pass the filtered `UserData` in an outbound call to GVP, configure the `userdata-map-filter` option to * (an asterisk) at the `Trunk Group` DN.

make-call-rfc3725-flow

Default Value: `2`
Valid Values: `1, 2`
Changes Take Effect: Immediately

Controls which SIP call flow to choose when a call is initiated by a `TMakeCall` request. The specified value is equal to the call flow number as described in RFC 3725. Only flow 1 and flow 2 from RFC 3725 are currently supported.

Note: This option is enabled only if the option `refer-enabled` is set to `false` for that DN.

music-on-hold

Default Value: An empty string
Valid Values: The subdirectory and name of the audio file in the MCP root directory, using the following format: `<subdirectory>/<music file name>`; for example: `music/in_queue_welcome.wav`
Changes Take Effect: On the next call
Related Feature: “Playing Music to Calls on Hold” on page 179

Specifies the name of the file that is played for the music-on-hold treatment when one of the parties in the call is placed on hold. The option is configured on a `Routing Point` DN and applies to calls that are passed through this Routing Point, unless a call is distributed with the `TRouteCall` request that contains the `music-on-hold` key in `AttributeExtensions`.

no-answer-action

Default Value: `none`
Valid Values:

`none`	SIP Server takes no action on agents when business calls are not answered.
`notready`	SIP Server sets agents to `NotReady` when business calls are not answered.
`logout`	SIP Server automatically logs out agents when business calls are not answered.

Changes Take Effect: Immediately
Related Feature: “No-Answer Supervision” on page 302

Determines the action taken if the agent receives a SIP Server business call but fails to answer the call within the time defined in option `no-answer-timeout`. This option is defined on any `Agent Login` object. When set, the value overrides the Application-level `agent-no-answer-action` SIP Server configuration option for that agent.

Note: If a call is abandoned before either the `no-answer-timeout` or `router-timeout` option expires (depending on which timer is applicable), SIP Server performs no action on this agent.

no-answer-overflow

Default Value: `none`
Valid Values:

`none`	SIP Server does not attempt to overflow a call on an agent desktop when the time specified in the `no-answer-timeout` option expires.
`recall`	SIP Server returns the call to the last distribution device (the device reported in the `ThisQueue` attribute of the call) when the time specified in the `no-answer-timeout` option.
`release`	SIP Server releases the call.
`default`	SIP Server stops execution of the current overflow sequence and continues with the SIP Server default overflow sequence, as defined by the relevant overflow option at the Application level.
Any valid overflow destination	SIP Server returns the call to the specified destination when the time specified in the `no-answer-timeout` option expires.

Changes Take Effect: Immediately
Related Feature: “No-Answer Supervision” on page 302

Defines a sequence of overflow destinations (separated by a comma) in the order listed:

1. When the first overflow destination fails, SIP Server attempts the next one in the list.
2. When all overflow destinations in the list fail, SIP Server abandons overflow. When the list of overflow destinations contains the value `recall` and the call is not distributed, SIP Server skips to the next destination in the list.

This option is defined in the Switches folder on any of the following objects:

- Agent Login (if defined, applies to logged-in agents only)
- DN of type Extension (if defined, applies when agents logged out)
- DN of type ACD Position (if defined, applies when agents logged out)

When set, this option overrides any of the following SIP Server configuration options for the object where it has been set (depending on the configuration object type):

- agent-no-answer-overflow if defined at an Application level
- extn-no-answer-overflow if defined at an Application level
- posn-no-answer-overflow if defined at an Application level

no-answer-timeout

Default Value: Same value as corresponding global option
Valid Value: Any integer from 0–600
Changes Take Effect: Immediately
Related Feature: “No-Answer Supervision” on page 302

Defines the time (in seconds) SIP Server waits for a call that is ringing on a device in question to be answered.

When the timer expires, SIP Server applies the appropriate overflow, and, in the case of agents, the appropriate Logout or Not Ready action.

This option is defined in the Switches folder on any of the following objects:

- Agent Login (if defined, applies to logged-in agents only)
- DN of type Extension (if defined, applies when agents logged out)
- DN of type ACD Position (if defined, applies when agents logged out)

If set to 0, the NoAnswer Supervision feature for this device is disabled. When set, this option overrides any of the following SIP Server configuration options for the object where it has been set (depending on the configuration object type):

- agent-no-answer-timeout if defined at an Application level
- extn-no-answer-timeout if defined at an Application level
- posn-no-answer-timeout if defined at an Application level

no-response-dn

Default Value: No default value
Valid Values: Any valid DN
Changes Take Effect: Immediately
Related Option: “sip-invite-timeout” on page 524

Specifies the DN to which a call will be sent when the SIP endpoint fails to respond to the incoming INVITE message during the creation of a new call. You can configure this option only for DNs of type Extension or ACD Position.

ocs-dn

Default Value: No default value
Valid Values: A valid OCS 2007 DN name
Changes Take Effect: Immediately
Related Feature: "Presence Integration with Microsoft Office Communications Server 2007" on page 331

Specifies the Microsoft OCS 2007 user to be associated with the DN configured with this option. Required for presence monitoring of OCS users on PSTN phones (and no Genesys client on which to log in) to map user status in the IM client to a Genesys agent state. For example, an `online` status in Microsoft Communicator maps to the Genesys agent state `Ready`.

oos-check

Default Value: `0`
Valid Values: `0`–`300`
Changes Take Effect: Immediately
Related Feature: "Active Out-of-Service Detection" on page 240 and "SIP Traffic Monitoring" on page 355

Specifies how often (in seconds) SIP Server checks a device for out-of-service status. This option can be used in conjunction with the `oos-force` and `recovery-timeout` options, as follows:

- When no response is received, and the `oos-force` option is configured, SIP Server will mark a device as out of service when the `oos-force` timeout expires.
- When the `recovery-timeout` option setting is less than the `oos-check` timeout, SIP Server will wait the amount of time specified as the `recovery-timeout` value before checking the DN that was previously detected as out of service.
- When the `oos-check` option is set to `0`, the feature is disabled.

Note: The `oos-check` option is only supported on the following DN types:

- `Voice over IP Service`
- `Trunk`
- `Trunk Group`

The `oos-check` option is not applicable to internal DNs (DNs of type `Extension` or `ACD Position`).

---

oos-error-check

Default Value: `false`
Valid Value: `true, false`
Changes Take Effect: Immediately
Related Feature: "Active Out-of-Service Detection" on page 240

Checks the response message received for the `OPTIONS` message sent by SIP Server. If set to `true`, if SIP Server receives any SIP error response (for example, `480 Temporarily Unavailable` or `503 Service Unavailable`) for the `OPTIONS` message, it places the DN in the out-of-service state. If set to `false`, then, if SIP Server receives an error response for the `OPTIONS` message, it leaves the DN in the in-service state.

Note: This option must be used together with the `oos-check` option.

oos-force

Default Value: `0`
Valid Values: `0`–`30`
Changes Take Effect: Immediately
Related Feature: "Active Out-of-Service Detection" on page 240 and "SIP Traffic Monitoring" on page 355

Specifies when SIP Server places a non-responding device into out-of-service state when the `oos-check` option is enabled, as follows:

- When this option is set to 0 (the default), SIP Server waits 32 seconds before placing the device into out-of-service state.
- When this option is set to a non-zero value, SIP Server waits that number of seconds before device into out-of-service state.

oos-options-max-forwards

Default Value: `0`
Valid Values: Any positive integer from `0` to `70`
Changes Take Effect: Next transaction
Related Feature: "Active Out-of-Service Detection" on page 240

Specifies the value to be used in the `Max-Forwards` header of the `OPTIONS` requests used for Active Out-of-Service Detection. A value greater than `0` allows a proxy device to forward the `OPTIONS` message to the monitored SIP device (for example, an `Extension` on a third-party switch).

oosp-transfer-enabled

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

If set to `true`, SIP Server puts itself in the Out Of Signaling Path (OOSP) after the single-step transfer or routing to the external destination has been completed.

Notes:
- This option is configured for `Trunk` DNs only, and the caller DN or the `Trunk` DN must support the `REFER` method.
- To ensure that the `REFER` method for this `Trunk` DN is used only for OOSP (single-step transfer) scenarios, set the `refer-enabled` option on this Trunk to `false`.
- For a detailed description about how `refer-enabled` and `oosp-transfer-enabled` affect the SIP transfer methods, see "Controlling Transfer Methods to External Destinations" on page 164.

override-call-type

Default Value: `0`
Valid Values:

`0`	`CallTypeUnknown`
`1`	`CallTypeInternal`
`2`	`CallTypeInbound`
`3`	`CallTypeOutbound`

Changes Take Effect: On the next call
Determines the value SIP Server will use as the `CallType` attribute for internal calls made directly to a DN of type `Routing Point`. If set to `0`, SIP Server specifies the `CallType` attribute as `Internal`.

override-domain

Default Value: `NULL`
Valid Values: Any computer name
Changes Take Effect: On the next call

Enables an override of the specified computer name in the SIP `To:` header for a DN. It is used to contact a particular DN in a domain in the `To:` header that is different than the SIP Server internal registrar computer name.

Note: This option must be specified for the DN that represents Microsoft Office Communicator behind LCS.

In IMS environments, for IMS endpoints configured with `enable-ims` set to `true`, the value of the `override-domain` option is used to replace the IP address in the `Request-URI` of all SIP messages. For any IMS-related DN that does not register with SIP Server, this value must match the domain used in that particular IMS deployment.

override-domain-oosp

Default Value: No default value
Valid Values: Valid SIP URL
Changes Take Effect: On the next call

Enables an override of the domain name inside the request URI of the `Refer-To` header in the `REFER` method, or the `Contact` header of `302 Moved Temporarily` responses. If you leave this option empty, no override takes place. To enable the override, configure this option in the `Trunk` DN that is to be selected as the transfer (routing) destination.

This `Trunk` DN should also include the following configurations:

- `contact`—Set to the SIP URL (IP address or FQDN) of the external destination.
- `refer-enabled`—Set to `false` to prevent `REFER` method for transfers to internal destinations.
- `oosp-transfer-enabled`—Set to `true` to use `REFER` method for external transfers, where SIP Server leaves the signaling path. The value of this option takes precedence over the option `override-domain` on the transferred party Trunk. This option does not play any role in how `override-domain-oosp` is configured on the transferred party Trunk.

Note: If the `override-domain-oosp` option is not configured for the DN, then the `override-domain` option value applies in Out Of Signaling Path (OOSP) scenarios.

override-domain-from

Default Value: `NULL`
Valid Values: Any computer name string
Changes Take Effect: On the next call

When set, SIP Server substitutes the computer name in the URI of the `From` headers with the value of this option when it sends the initial `INVITE` message to a DN or `Trunk` DN.

override-domain-refer-to

Default Value: `NULL`
Valid Values: Any computer name string
Changes Take Effect: On the next call
Related Feature: “Referred-By Header Support” on page 172

This option must be configured on a DN associated with the transferred/routed party where `REFER` is sent. If set, SIP Server substitutes the “hostport” component of the SIP URI passed in the `Refer-To` header of the outgoing `REFER` request with the value of this option. Applies only if `sip-referred-by-support` is set to `true`.

override-domain-referred-by

Default Value: NULL
Valid Values: Any computer name string
Changes Take Effect: On the next call
Related Feature: "Referred-By Header Support" on page 172

This option must be configured on a DN associated with the transferred/routed party where REFER is sent to. If set, SIP Server substitutes the "hostport" component of the SIP URI passed in the Referred-By header of the outgoing REFER request with the value of this option. Applies only if sip-referred-by-support is set to true.

override-domain-ruri

Default Value: No default value
Valid Values: A non-empty string
Changes Take Effect: On the next call

Defines what SIP Server inserts in the host part of the Request URI.

override-from-on-conf

Default Value: false
Valid Values:

true	The username is equal to the conference initiator DN.
false	The username is equal to the conf=conf-id or msml=conf-id number.

Changes Take Effect: Immediately
Controls the username part of the From header URI for outgoing INVITE messages to the new party DN added in the single-step conference.
If set to true, SIP Server takes the value of the conference initiator DN as the username part of the From header. If set to false, SIP Sever takes the value of the conf=conf-id or msml=conf-id for the username part of the From header. This option is set on the new party DN added in the single-step conference.

override-to-on-divert

Default Value: false
Valid Values:

true	The username is equal to the destination DN.
false	The username is equal to the Routing Point or ACD Queue number.

Changes Take Effect: Immediately

Controls the username part of the To header URI for outgoing INVITE messages when a call is diverted from a Routing Point or an ACD Queue. This option can be configured on the destination DN only. If set to true, the outgoing INVITE message will contain the username equal to the destination DN in the To header. If set to false, the outgoing INVITE message will contain the username equal to the Routing Point or ACD Queue number in the To header.

This option setting also applies to 1pcc transfers (using the `REFER` method). If set to `true`, SIP Server takes the value of the `REFER`-to DN as the username part of the `To` header. If set to `false`, SIP Sever takes the value of the `From` DN (originator of the `REFER` message) for the username part of the `To` header.

Note: This option can also be configured at the Application level. The DN-level setting takes precedence over the Application-level setting.

p-asserted-identity

Default Value: No default value (empty string)
Valid Values: Any string in accordance with RFC 3325
Changes Take Effect: On the next call
Related Feature: "Network Asserted Identity" on page 292
Related Option: `privacy`

Specifies the preferred SIP URI or telephone number that SIP Server inserts in the `P-asserted-identity` header of `INVITE` messages, when required according to the scenarios described in "Network Asserted Identity" on page 292. If you configure this option on an `Extension` DN, the value provides the content of the `P-asserted-identity` header for this particular DN.

partition-id

Default Value: SIP Server Default Partition
Valid Values: Any string (the name of one partition)
Changes Take Effect: Immediately

Application-level: Specifies the default partition for the SIP Server application.

DN-level: Specifies the partition to which this DN belongs. If you leave the option undefined, SIP Server considers the DN as belonging to the default partition. You can only define this option on `Trunk` and `Voice over IP Service` type DNs.

SIP Server assigns a partition to each call based on the call origination device. If the call origination device does not have a defined `partition-id` parameter, the call is assigned to the default partition. SIP Server uses information about `partition-id` in two ways:

1. To select a `Voice over IP Service` DN for a call.
2. To select a `Trunk` DN for the outbound call.

If multiple resources (`Voice over IP Service` DNs or `Trunk` DNs) are available for call processing, SIP Server selects one that belongs to a call partition.

password

Default Value: No default value
Valid Values: Any alphanumeric string
Changes Take Effect: Immediately
Related Option: `authenticate-requests`

In the endpoint configuration: Specifies the password for the SIP endpoint registration with the local registrar. If it is present, registration attempts are challenged and the password is verified. If it is not present, the registration is not challenged. The realm for password authentication is configured globally; there is one realm per SIP Server. The authentication procedure can also be applied to `INVITE` requests, depending on the value of the `authenticate-requests` option.

Both `password` and `authenticate-requests` must be configured for the authorization process to take place.

In the gateway configuration: Contains the password for gateway registration with the local registrar. This is used for incoming `REGISTER` requests, not for outgoing `INVITE` requests.

peer-proxy-contact

Default Value: No default value
Valid Values: A valid address
Changes Take Effect: Immediately

Specifies the address of the SIP Proxy pool that serves the remote SIP Server. The value must be the same as the value of the `external-contact` option of the `sip-outbound-proxy` DN at the remote switch.

This option must be set on a `Trunk` DN that belongs to the remote SIP Server. Its value is used only when `sip-outbound-proxy` is set to `true`.

The value of the `peer-proxy-contact` option is used to override an FQDN during URI construction in OOSP transfer scenarios, where the transfer destination is the respective remote SIP Server and the transferred party is the external SIP device. If the option value is empty, then the URI is not changed.

This option is mandatory in Business Continuity (BC) deployments with SIP Proxy on BC peers, and on DNs of type `Trunk` which are pointed to by the respective Application-level `dr-peer-trunk` option.

prefix

Default Value: No default value
Valid Values: A string containing any characters allowed in the user part of a SIP URI (according to RFC 3261)
Changes Take Effect: Immediately

This option can be configured on a `Trunk` DN or `Voice over IP Service` DN with the `service-type` option set to `softswitch`, `msml`, or `mcu`.

When configured on a `Trunk` DN, the value of this option is used by SIP Server to select the proper trunk for an outgoing call. For each available trunk, SIP Server compares the value of this option with the initial characters of the call's destination name; the trunk with the longest possible match is selected.

When configured on a `Voice over IP Service` DN with `service-type=softswitch`, the value of this option is used by SIP Server to select the proper softswitch for an outgoing call. For each available softswitch, SIP Server compares the value of this option with the initial characters of the call's destination name; the softswitch with the longest possible match is selected.

When configured on a `Voice over IP Service` DN with `service-type=msml`, the only supported value is `msml=`. It must be configured if the deployment must support conferences.

When configured on a `Voice over IP Service` DN with `service-type=mcu` (when `msml-support=false`), the only supported value is `conf=`.

predictive-timerb-enabled

Default Value: `true`
Valid Values: `true`, `false`
Changes Take Effect: On the next call
Related Feature: "Increasing Ringing Period for Predictive Calls" on page 96

Enables or disables the timer that causes SIP Server to drop the call if an `ACK` message is not sent to the Media Server within 32 seconds after the `200 OK` is received.

If set to `true`, SIP Server uses the 32-second timer.

If set to `false`, SIP Server disables this timer and instead times the call using the `AttributeTimeout` value included in the `TMakePredictiveCall` request. If this timeout expires before the call is answered, or if SIP Server receives a `BYE` message from the Media Server, SIP Server terminates the call. Genesys recommends setting the `AttributeTimeout` to a value greater than zero (0) to prevent inadvertent call termination.

preview-interaction

Default Value: `false`
Valid Values:

`none`	Disables the protocol.
`tlib`	Enables preview interaction through T-Library messaging.
`chat`	Enables preview interaction through SIP Instant Messaging (IM).

Changes Take Effect: On the next call
Related Feature: "Preview Interactions" on page 335

Determines if the `Preview Interaction` protocol is enabled when incoming calls are diverted from a Routing Point.

To enable the preview mechanism for IM interactions, set this option to `chat`. SIP Server sends a preview IM allowing the IM user to accept or reject the request before SIP Server starts the main IM dialog. If the IM user rejects the request, SIP Server returns the call to the routing point. For details about the call flow, see "Preview Interaction" on page 251.

Notes:
- This option works with the `preview-expired` and `forced-notready` options to determine what action to take when a desktop does not respond to a preview interaction before the time expires.
- For backward compatibility, SIP Server also accepts the following valid values for this option:
  - `true`—The protocol is enabled (equivalent to T-Library).
  - `false`—The protocol is disabled (equivalent to `none`).

priority

Default Value: `0`
Valid Values: Any non-negative integer
Changes Take Effect: Immediately

Specifies the device priority for the device selection algorithm. A smaller value designates a higher priority. SIP Server will choose a device in round-robin fashion across all devices if more than one device with the same priority is configured. This option is used to control the device switchover during a failure and to provide lowest-cost routing.

privacy

Default Value: No default value (string is empty)
Valid Values: Valid Values: `id`—as defined in RFC 3325: Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks
Changes Take Effect: On the next call
Related Feature: "Network Asserted Identity" on page 292

Specifies the level of privacy requested by this DN, as described by the Network Asserted Identity feature. If set to `id`, SIP Server includes the `Privacy:id` header in SIP messaging. If not configured, SIP Server does not include the `Privacy:id` header (privacy is not requested by this DN).

private-line

Default Value: No default value
Valid Value: Any valid SIP user name
Changes Take Effect: Immediately
Related Feature: "Presence from Switches and Endpoints" on page 325

Specifies the value of the Private Line User Part ID for the Busy Lamp Field feature. When this option is configured on a particular DN, SIP Server creates an association between the value of this option and the name of the DN. When the softswitch sends a notification regarding the state of the Private Line, SIP Server maps the User Part ID to the value of the `private-line` option on the associated DN, and then issues an `EventAgentReady`/`EventAgentNotReady` message for that DN.

privilege-level

Default Value: No default value
Valid Values: `X, Y, Z...` (each value must be a minimum of `1`, maximum of `10`)
Changes Take Effect: On the next call
Related Option: `fwd-privilege-level` (page 582)
Related Features: "Dial Plan" on page 195

Specifies a list of integers that define which dial-plan rules are available for outgoing calls made by the caller associated with this Class of Service (COS) DN.

When a caller with this COS makes a call that matches a dial-plan rule, the privilege of that dial-plan rule must be included here, otherwise SIP Server will block the call. If the dial-plan rule does not define a privilege level (or defines a privilege-level of `0`), then the call is allowed regardless of the privilege levels defined here.

Applicable to the following 3pcc requests:

- `TMakeCall`
- `TInitiateTransfer`
- `TInitiateConference`
- `TSingleStepTransfer*`
- `TSingleStepConference*`
- `TRedirectCall*`

SIP Server will also apply dial-plan logic to 1pcc `INVITE`, `REFER*`, and `302 (moved temporarily)*` operations.

* The `fwd-privilege-level` option, if configured, overrides the `privilege-level` setting for these operations.

public-contact

Default Value: No default value
Valid Value: Any alphanumeric string
Changes Take Effect: Immediately

Contains the public `host:port` pair for a softswitch. This is the public IP address of the softswitch. SIP Server uses this address to fill the destination (`Refer-To`) address in `REFER` requests. On some switches, this is the same as the `contact` address; if this is the case, you do not need to specify this parameter.

SIP Server uses this address to fill the `host` parameter in the `To` header of the `INVITE` request if the option is set on the Voice over IP Service object containing `service-type=softswitch`.

record

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: When the next call is established on the DN

If set to `true,` call recording begins automatically when the call is established on the DN. Call recording stops when the call is released on the DN.

recovery-timeout

Default Value: `0`
Valid Values: `0–86400 seconds`
Changes Take Effect: Immediately
Related Feature: “Passive Out-of-Service Detection” on page 239
Related Options: `sip-oos-enabled` on page 620

Controls whether a device is taken out of service when an error is encountered, and if so for how long. If set to `0,` setting the DN to out-of-service due to SIP failure is disabled for DNs of type `Trunk` or `Voice over IP Service`. For DNs of type `Extension` (ACD Position), the zero value does not have any affect.

SIP Server supports this option on the following DN types:

- `Voice over IP Service`
- `Trunk`
- `Extension`
- `ACD Position`
- `Voice Treatment Port`

Note: Genesys recommends that you disable the `recovery-timeout` option when using Active Out-of-Service Detection. Recovery-timeout is intended for Passive Out-of-Service Detection only.

However, if `recovery-timeout` and Active Out-of-Service Detection are enabled at the same time, when the device is detected as out of service, and the `recovery-timeout` option is configured to a value less than the `oos-check` value, SIP Server will wait the amount of time that is specified in the `recovery-timeout` option before it checks to see if the device is back in service.

For more information about active out-of-service-detection, see "Active Out-of-Service Detection" on page 240.

rfc-2976-dtmf

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

When this option is set to `true` in a particular DN (type of `Trunk` or `Extension`) configuration, SIP Server will send DTMF tones in the RFC 2976 format to that device using the `INFO` request method when an agent issues a `TSendDTMF` request.

If a `TSendDTMF` request contains a string with multiple digits (for example, `12345#`), SIP Server issues multiple `INFO` requests (one per digit).

If a `TSendDTMF` request contains a string with multiple digits, and there are unsupported DTMF tones in this string (for example, `123a67`), SIP Server still attempts to send the `INFO` request for each digit contained in the string, ignoring possible error responses from a gateway, and continuing to send subsequent digits.

refer-enabled

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: With the next new call on this DN

Specifies whether the `REFER` method is sent to an endpoint. If set to `true`, the `REFER` method is sent to:

- The call party that originates a `TMakeCall` request.
- The call party that initiates a consultation call.
- The call party that is transferred to another destination during a single-step transfer.

If set to `false`, SIP Server uses the re-`INVITE` method instead.

For IMS deployments: When integrated with an IMS environment, you must set this option to `false` on all IMS-enabled DNs.

Notes: To control the SIP messaging (`REFER` or re-`INVITE`) that SIP Server uses to initiate transfers or routing to an external DN, configure the outbound `Trunk` DN according to the following rules:

- For two-step transfers, the `refer-enabled` setting on the `Trunk` DN takes precedence over `oosp-transfer-enabled`.
- For single step transfers, the `oosp-transfer-enabled` setting on the `Trunk` DN takes precedence over `refer-enabled`.

For a table describing how these two options control the SIP methods used, see "Controlling Transfer Methods to External Destinations" on page 164.

---

reinvite-requires-hold

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

If set to `true`, this option instructs SIP Server to always precede a re-`INVITE` request to an endpoint with a special re-`INVITE` containing hold SDP and `0` audio port.

Note: This option must be enabled (set to `true`) only for Microsoft RTC-based devices. In this case, the option prevents an audio delay during 3pcc (third-party call control) conferencing with the RTC-based endpoint.

In all other cases, consult Genesys Customer Care for recommendations about enabling this option.

---

reject-call-incall

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

If set to `true`, SIP Server rejects a call attempt to a DN that is already on a call, and generates an `EventError` message with the reason code `Destination Invalid State (93)`. When rejecting 1pcc calls, SIP Server generates a SIP `603 Decline` error response.

reject-call-notready

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

If set to `true`, SIP Server rejects a call attempt to a DN at which an agent is in a `Logout`, `NotReady`, or `AfterCallWork` state; and generates an `EventError` message with the reason code `Destination Invalid State (93)`. When rejecting 1pcc calls, SIP Server generates a SIP `603 Decline` error response.

Note: The `reject-call-incall` and `reject-call-notready` options must be configured on destination DNs and not on origination DNs. These options are applicable to the following T-Library requests:

- `TMakeCall`
- `TInitiateTransfer`
- `TInitiateConference`
- `TSingleStepTransfer`
- `TSingleStepConference`

replace-prefix

Default Value: No default value
Valid Values: A string containing any characters allowed in the user part of a SIP URI (according to RFC 3261), or empty
Changes Take Effect**:** Immediately

This option can be configured on a Trunk DN or Voice over IP Service DN with the `service-type` option set to `softswitch`, under a condition that the option `prefix` is also configured on these DNs.

When a device with both `prefix` and `replace-prefix` options is selected to conduct the call, initial characters of the call's destination name that match the prefix will be substituted with the value of `replace-prefix`. If this option contains an empty value, the call's destination name will be stripped of initial characters matching the value of the prefix.

replace-uri-contact

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "DNS Name Resolution" on page 217

Specifies whether SIP Server replaces the IP address and port in the SIP URI with the address and port of the active destination, as determined by resolving the contact or contacts-backup option to DNS records, using Active Out-of-Service Detection that factors in DNS priority and weight values.

request-uri

Default Value: No default value
Valid Values: Any SIP URI
Changes Take Effect: Immediately

Specifies the value of the `Request-URI` address inside the `INVITE` message that is different from the address to which the message will be sent—for example, if a service that is provided by a particular application/server requires a different URI from the Contact URI for that application/server.

In a video support configuration, this option creates a template for specifying the source of the video stream as the value of the `Request-URI` parameter in the `INVITE` message:

`annc@<stream_manager_hostport>;play=<file>`

Note: For `moh`, `treatment`, or `mcu` services provided by Genesys Media Server, do not configure the `request-uri` option. Configuring this option can improperly override media-related options in the SIP Server Application. Specifically, if the play parameter in the `Request-URI` from a `Voice over IP Service` DN with a `service-type` of `moh` contains a value, this value overrides the `music-in-conference-file` option as specified in the SIP Server Application.

resolve-external-contact

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

Specifies whether SIP Server resolves the contact as external if the internal resolution has failed. The value will be taken from the trunk which was the source of the OOSP-causing message, which is the trunk to the transfer initiator party.

This option affects only processing of the OOSP (Out Of Signaling Path) transfer SIP operations, specifically `REFER` requests or `302` responses. It applies only to DNs of type `Trunk`.

SIP Server tries to find the destination device using the URI in the OOSP message, as follows:

- Resolving the user part—SIP Server searches among locally configured and registered DNs and tries to match trunk prefixes.
- If no matching DNs are found and if the `resolve-external-contact` option is set to `true`, SIP Server tries to find the destination trunk by matching the domain part and transport protocol of the received URI with the contact of the configured `Trunk` DNs.

Note: This option can also be configured at the Application level. The DN-level setting takes precedence over the Application-level setting.

reuse-sdp-on-reinvite

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

When a call is routed to an endpoint, and this option is set to `true` in the destination endpoint configuration, SIP Server generates an offer by sending a re-`INVITE` message to the origination party (or to the MCU). When the origination party answers the offer, SIP Server sends the `INVITE` message with SDP information to the destination.

This option was introduced to handle obsolete devices that do not work properly with empty `INVITE` requests.

Notes:
- The value must be set to `true` when using EyeBeam version 1.1.
- If one of the DNs has this option set to `true` and the other DN has this option set to `false`, or not configured, SIP Server will not start renegotiating with the DN that has the option value set to `true`. Therefore, the empty `INVITE` will never be sent to that DN. If both DNs have this option set to `true`, one of the DNs will receive an empty `INVITE`.
  SIP Server sends a re-`INVITE` for SDP re-negotiation first to the device configured with `reuse-sdp-on-reinvite=false` (or if the option is not configured).

ring-tone-on-make-call

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

Affects the `TMakeCall` request when using the re-`INVITE` procedure.

If set to `true`, SIP Server connects the caller with an audio ringtone from Media Server when the destination endpoint responds with a `180 Ringing` message. In addition, the following options must also be configured for these scenarios:

- The calling DN initiates a `TMakeCall` request must be configured with the following options:
  - `refer-enabled` set to `false`
  - `make-call-rfc3725-flow` set to `1`
- The calling DN initiates a consultation call must be configured with the following option:
  - `dual-dialog-enabled` set to `false`

When the `ring-tone-on-make-call` option is set to `false`, there is no ringtone.

Notes:
- SIP Server does not support internal ringtones in conference scenarios where the `sip-ring-tone-mode` option is set to `1`. In this case, SIP Server provides a ringtone only if the endpoint returns an SDP in the provisional message.
- SIP Server plays internal ringtones based on the `ring-tone-on-make-call` configuration option when the `sip-ring-tone-mode` option is set to `0`.

sca-preferred-site

Default Value: No default value
Valid Values: Any string value
Changes Take Effect: On the next call
Related Feature: See "Shared Call Appearance in Business Continuity" in the *SIP Server 8.1 High-Availability Deployment Guide.*

Specifies the name of the SIP Server DR Peer application corresponding to the preferred SCA site. If not set or set to an invalid application name, the preferred SCA site cannot be determined, and inbound SCA calls are processed at the site where they are received. The option can be configured only for the Primary shared line DN, where the `shared-line` option is set to `true`.

service-type

Default Value: No default value
Valid Values: Any string
Changes Take Effect: Immediately

Specifies the configured SIP device type or service (see Table 114). See Chapter 4, "SIP Devices Support," on page 77 for more information on using this option. For more information about configuring a Class of Service DN, see "Class of Service" on page 173. For more information about configuring a dial-plan DN, see "Dial Plan" on page 195.

Table 114: Service-Type Settings for SIP Devices

SIP Device Type or Service	Genesys DN Type	service-type Setting
Conference Server/MCU	Voice over IP Service	`mcu`
Softswitch	Voice over IP Service	`softswitch`
Music-on-Hold servers	Voice over IP Service	`music` or `moh`
Treatment service	Voice over IP Service	`treatment`
Recording service	Voice over IP Service	`recorder`

Table 114: Service-Type Settings for SIP Devices (Continued)

SIP Device Type or Service	Genesys DN Type	service-type Setting
Application service	Voice over IP Service	`application`
Class of service	Voice over IP Service	`cos`
Dial plan	Voice over IP Service	`dial-plan`

session-refresh-enforced

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

Controls whether SIP Server activates the SIP Session timer within a SIP dialog. If set to `false`, SIP Server activates the SIP Session timer only if both an initial `INVITE` and `200 OK` response to that `INVITE` contains the `Session-Expires` header. If set to `true`, SIP Server, while activating that timer, ignores the absence of the `Session-Expires` header in the response and starts the timer based on the header presence in the request. If an endpoint does not support the session refresh mechanism, set this option to `false`. The option has an affect only when the `session-refresh-interval` option is set to a non-zero value.

You can define this option at both the Application and the DN level. The DN-level setting takes precedence over the Application-level setting.

shared-line

Default Value: false
Valid Values: true, false
Changes Take Effect: On the next call
Related Feature: “Shared Call Appearance” on page 357

Indicates if this DN is used as a Primary shared line number.

shared-line-capacity

Default Value: `2147483646`
Valid Values: Integer in the range `1-2147483646`
Changes Take Effect: On the next call
Related Feature: “Shared Call Appearance” on page 357

Specifies the maximum number of line appearances (or simultaneous calls) for a Primary shared line DN. These calls are distributed among shared line users and one user can handle only one call at a time. This option can be configured only for a Primary shared line DN (`shared-line=true`). The default value means that the number of simultaneous calls is (almost) unlimited.

shared-line-number

Default Value: No default value
Valid Values: Primary shared line DN
Changes Take Effect: On the next call
Related Feature: "Shared Call Appearance" on page 357

Specifies the Primary shared line DN to be used by the Secondary shared line DN to receive incoming calls and make outgoing calls.

sip-accept-body

Default Value: An empty string
Valid Values: `cid` or empty
Changes Take Effect: On the next call
Related Feature: "Caller Information Delivery Content for AT&T Trunks" on page 117

Specifies content types that SIP Server retrieves from the incoming `INVITE` with a multipart body received from an origination DN.

- If set to an empty string (default), SIP Server ignores the multipart body of an `INVITE`.
- If set to `cid`, SIP Server extracts the CID body from the `INVITE` and stores it as the caller's property.

This option:

- ...does not affect SDP.
- ...is supported only on `Trunk` DNs, ignored by all other DN types.

sip-add-via

Default Value: No default value
Valid Values: `peer-address`
Changes Take Effect: On the next call

If this option is set to `peer-address`, SIP Server adds the additional bottom-most `Via` header with the IP address of the peer SIP endpoint.

sip-alert-info

Default Value: No default value
Valid Values: Any string
Changes Take Effect: Immediately
Related Feature: "Alternate Ringtones" on page 102

Specifies the content to be added to the `Alert-Info` header, used to trigger alternate ringtones or auto-answer functionality in the destination endpoint.

If configured, SIP Server includes the `Alert-Info` header with the value of this option whenever it sends an `INVITE` to this `Extension` or `ACD Position` DN—unless a different value is configured in the `SIP_HEADERS` extension of the initiating T-Library request.

For example, the following value points the endpoint to a ringtone file that can be used for internal calls:

`<http://www.provider.com/tones/internal_caller.pcm>`

Note: The URI must be enclosed in angle brackets.

If alternate ringtones are also configured for external or consultation calls (`sip-alert-info-external` or `sip-alert-info-consult`), that configuration takes precedence over `sip-alert-info`.

Note: The value of the option `sip-alert-info` applies to the individual DN in all cases, regardless of whether this DN is configured behind a softswitch.

sip-alert-info-external

Default Value: No default value
Valid Values: Any string
Changes Take Effect: Immediately

Related Feature: "Alternate Ringtones" on page 102

Specifies the content to be added to the `Alert-Info` header for inbound external calls. If configured, SIP Server will include the value of this option in the `Alert-Info` header of the `INVITE` messages that it sends for an external call to this `Extension` or `ACD Position` DN—unless a different value is configured in the `SIP_HEADERS` extension of the initiating T-Library request.

For example, the following value points the endpoint to the ringtone file that will be used for external calls:

`<http://www.provider.com/tones/internal_caller.pcm>`

Notes:
- The URI must be enclosed in angle brackets.
- This option is not supported on a `Voice over IP Service` DN with `service-type` set to `softswitch`.

sip-alert-info-consult

Default Value: No default value
Valid Values: Any string
Changes Take Effect: Immediately
Related Feature: "Alternate Ringtones" on page 102

Specifies the content to be added to the `Alert-Info` header for consultation calls. If configured, SIP Server will include the value of this option in the `Alert-Info` header of the `INVITE` messages that it sends to establish a consultation call with this `Extension` or `ACD Position` DN—unless a different value is configured in the `SIP_HEADERS` extension of the initiating T-Library request.

For example, the following value points the endpoint to the ringtone file that will be used for external calls:

`<http://www.provider.com/tones/consultation_call.pcm>`

Notes:
- The URI must be enclosed in angle brackets.
- This option is not supported on a `Voice over IP Service` DN with `service-type` set to `softswitch`.

sip-answer-mode

Default Value: An empty string
Valid Values: `Auto`
Changes Take Effect: Immediately

Specifies the content to be added to the `Answer-Mode` header used to trigger the auto-answer functionality in the destination endpoint. SIP Server sends this header regardless of whether an endpoint has advertised support for the "answermode" sip.extension in the contact of a `REGISTER` message. If this option is configured, SIP Server includes the `Answer-Mode` header with the value of this option whenever it sends an initial `INVITE` message.

Notes:
- The `sip-answer-mode` option can be set at both DN and Application levels. Setting at a DN level takes precedence over Application-level setting.
- Support of the `Answer-Mode` SIP header in `Auto` mode as described in RFC 5373 is compatible with Avaya 96xx phones. Avaya phones send `INVITE` messages without a `Referred-By` header in response to `REFER` from SIP Server; therefore, the `refer-enabled` configuration option must be set to `false`. Also, for Avaya phones, the `dual-dialog-enabled` configuration option must be set to `true` and the `sip-cti-control` configuration option should not be configured.

sip-busy-type

Default Value: `0`
Valid Values: `0, 1, 2`
Changes Take Effect: Immediately

When this option is set to `0` (the default), a busy tone is always played. When this option is set to `1`, a busy tone is played for a calling party only if a treatment is previously applied to a call or a call is originated by a 3pcc make call operation, and the `refer-enabled` option is set to `false`. Otherwise, the rejected response is sent back to the calling party. When this option is set to `2`, a busy tone is not applied, and if SIP Server does not accept an `INVITE` session from a calling party, the rejected response is sent back to the calling party.

sip-chat-format

Default Value: `text`
Valid Values: `text`, `html`
Changes Take Effect: Immediately
Related Feature: "Instant Messaging" on page 250

Specifies the format of the `UserData` IM content when different SIP endpoints support different IM formats.

If you set this option to `text`, the `UserData` content in the IM is encoded in text (text/plain) format. You must use this value for Microsoft Office Communicator endpoints.

If you set this option to `html`, the `UserData` content in the IM is encoded in HTML (text/HTML) format. You must use this value for Eyebeam SIP endpoints.

sip-contact-user

Default Value: No default value
Valid Values: `as-from`
Changes Take Effect: On the next call

If this option is set to `as-from`, SIP Server inserts into the `Contact` header the same user name found in the `From` header of the `INVITE` message.

sip-cti-control

Default Value: No default value
Valid Values: `beep`, `dtmf`, `talk`, `hold` (See Table below)

Table 115: sip-cti-control Values

Endpoint	Value	Description
Genesys SIP Endpoint	beep	Enables SIP Server to remotely control the playing of beep tones during call recording on a SIP endpoint built on the Genesys SIP Endpoint SDK 8.0.
	dtmf	Enables SIP Server to remotely control DTMF generation on SIP endpoint built on the Genesys SIP Endpoint SDK 8.0.
	talk	The `TAnswerCall` request is issued against the DN, which means that the call is answered remotely by a T-Library client. The SIP method `NOTIFY (event talk)` is used. Otherwise, the `TAnswerCall` request is not supported.
	hold	The `THoldCall` request is processed by a `NOTIFY (event hold)` message. The `TRetrieveCall` request is processed by a `NOTIFY (event talk)` message.

Table 115: sip-cti-control Values (Continued)

Endpoint	Value	Description
BroadSoft SIP Endpoint	talk	The `TAnswerCall` request is issued against the DN, which means that the call is answered remotely by a T-Library client. The SIP method `NOTIFY (event talk)` is used. Otherwise, the `TAnswerCall` request is not supported.
	hold	The `THoldCall` request is processed by a `NOTIFY (event hold)` message. The `TRetrieveCall` request is processed by a `NOTIFY (event talk)` message.

Changes Take Effect: On the next call
Related Feature (for values `beep` and `dtmf`): "Remote Media on Genesys SIP Endpoint SDK 8.x" on page 347

Specifies the behavior of a DN that represents either of the following types of SIP endpoints:

- SIP endpoint built on the Genesys SIP Endpoint SDK 8.0, using proprietary SIP extensions. For this endpoint, you can configure this option with the values both `beep` and `dtmf`. For more information, see the *SIP Endpoint SDK 8.x API Reference*.
- SIP endpoint which supports the BroadSoft SIP Extension Event Package. For this endpoint, you can configured this option with the values `talk` and `hold`.

Note: For either SIP endpoint, the two supported values for that endpoint can be used simultaneously as a list of comma-separated values.

sip-disable-greeting

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

If set to `true` on a `Trunk` DN and SIP Server sends an outgoing `INVITE` message to this trunk, the greeting is not started and the extension's greeting parameters are added to the outgoing `INVITE` in a specific header. If set to `false`, SIP Server behavior is not changed.

sip-disable-unreliable-sdp

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

When trying to establish a third-party recorder connection for an outbound call, SIP Server can ignore unreliable `18x` response messages containing the SDP by setting this option to `true` on the `Trunk` DN for outbound calls. If set to `false`, this feature is disabled.

sip-early-dialog-mode

Default Value: `0`
Valid Values: `0`, `1`
Changes Take Effect: At the next incoming dialog
Related Feature: "Early Media for Inbound Calls" on page 232
Related Options: `charge-type` and `sip-server-inter-trunk`

For devices that support an offer/answer exchange using the `UPDATE` method, SIP Server will send the `UPDATE` to the called device if the `Trunk` DN is configured with `sip-early-dialog-mode` set to `1`. If set to `0`, this functionality is disabled.

The following scenarios are supported:

- Transfer is completed to a ringing destination.
- Conference is completed to a ringing destination.
- Alternate call when destination is ringing.
- `SendDTMF`, `Hold`, or `Retrieve` on the calling party.

Note: For compatibility reasons, DNs with the option `sip-server-inter-trunk` set to `1` also support early media, if support for early media (`PRACK`) is reported in the `Allow` header.

sip-enable-100rel

Default Value: `true`
Valid Values:

`true`	SIP Server advertises support for `100rel`, and requires it whenever the other side indicates support.
`false`	SIP Server does not negotiate support for the reliability of provisional responses.

Changes Take Effect: Immediately

If set to `true`, SIP Server places the option tag `100Rel` inside the `Supported` header of outgoing initial `INVITE` requests. This informs SIP clients that SIP Server is able to process provisional responses reliably.

sip-enable-diversion

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Next request
Related Feature: "Diversion Header" on page 188

For call forwarding or call redirection through 3pcc request, this option specifies whether the `Diversion` header will be included in `INVITE` sent to the destination DN. If this option is set to `true` on the destination DN, then SIP Server includes the `Diversion` header. Similarly, when SIP Server receives a `302 Moved Temporarily` response that includes the `Diversion` header, this option controls whether SIP Server will forward the `Diversion` header in the resulting `INVITE` message.

sip-enable-ivr-metadata

Default Value: No default value
Valid Values: `true, false`
Changes Take Effect: On the next call

This option is used for IVR recording call scenarios. Specifies whether SIP Server passes its Application name in the initial `INVITE` message (in the `X-Genesys-sipsAppName` header) to Media Server. If this option is set to `true`, SIP Server includes its Application name in the custom header of the `INVITE` that it sends to Media Server. If this option is set to `false`, SIP Server does not include its Application name in the initial `INVITE` sent to Media Server. This option applies to DNs of type `Trunk, Voice over IP Service (msml), Trunk Group`, and `Voice Treatment Port`.

Notes:
- If the IVR recording feature is enabled, then it is not required to explicitly enable recording by setting the `record` option to `true` on DNs representing GVP, such as `Trunk, Trunk Group`, or `Voice Treatment Port`. Recording is started by the VXML application running on the Media Server.
- This option can also be configured at the Application level. The DN-level setting takes precedence over the Application-level setting.

sip-enable-moh

Default Value: No default value
Valid Value: `true, false, na`
Changes Take Effect: On the next call (if the value is empty, changes do not take effect until after application restart)
Related Feature: “Customizing Music on Hold and in Queue” on page 179

Set this option to `true` to enable music-on-hold for any party engaged with this device in the call.

If this option is set to `false` in the device configuration, it disables the `music-on-hold treatment` for any party that is engaged with this device in the call, even if the device sends an `INVITE` request containing a hold SDP. If you set this option to na (non-applicable), SIP Server processes an `INVITE` with a hold SDP as a regular `INVITE`, by simply propagating that `INVITE` to the opposite party without attempting to apply music.

Genesys recommends setting this option to either `false` or `na` on `Trunk` DNs that represent gateways. This ensures that `INVITE` requests containing hold SDPs sent from these gateways will not trigger music-on-hold to be played on agent DNs.

The `na` value can be used on `Trunk` DNs only.

Note: This option can also be configured at the Application level. The DN-level setting takes precedence over the Application-level setting.

sip-enable-replaces

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

This option applies only to outbound `INVITE` messages and works as follows:

- If this option is set to `true`, SIP Server sends the `replaces` tag in the `Supported` header in `INVITE` messages.
- If this option is set to `false`, SIP Server does not send the `replaces` tag in the `Supported` header in `INVITE` messages.

sip-enable-sdp-codec-filter

Default Value: `false`
Valid Values:

`true`	SIP Server modifies the SDP message body during SIP renegotiation.
`false`	SIP Server does not modify the SDP message body.

Changes Take Effect: On the next call

Related Option: `audio-codecs`

Specifies whether SIP Server modifies the SDP message body during SIP renegotiation. All codecs that are not in the list of values for the `audio-codecs` option are deleted from the SDP. As a result, all call center audio traffic is established based on the codecs listed in the `audio-codecs` option.

If `sip-enable-sdp-codec-filter` is set to `true` in the DN configuration, SIP Server, as it propagates the SDP to and from the device represented by this DN, will use as its list of available codecs the value configured in the `audio-codecs` option on the DN rather than on the application. If `sip-enable-sdp-codec-filter` is set to `true` at both the Application and the DN level, then the `audio-codecs` configured in the DN should contain a subset of the `audio-codecs` configured in the Application.

Note: Currently, SIP Server does not support filtration of video codecs.

sip-error-conversion

Default Value: No default value
Valid Values: A comma-separated list of value pairs:
`<received error code>=<converted error code>` (for example, `480=486`, `500=486`), or
`0=<converted error code>` (for example, `0=486`)
Changes Take Effect: Immediately

Set on Destination DN only

When this option is set to `<received error code>=<converted error code>` on the destination DN, SIP Server converts the received error response code to the configured code and sends the converted SIP response code to the origination device. This setting affects the following:

- How the SIP error code is processed by SIP Server.

  For example, SIP error code `486 (Busy Here)` means a destination is busy. SIP error code `408` (`Request Timeout`) received for a DN places a DN in out-of-service state. If this option is set to `408=486` and the DN responds with the `408` error code, SIP Server will not place the DN in out-of-service state.
- The `ErrorCode` that is returned in `EventError` to a routing application when a routing attempt is unsuccessful.

  For example, when a routing destination responds to the `INVITE` message with code `486` (`Busy Here`), SIP Server sends `ErrorCode 231` (`DN is Busy`) to a routing application. If this option is set to `486=404`, SIP Server returns `ErrorCode 71` (`Invalid Called DN`) to a routing application.

Set on Origination DN only

When this option is set to `0`=<converted error code> on an origination DN, SIP Server sends the converted SIP error code if one of the following occurs:

- The destination device fails to respond to the incoming `INVITE` message.
- No active DN is found for SIP Server to send a call.

Note: If a destination DN of type `Extension` fails to respond to the incoming `INVITE` message, SIP Server places the `Extension` DN in out-of-service state regardless of the `sip-error-conversion` setting.

When this option is set to <received error code>= converted error code> on the origination DN, SIP Server converts the received error response code to the configured code and sends the converted SIP response code to the origination device.

This setting affects the SIP Server behavior in the same way as this option is set on the destination DN.

Set on both Destination and Origination DNs

The `sip-error-conversion` option can be configured on both destination and origination DNs at the same time. If an error message is received from a destination device, the option value of the destination device will be used. If no response is received from the destination device or no active DN is found, SIP Server will use the option value configured on the origination device.

The option can be configured on the following levels and in the following order of precedence:

1. DN level
2. Application level

The option can be configured for the following DN types:

- `Trunk`, including GVP Trunk
- `Trunk Group`
- `Extension`
- `Voice over IP Service` with `service-type=softswitch`
- `Voice Treatment Port`

sip-error-overflow

Default Value: An empty string
Valid Values: A string
Changes Take Effect: Immediately

Specifies the destination number to which SIP Server will forward a call if this device responds with a failure (error) to a SIP `INVITE`.

Notes:
- If an outbound call is made from the DN where the `sip-error-overflow` option is defined and an error response is received from the external destination, the outbound call is transferred to the overflow DN.
- Forwarding using the Dial Plan functionality will take priority over the `sip-error-overflow` configuration option.
- The `sip-busy-type` configuration option is not applicable if `sip-error-overflow` is configured, because the device can no longer be busy (it will forward to the `sip-error-overflow` destination instead).
- Setting `sip-error-overflow=gcti:voicemail` is not supported.

sip-filter-media

Default Value: No default value
Valid Values: `none, video`
Changes Take Effect: Immediately
Related Feature: "Video Blocking" on page 381

When set to `video`, SIP Server blocks video media streams in calls coming to or originating from this DN. When set to `none`, SIP Server does not block video media streams, even if the `sip-filter-media` option is enabled at the Application level. The option can be configured on DNs of type `Extension`, `Trunk`, `Trunk Group`, or `Voice over IP Service`.

Note: This option can also be configured at the Application level. The DN-level setting takes precedence over the Application-level setting.

sip-from-pass-through

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

Specifies whether SIP Server will use the content of the `From` header from the original `INVITE` to generate the content for the `From` header in the outgoing `INVITE` message.

When set to `true`, this option takes precedence over any cpn-controlling option or the `CPNDigits` key in `AttributeExtensions` of a T-Library request.

sip-hold-rfc3264

Default Value: `false`
Valid Values:

`true`	RFC3264-compliant implementation.
`false`	RFC2543-compliant implementation.
`passthrough`	Prevents SIP Server from changing `sendonly` and `recvonly` SDP attributes to inactive in the answering SDP; in other words, SIP Server simply passes these attributes through unchanged.

Changes Take Effect: On the next call

Specifies which implementation of hold media SDP is used by SIP Server for hold operations.

Note: This option can also be configured at the Application level. The DN-level setting takes precedence over the Application-level setting.

sip-oos-enabled

Default Value: `true`
Valid Value: `true, false`
Related Feature: "Passive Out-of-Service Detection" on page 239
Related Option: `recovery-timeout` on page 602

Specifies whether a DN can be placed into the out-of-service state in case of SIP failure. DNs can be placed in this state by default. If set to `false`, setting to the out-of-service state is disabled for this DN.

sip-pass-body

Default Value: An empty string
Valid Values: `cid`
Changes Take Effect: On the next call
Related Feature: "Caller Information Delivery Content for AT&T Trunks" on page 117

Specifies the content type that should be passed in the multipart body of the origination `INVITE` to this device, if that content type is received from the caller.

- If set to an empty string (default), SIP Server does not send any special content types.
- If set to `cid`, SIP Server sends the CID body to the DN.

This option...

- ...does not affect SDP.
- ...is supported on `Trunk` DNs, `Trunk Group` DNs, and `VoIP Service` DNs with `service-type` set to `msml` and `voicemail`.

sip-pass-xfer-params-enabled

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

If set to `true`, SIP Server passes Request-URI parameters received in the `Refer-To` header of the incoming `REFER` request to the Request-URI of the outgoing `INVITE` request.

If set to `false`, SIP Server does not pass Request-URI parameters received in the `Refer-To` header of the `REFER` request to the outgoing `INVITE` request.

This option can be applied to a DN object that is the target of the `REFER-INVITE` transfer.

sip-preserve-contact

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately with the exception (see the option description)

Specifies whether SIP Server preserves session information (a cookie) that is appended to the user-info part of the `Contact` header in `REGISTER` requests. If you set this option to `true`, SIP Server preserves the cookie from the `REGISTER` request, and then includes the cookie in the `Request-URI` of the outgoing `INVITE` request.

The `sip-preserve-contact` option affects DNs that contain the username in the `contact` option. If the `sip-preserve-contact` option is set to `true`, SIP Server uses the username from the configured contact in the `Request-URI` of an outgoing `INVITE` message. If you change the `sip-preserve-contact` option value, Genesys recommends restarting SIP Server for changes to take effect.

Genesys does not recommend using the `sip-preserve-contact` option at the DN level if a DN has the `contact` option containing a username.

Note: This option can also be configured at the Application level. The DN-level setting takes precedence over the Application-level setting.

sip-progress-response-code

Default Value: `200`
Valid Values: `100-699`
Changes Take Effect: On the next call

Specifies the response code that SIP Server sends to an incoming re-`INVITE` or `REFER` message, which arrives when a dialog with Media Server is in progress. If set to `200`, SIP Server responds with a `200 OK` message containing the latest SDP (backward compatible behavior). If set to a value in a range of `400-699`, SIP Server rejects a re-`INVITE` message with a respective error code.

sip-proxy-uri-parameters

Default Value: `false`
Valid Values: true, false
Changes Take Effect: On the next call

Specifies whether SIP Server will forward URI parameters from an incoming `INVITE` request to an outgoing `INVITE`. If you set this option to true, SIP Server will forward all `Request-URI` parameters from the incoming `INVITE` request to the outgoing `INVITE` message.

For example, you can use this option to enable network requests for media services provided by Genesys Media Server. For details, see Table 15: Enabling Network Requests for Media Services, on page 96.

sip-reinvite-action

Setting: `TServer` section, the Trunk DN, the VOIP Service DN with `service-type=softswitch`
Default Value: `default`
Valid Values: `default, after-hold`
Changes Take Effect: On the next call

Specifies how SIP Server processes a non-hold re-`INVITE` message from a party that is connected to the music-on-hold service while the process of placing a call on hold is not fully completed. When set to `default`, SIP Server responds to the re-`INVITE` with `200 OK` containing the latest known SDP. When set to `after-hold`, SIP Server sends a `100 Trying` message and waits for a hold procedure to be fully completed. After that the re-`INVITE` message is propagated to Media Server triggering a new SDP offer/answer exchange with Media Server.

sip-rel-200-retransmit

Default Value: `false`
Valid Values: `false, true`
Changes Take Effect: On the next call

Specifies if SIP Server retransmits `200 OK` in response to an `INVITE` message on reliable transports if `ACK` is not received for `200 OK`. (The default value of this option (`false`) enables the previous behavior in SIP Server.)

Note: This option can also be configured at the Application level. The DN-level setting takes precedence over the Application-level setting.

sip-replaces-mode

Default Value: `0`
Valid Values: `0, 1, 2, 3`
Changes Take Effect: On the next call
Related Feature: "TCompleteTransfer using REFER or REFER with Replaces" on page 171

Specifies the SIP method used by SIP Server to complete a two-step transfer.

- With a value of `0`, SIP Server uses the `REFER` method if the `transfer-complete-by-refer` option is set to `true`.
- With a value of `1`, SIP Server uses the `REFER` method with `Replaces` if the `Allow` header contains `REFER` as a supported method and the `Supported` header contains `Replaces`. If `REFER` with `Replaces` is not supported by a device, then `TCompleteTransfer` will be performed using the `REFER` method. If a device does not support the `REFER` method, then the transfer will be completed using the re-`INVITE` method.
- With a value of `2`, SIP Server uses the `REFER` method with `Replaces` to process `TCompleteTransfer`. The `Allow` and `Supported` headers will not be analyzed.
- With a value of `3`, when the DN-level `sip-server-inter-trunk` option is set to `true`, SIP Server uses the re-`INVITE` method instead of the `REFER` method for transfers and call routing.

Note: For this functionality to work, the `refer-enabled` option must be set to `true` in the DN from which a call party is transferred to another destination during a two-step transfer.

sip-request-oos-timeout

Default Value: `0`
Valid Values: `0-31` (seconds)
Changes Take Effect: Immediately
Related Feature: "DNS Name Resolution" on page 217

Controls the length of time, in seconds, SIP Server continues trying requests for a new SIP transaction. After the timeout expires, SIP Server considers the transport as failed, and will instead try the next transport type.

If the value is set to `0`, or no value is specified, SIP Server does not start this timer.

If the value is set to any number higher than `0`, SIP Server will continue retrying a transaction for the specified length of time; after the timer expires, SIP Server will instead try the next active destination, for the next SIP transaction.

Note: Genesys recommends setting this timeout to a shorter length than the default SIP timeout (32 seconds), or the value of the `sip-invite-timeout` and `sip-invite-treatment-timeout` options.

sip-response-msml-oos

Default Value: An empty string
Valid Values: Valid SIP response code between 400 and 699, inclusive
Changes Take Effect: On the next call

Specifies the SIP response code that SIP Server sends in response to an incoming `INVITE`. This option takes effect only for inbound calls received when the MSML DN is out of service. It is supported on Trunk DNs only. It must be set on the inbound trunk and applies to calls for which this trunk is used as an origination device. If the option is not set, or set to an invalid value, this feature is disabled.

The `sip-response-msml-oos` option can be configured with the existing DN-level option `sip-error-conversion`, when the MSML service is available but a response to an `INVITE` requires a SIP response code. For example, if `sip-response-msml-oos = 503`, `sip-error-conversion = 404=603`, and a call is made to an unknown DN, SIP Server will respond to an incoming `INVITE` with the 603 SIP message.

sip-ring-tone-mode

Default Value: `0`
Valid Values: `0`, `1`, `2`
Changes Take Effect: Immediately

When set to `0`, SIP Server connects Media Server to a call to play an audio ringtone. When set to `1`, SIP Server waits for a response from the called device, and connects Media Server to a call to play an audio ringtone, only when the returned response cannot be used as the offer to a calling device.

When set to `2`, SIP Server plays an audio ringtone only to an inbound external call, by connecting Media Server, before the call is placed to an agent. This option is set in the inbound `Trunk` DN.

Notes:
- SIP Server does not support internal ringtones in conference scenarios where the `sip-ring-tone-mode` option is set to `1`. In this case, SIP Server provides a ringtone only if the endpoint returns an SDP in the provisional message.
- This option can be set at both the SIP Server Application level and at the Switch/DN level. The setting at the Switch/DN level takes precedence over the Application-level setting.

sip-route

Default Value: No default value
Valid Values: Any valid SIP URI
Changes Take Effect: Immediately

Specifies the default SIP URI SIP Server uses to route SIP messages involving this DN, which should be registered in IMS (and so configured with enable-ims set to `true`).

sip-route-active-transport

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

When set to `true`, SIP Server updates the `Route` header with the active Resource Manager (RM) contact. This option must be set to `true` on all DNs that are configured to point to the RM pair:

- `Trunk`
- `Trunk Group`
- `Voice Treatment Port`
- `Voice over IP service` with `service-type=msml`
- `Voice over IP service` with `service-type=voicemail`

sip-server-inter-trunk

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Features: “Trunk Optimization for Multi-Site Transfers” on page 376 and “Early Media for Inbound Calls” on page 232
Related Options: “charge-type” on page 564 and “sip-early-dialog-mode” on page 615

When this option set to `true`, depending on the scenario, SIP Server determines whether to complete the transfer operation using the `REFER` or `INVITE` request with the `Replaces` header.

sip-signaling-chat

Default Value: `session`
Valid Values: `session, none`
Changes Take Effect: Immediately
Related Feature: “Instant Messaging” on page 250

Specifies the chat mode for the Instant Messaging (IM) DN on which this option is configured.

If you set this option to `session`, when the first SIP dialog containing an IM SDP is created, the `MESSAGE` requests are exchanged only in this dialog. You must use this value for Microsoft Office Communicator endpoints.

If you set this option to `none`, SIP Server does not send a SIP `MESSAGE` to a SIP Endpoint during IM. Chat communication (IM) will be provided to the agent desktop by only the T-Library protocol.

sip-to-pass-through

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

Specifies whether SIP Server uses the content of the `To` header from the original `INVITE` request, in cases where SIP Server passes through the call.

Note: This option does not work when enabled (set to `true`) on the destination agent for a two-step transfer initiated by another agent with `dual-dialog-enabled` set to `false`.

sip-transfer-complete-message

Default Value: An empty string
Valid Values: `1XX, 18X, 180, 183, 200`
Changes Take Effect: Immediately
Related Option: `sip-transfer-complete-timeout`

Defined for a DN through which SIP Server sends a `REFER` request, this option specifies on which message SIP Server completes `REFER`-based transfers. When this option is set to:

- An empty string (no value is specified)—SIP Server completes the `REFER`-based transfer after receiving a `200 OK` message within the `NOTIFY` request (the default behavior).
- `1XX`—SIP Server completes the `REFER`-based transfer after receiving any provisional response within the `NOTIFY` request.
- `18X`—SIP Server completes the `REFER`-based transfer after receiving a `180` or `183` provisional response within the `NOTIFY` request.
- `180`—SIP Server completes the `REFER`-based transfer after receiving a `180 Ringing` provisional response within the `NOTIFY` request.
- `183`—SIP Server completes the `REFER`-based transfer after receiving a `183 Session Progress` provisional response within the `NOTIFY` request.
- `200`—SIP Server completes the `REFER`-based transfer after receiving a `200 OK` provisional response within the `NOTIFY` request, even if the transfer destination terminates its dialog (according to RFC 5589).

Note: Use the Application-level `sip-transfer-complete-timeout` configuration option to specify how many seconds SIP Server waits for a `NOTIFY` message before considering the Out Of Signaling Path transfer as failed.

sip-trying-timeout

Default Value: An empty string
Valid Values: `0-256`
Changes Take Effect: On the next call
Related Feature: “Setting SIP INVITE Timeout for Individual DNs” on page 108

Specifies the period of time (in seconds) that a SIP call remains in an active state if the only provisional response received was `100 Trying`. When this timeout expires, the call is either sent to the DN configured in the `no-response-dn` option, or is released if that option is not configured. If the `sip-trying-timeout` option is not specified, the value of the Application-level option `sip-invite-timeout` is used instead. If the `sip-invite-timeout` option is set to `0`, the default value of `32` seconds is used.

The `sip-trying-timeout` option can be set on the following DN types:

- `Extension`
- `Trunk`
- `Voice over IP Service` with `service-type=softswitch`

sip-uri-params

Default Value: No default value
Valid Values: A string that contains valid URI parameters
Changes Take Effect: On the next call
Related Feature: “Enabling Additional Parameters in Request-URI” on page 183

Specifies which URI parameters SIP Server will add to the initial `INVITE` request to start a dialog with this DN. If configured, SIP Server sends the specified parameters in the initial `INVITE` to this DN. To include multiple parameters, enter in a semicolon separated list.

sip-user-agent

Default Value: No default value
Valid Values: A valid string or the special character `*`
Changes Take Effect: On the next call

Specifies whether SIP Server includes the `User-Agent` header in all request messages that it sends. The value for this option can contain the following placeholders:

- `$VERSION$` = will be replaced with the current SIP Server build
- `$APP-NAME$` = will be replaced with the name of the application in the environment

You can also use the special value *, which is equivalent to `Genesys SIP Server $VERSION$ ($APP-NAME$)`.

stranded-calls-overflow

Default Value: No default value
Valid Values: `<destination_number>`, `default` or `<empty string>`, `recall`, `release`, `none`
Changes Take Effect: Immediately
Related Feature: “Alternate Routing for Stranded Calls” on page 106
Related Option: `stranded-call-redirection-limit`

Specifies a list of actions that SIP Server attempts to take for calls stranded on ACD Queues. You can configure these actions globally for all queues (at the Application-level) or individually for a particular `ACD Queue` DN. Configure the actions using a comma-separated list of valid values; SIP Server tries to process each item in the list sequentially, moving on to the next item if any action fails, and stopping after the first successful action begins (subsequent failure of the successful action does not restart the list).

Note: For a description of the valid values and their related SIP Server actions, see “Stranded Calls Overflow Valid Values” on page 107.

stranded-on-arrival-calls-overflow

Default Value: No default value
Valid Values: `<destination_number>`, `default` or `<empty string>`, `recall`, `release`, `none`
Changes Take Effect: Immediately
Related Feature: “Alternate Routing for Stranded Calls” on page 106
Related Option: `stranded-call-redirection-limit`

Specifies a list of actions that SIP Server attempts to take for calls arriving on ACD Queues that have no logged-in agents. You can configure these actions globally for all queues (at the Application-level) or individually for a particular `ACD Queue` DN. Configure the actions using a comma-separated list of valid values; SIP Server tries to process each item in the list sequentially, moving to the next item if any action fails, and stopping after the first successful action begins (subsequent failure of the successful action does not restart the list).

Note: For a description of the valid values and their related SIP Server actions, see “Stranded Calls Overflow Valid Values” on page 107.

subscription-id

Default Value: No default value
Valid Values: Any valid string
Changes Take Effect: Immediately

This parameter is required for SIP Server integration with GVP.

For GVP 7.6, if the GVP Resource Manager (RM) is configured as a redirection server, you must configure the `subscription-id` option on all DNs

representing GVP media servers (or ports). Set the value of this parameter to `GVP`. In this configuration, GVP Resource Manager subscribes to notifications about the call status at SIP Server, which works as a notification provider. SIP Server sends a SIP `NOTIFY` message to the GVP RM at the end of every call, supporting GVP resource management functionality.

In 8.0, GVP can be integrated with SIP Server using MSML protocol. In this case, the GVP RM is represented in the Configuration Layer as a `Trunk` DN with the required option `subscription-id` properly configured. You must set the value of this option to `msml`. In this configuration, SIP Server works as a subscribing client—it activates a subscription to the GVP Resource Manager, and then receives notifications whenever a media server goes out of service, so that SIP Server can perform a recovery for any ongoing media sessions (restarting on remaining active media server instances).

Notes:
- For the Outbound IP Solution, you must not set the value of this option to `GVP`. This value is reserved for GVP 7.6 integrations. If you set the `subscription-id` on a `Trunk Group` DN to this value, SIP Server will not activate the subscription required for the Outbound IP Solution. See Table 77 on page 311 for configuration details.
- You can configure multiple Resource Manager `Trunk Group` DNs on the same tenant as the SIP Server switch. However, Genesys recommends that you plan deployment to use a minimal number of `Trunk Group` DNs for a single tenant—ideally, one per tenant—because SIP Server activates a separate subscription for every `Trunk Group` DN configured for `subscription-id`, which can affect system performance due to the large number of `NOTIFY` messages exchanged between the GVP RM and SIP Server.

subscribe-presence-domain

Default Value: `NULL`
Valid Values: Any valid computer name on the softswitch
Changes Take Effect: Immediately
Related Feature: “Presence from Switches and Endpoints” on page 325

Specifies the subscription domain information for the `Trunk` DN. This option value will be used with the DN name to form the `SUBSCRIBE` request `URI` and the `To` header.

subscribe-presence-from

Default Value: NULL
Valid Values: Any valid SIP URI
Changes Take Effect: Immediately
Related Feature: "Presence from Switches and Endpoints" on page 325

Specifies the subscription endpoint information. This option value will be used to form the From: header in the SUBSCRIBE request to the softswitch.

Note: For softswitches such as Microsoft LCS and Asterisk, the username part of this SIP URI must not be configured in the softswitch.

subscribe-presence-expire

Default Value: 1800
Valid Values: Any valid positive integer from 10 to 259200 (from 10 seconds to 72 hours)
Changes Take Effect: Immediately
Related Feature: "Presence from Switches and Endpoints" on page 325

Specifies the subscription renewal interval (in seconds).

subscribe-presence

Default Value: NULL
Valid Values: publish, or the name of the Trunk DN representing the softswitch
Changes Take Effect: Immediately
Related Feature: "Presence from Switches and Endpoints" on page 325

Enables presence subscription and mapping of a presence state to an agent state:

- If set to publish, SIP Server uses presence updates from a PUBLISH SIP request sent by a SIP Endpoint, and maps the presence state from the PUBLISH request to the agent state.
- If set to the name of a Trunk DN that contains the subscription parameters is specified, the enable-agentlogin-presence option (see page 577) must also be configured for the same Trunk DN.

transfer-complete-by-refer

Default Value: false
Valid Values: true, false
Changes Take Effect: On the next call
Related Feature: "TCompleteTransfer using REFER or REFER with Replaces" on page 171

If set to true, this option enables SIP Server to complete a two-step transfer by sending a REFER message to the party in the primary call. SIP Server uses the same content as in the REFER message that is sent for a single-step transfer. For this option to work, you must configure refer-enabled on the Trunk DN to true.

Limitations for this option include the following:

- REFER is not used if the primary party on the consultation call is involved in a conference.
- REFER is not used if the call is currently being recorded.
- This option is not supported on Trunk DNs that are configured between different SIP Server instances, and it is ignored on Trunk DNs where the sip-server-inter-trunk option is set to true.

use-contact-as-dn

Default Value: false
Valid Values: true, false
Changes Take Effect: On the next call

Specifies whether SIP Server will use the username of the Contact header as attribute ThisDN.

use-display-name

Default Value: true
Valid Values: true, false
Changes Take Effect: On the next call

Specifies whether SIP Server takes any action to populate the Display-Name of a call originator in the From header of an outgoing INVITE message with the value of the display-name option:

- If set to false, SIP Server does not take any action.
- If set to true, SIP Server uses the value of the display-name configuration option.

user-data-im-enabled

Default Value: false
Valid Values: true, false
Related Feature: “Instant Messaging” on page 250

Enables the UserData content in the Instant Messaging (IM) for a DN.

Note: Only a subset of the UserData is sent through IM (whatever is contained in the IMDelivery sublist in the KVList in the UserData). In addition to this option, you must also configure the routing strategy so it includes the IMDelivery sub-list in attached UserData.

T-Library clients (for example, Genesys Agent Desktop) are able to see all of the attached UserData.

userdata-map-filter

Default Value: No default value
Valid Values:

*	All data is mapped or blocked.
A list of prefixes	A comma-separated list of prefixes that must match the initial characters of the key in the `UserData` key-value pair. If the list of prefixes contains an * (asterisk), then the * is processed as a prefix.

Changes Take Effect: Immediately
Related option: `userdata-map-filter-mode`
Related Feature: “Mapping SIP Headers and SDP Messages” on page 261

Specifies a prefix (or a list of prefixes) that must match the initial characters of the key in the `UserData` key-value pair. When the initial characters match, then SIP Server either allows or blocks mapping of `UserData` into SIP messages, based on the setting in the `userdata-map-filter-mode` option.

If this option is not specified, no data will be mapped.

Example

If `userdata-map-filter=test` and `AttributeUserData` contains `'test'='value1', 'testlocal'='value2',` and `'generaltest'='value3'`, only key-value pairs `'test'='value1'` and `'testlocal'='value2'` are matched the prefix pattern and considered for mapping. The `'generaltest'='value3'` is ignored because its initial characters do not have the prefix `test`.

userdata-map-format

Default Value: `sip-headers`
Valid Values: `sip-headers, sip-headers-encoded`
Changes Take Effect: For the next request

If set to `sip-headers`, the userdata is passed as a SIP header to the GVP.

If set to `sip-headers-encoded`, the userdata containing special characters—such as a comma (,), percentage (%), and/or a semicolon (;)—sent to the Media Server will be encoded (escaped) and decoded when it is received.

Note: The `userdata-map-format` option must be set to `sip-headers-encoded` in the `TServer` section of the GVP Trunk DN and Voice over IP Service DN (msml) if the userdata contains embedded newlines or other special characters.

use-register-for-service-state

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Determines the `Extension` DN service state after it has been unregistered with SIP Server.

- If set to `true`, the DN is set to the `Out of Service` state in the following scenarios:
  - The SIP `REGISTER` request contains an `Expires` header value equal to `0`.
  - The SIP registration timer has expired.

  An `EventDNOutOfService` message is generated to indicate the DN is currently out of service.

  Note: If the client sending the `REGISTER` request with an `Expires` header of `0` was not previously registered with SIP Server, SIP Server generates a `403 Forbidden` response.

- If set to `false` (or not configured), the DN service state is not set to the `Out of Service` state when it has been unregistered with SIP Server.

voicemail-pattern-<n>

Default Value: No default value
Valid Values: A string pattern in the Asterisk format
Changes Take Effect: At the next established call

Specifies the pattern SIP Server looks for in the `redirectNumber` header of `181 Call Is Being Forwarded` messages received in response to an `INVITE` triggered by a `TMakePredictiveCall`. If SIP Server matches the pattern in the header to the pattern configured in this option, it cancels the call, mapping the reason for the redirection to a Genesys CallState. If SIP Server does not make a match between the header and this option, the call proceeds as normal.

The `redirectNumber` header arrives in the format:

`XYYYYYYYYYYYY`

where `X` provides the reason for the redirection, and `YYYYYYYYYYYY` provides the number where the call is being redirected. SIP Server matches X to the corresponding Genesys `CallState` as follows:

Code	Reason	Genesys CallState
x=1	Call forward; line busy	Busy
x=2	Call forward; no reply	NoAnswer
x=3	Call forward; unconditional	AnsweringMachineDetected
x=4	Call forward; not reachable/Other	AnsweringMachineDetected

SIP Server uses the Asterisk pattern-matching format to match the value of the header to the value of this option. For a full description of Asterisk pattern-matching syntax, see Table 53, "Asterisk Dial Plan Syntax for Pattern Matching," on page 200.

xs-heartbeat-timeout

Setting: `TServer` section, the VOIP Service DN with `service-type=extended`
Default Value: `5`
Valid Values: `2-120`
Changes Take Effect: For the next XS heartbeat request
Related Feature: "Enhanced Handling of XS Requests" on page 214

Specifies the timeout, in seconds, for an XS heartbeat request. The timeout starts when an XS heartbeat request is posted to a SIP Feature Server URL and stops when a response for a heartbeat is received from SIP Feature Server. When the timeout expires, SIP Server counts the number of failures and marks the URL as out of service if the threshold specified by the `xs-missed-heartbeat-threshold` option is reached. The heartbeat timeout must be greater than the `xs-post-timeout` option value.

xs-missed-heartbeat-threshold

Setting: `TServer` section, the VOIP Service DN with `service-type=extended`
Default Value: `3`
Valid Values: `1-10`
Changes Take Effect: Immediately
Related Feature: "Enhanced Handling of XS Requests" on page 214

Specifies the maximum number of failed heartbeat requests that SIP Server receives from a SIP Feature Server URL, before marking the corresponding URL as out of service.

xs-post-timeout

Setting: `TServer` section, the VOIP Service DN with `service-type=extended`
Default Value: `4`
Valid Values: `2-16`
Changes Take Effect: For the next XS dial plan request
Related Feature: "Enhanced Handling of XS Requests" on page 214

Specifies the timeout, in seconds, for an XS request in transit. The timeout starts when the XS request is sent out and stops when a response is received from Feature Server. When the timeout expires, SIP Server either resends the XS request to an alternative Feature Server URL or rejects with a corresponding error if the limit of retries (more than 1) has exceeded. The post timeout must not be more than half of the xs-request-timeout option value.

xs-request-timeout

Setting: `TServer` section, the VOIP Service DN with `service-type=extended`
Default Value: `8`
Valid Values: `4-32`
Changes Take Effect: For the next XS dial plan request
Related Feature: "Enhanced Handling of XS Requests" on page 214

Specifies the timeout, in seconds, that SIP Server waits for a SIP Feature Server response on an XS request. The timeout starts when the XS request is added to the queue and stops when a response is received from SIP Feature Server. When the timeout expires, SIP Server rejects the XS request with a corresponding error. The request timeout must be at least twice as long as the `xs-post-timeout` option value.

GVP Integration Options

This section describes a configuration option specific to the Genesys Voice Platform (GVP) functionality with SIP Server. Configure this option in the `extrouter` section of the SIP Server `Application` object.

handle-vsp

Default Value: `no`

Valid Values:

`requests`	The ISCC component of SIP Server will attempt to translate requests related to this DN before submitting them to the service provider.
`events`	The ISCC component of SIP Server will attempt to process events received from the service provider before distributing them to SIP Server clients.
`all`	The ISCC component of SIP Server will handle both the events and requests.
`no`	No processing will take place.

Changes Take Effect: Immediately

Specifies the way SIP Server will handle events from, and requests to, an external service provider registered for a DN using the `AddressType` attribute set to `VSP`.

Reserved Options

Warning! The options documented in this section are reserved for Genesys Engineering and their values cannot be changed.

Table 116: Reserved Configuration Options

Option Name	Option Section
accept-dn-type	Application level > `TServer` section
backup-mode	Application level > `TServer` section
call-max-outstanding	Application level > `TServer` section
call-rq-gap	Application level > `TServer` section
clid-withheld-name	Application level > `TServer` section
correct-rqid	Application level > `TServer` section

Table 116: Reserved Configuration Options (Continued)

Option Name	Option Section
default-dn-type	Application level > `TServer` section
device-rq-gap	Application level > `TServer` section
dn-del-mode	Application level > `TServer` section
emulate-login	Application level > `TServer` section
expire-call-tout	Application level > `TServer` section
kpl-interval	Application level > `TServer` section
kpl-loss-rate	Application level > `TServer` section
kpl-tolerance	Application level > `TServer` section
max-pred-req-delay	Application level > `TServer` section
nas-indication	Application level > `TServer` section
override-switch-acw	Application level > `TServer` section
prd-dist-call-ans-time	Application level > `TServer` section
quiet-cleanup	Application level > `TServer` section
quiet-startup	Application level > `TServer` section
recall-no-answer-timeout	Application level > `TServer` section
recording-client-stop-enable	Application level > `TServer` section
reg-delay	Application level > `link-control` section
reg-interval	Application level > `TServer` section
reg-silent	Application level > `link-control` section
rq-conflict-check	Application level > `TServer` section
sync-emu-agent	Application level > `TServer` section
unknown-xfer-merge-udata	Application level > `TServer` section
wrap-up-threshold	Application level > `TServer` section

Changes from Release 8.0 to Release 8.1

Table 117 lists the configuration options that:

- Are new or changed in the 8.1 release of SIP Server
- Have been added or changed since the most recent 8.0 release of this document

If a configuration option has been replaced with another that enables the same functionality, the new option name and its location in this chapter are noted.

Note: Starting in Deployment Guide version 8.1.101.35, Table 117 is no longer updated. For new and updated options, see "Document Change History" on page 19.

Table 117: Option Changes from Release 8.0 to Release 8.1

Option Name	Option Values	Type of Change	Details
Application Level > TServer Section			
agent-emu-login-on-call	true, false	New in 8.1.0	See the option description on page 439.
agent-logout-on-unreg	true, false, emu-only	New in 8.1.0	See the option description on page 440.
agent-logout-reassoc	true, false	New in 8.1.0	See the option description on page 441.
alternate-route-profile	String	New in 8.1.1	See the option description on page 443.
bsns-call-dev-types	String	New in 8.1.0	See the option description on page 448.
call-monitor-acw	String	New in 8.1.1	See the option description on page 449.
call-observer-with-hold	true, false	New in 8.1.0	See the option description on page 449.
cancel-monitor-on-unpark	true, false	New in 8.1.0	See the option description on page 450.
capacity-sip-error-code	400–699	New in 8.1.1	See the option description on page 450.
capacity-tlib-error-code	Integer	New in 8.1.1	See the option description on page 450.
clamp-dtmf-allowed	true, false	New in 8.1.1	See the option description on page 451.
connect-nailedup-on-login	String	New in 8.1.1	See the option description on page 451.
control-remote-vip-scripts	String	New in 8.1.1	See the option description on page 452.
control-vip-scripts	String	New in 8.1.1	See the option description on page 452.

Table 117: Option Changes from Release 8.0 to Release 8.1 (Continued)

Option Name	Option Values	Type of Change	Details
default-route-point-order	String	New in 8.1.1	See the option description on page 456.
disable-media-before-greeting	true, false	New in 8.1.1	See the option description on page 457.
disconnect-nailedup-timeout	Integer	New in 8.1.1	See the option description on page 457.
drop-nailedup-on-logout	true, false	New in 8.1.0	See the option description on page 458.
dr-forward	String	New in 8.1.0	See the option description on page 458.
dr-peer-location	String	New in 8.1.1	See the option description on page 459.
dr-peer-trunk	Integer	New in 8.1.0	See the option description on page 459.
enable-busy-on-routed-calls	true, false	New in 8.1.0	See the option description on page 461.
enable-ims	true, false	New in 8.1.0	See the option description on page 461.
enable-strict-location-match	true, false	New in 8.1.1	See the option description on page 463.
feature-code-park	Integer	New in 8.1.1	See the option description on page 469.
feature-code-pickup	Integer	New in 8.1.1	See the option description on page 469.
feature-code-retrieve	Integer	New in 8.1.1	See the option description on page 469.
fmfm-confirmation-digit	Integer	New in 8.1.1	See the option description on page 470.
fmfm-confirmation-timeout	Integer	New in 8.1.1	See the option description on page 471.
fmfm-prompt-file	String	New in 8.1.1	See the option description on page 471.
fmfm-trunk-group	String	New in 8.1.1	See the option description on page 471.
force-p-early-media	true, false	New in 8.1.0	See the option description on page 472.
graceful-shutdown-sip-timeout	String	New in 8.1.1	See the option description on page 472.
ims-puid-domain	String	New in 8.1.0	See the option description on page 478.
ims-sip-domain	String	New in 8.1.0	See the option description on page 478.
ims-sip-params	String	New in 8.1.0	See the option description on page 479.
ims-use-term-legs-for-routing	String	New in 8.1.1	See the option description on page 479.
init-dnis-by-ruri	true, false	New in 8.1.1	See the option description on page 481.
keep-mute-after-conference	true, false	New in 8.1.0	See the option description on page 483.

Table 117: Option Changes from Release 8.0 to Release 8.1 (Continued)

Option Name	Option Values	Type of Change	Details
max-parking-time	Integer	New in 8.1.1	See the option description on page 485.
monitor-consult-calls	String	New in 8.1.0	See the option description on page 485.
monitor-party-on-hold	true, false	New in 8.1.1	See the option description on page 485.
msml-location-alarm-timeout	String	New in 8.1.1	See the option description on page 486.
msml-record-support	true, false	New in 8.1.0	See the option description on page 487.
music-listen-disconnect	String	New in 8.1.1	See the option description on page 488.
music-on-pbxpark	String	New in 8.1.1	See the option description on page 489.
mwi-implicit-notify	true, false	New in 8.1.0	See the option description on page 490.
mwi-subscribe-vmb	true, false	New in 8.1.1	See the option description on page 492.
network-monitoring-timeout	Integer	New in 8.1.1	See the option description on page 492.
operational-stat-timeout	true, false	New in 8.1.1	See the option description on page 493.
overflow-location-map	String	New in 8.1.1	See the option description on page 494.
overload-ctrl-call-tapplytreatment-requests-rate	Integer	New in 8.1.1	See the option description on page 494.
overload-ctrl-call-trequests-rate	Integer	New in 8.1.1	See the option description on page 495.
overload-ctrl-call-tupdateuserdata-requests-rate	Integer	New in 8.1.1	See the option description on page 495.
overload-ctrl-trequests-rate	Integer	New in 8.1.1	See the option description on page 496.
p-asserted-identity	String	New in 8.1.0	See the option description on page 496.
privacy	String	New in 8.1.0	See the option description on page 499.
reason-in-extension	true, false	New in 8.1.0	See the option description on page 500.
record-after-merge	true, false	New in 8.1.0	See the option description on page 500.
record-moh	true, false	New in 8.1.1	See the option description on page 502.
recording-failure-alarm-timeout	Integer	New in 8.1.1	See the option description on page 502.
refer-enabled	true, false	New in 8.1.0	See the option description on page 503.

Table 117: Option Changes from Release 8.0 to Release 8.1 (Continued)

Option Name	Option Values	Type of Change	Details
registrar-default-timeout	Integer	Changed in 8.1.0	See the option description on page 503.
releasing-party-report	true, false	New in 8.1.0	See the option description on page 504.
resolve-external-contact	true, false	New in 8.1.1	See the option description on page 505.
resolve-internal-rp-by-host	true, false	New in 8.1.1	See the option description on page 505.
resolve-sip-address	true, false	New in 8.1.0	See the option description on page 505.
resource-management-by-RM	true, false	New in 8.1.0	See the option description on page 506.
restart-period	Integer	New in 8.1.0	See the option description on page 506.
route-failure-alarm-high-wm	Integer	New in 8.1.0	See the option description on page 506.
route-failure-alarm-low-wm	Integer	New in 8.1.0	See the option description on page 507.
route-failure-alarm-period	Integer	New in 8.1.0	See the option description on page 507.
send-200-on-clear-call	true, false	New in 8.1.0	See the option description on page 508.
shutdown-sip-reject-code	Integer	New in 8.1.0	See the option description on page 510
session-refresh-enforced	true, false	New in 8.1.1	See the option description on page 509.
silence-detected	String	New in 8.1.0	See the option description on page 510.
sip-3pcc-from-pass-through	true, false	New in 8.1.1	See the option description on page 511.
sip-491-passthrough	true, false	New in 8.1.0	See the option description on page 511.
sip-add-contact-early-dialog	true, false	New in 8.1.0	See the option description on page 511.
sip-address-srv	String	New in 8.1.0	See the option description on page 512.
sip-alert-info	String	New in 8.1.0	See the option description on page 512.
sip-alert-info-external	String	New in 8.1.0	See the option description on page 513.
sip-alert-info-consult	String	New in 8.1.0	See the option description on page 513.
sip-answer-mode	Auto	New in 8.1.0	See the option description on page 514.
sip-enable-call-info-extended	true, false	New in 8.1.1	See the option description on page 517.
sip-enable-gdns	true, false	New in 8.1.0	See the option description on page 517.

Table 117: Option Changes from Release 8.0 to Release 8.1 (Continued)

Option Name	Option Values	Type of Change	Details
sip-enable-rfc3263	true, false	New in 8.1.0	See the option description on page 519.
sip-enable-tcp-keep-alive	true, false	New in 8.1.1	See the option description on page 519.
sip-filter-media	true, false	New in 8.1.1	See the option description on page 523.
sip-from-pass-through	true, false	New in 8.1.0	See the option description on page 523.
sip-iptakeover-monitoring	true, false	New in 8.1.1	See the option description on page 525.
sip-link-type	Integer	Changed in 8.1.0	See the option description on page 526.
sip-max-uui-length	Integer	New in 8.1.0	See the option description on page 527.
sip-max-retry-listen	Integer	New in 8.1.0	See the option description on page 527.
sip-nic-address	String	New in 8.1.1	See the option description on page 527.
sip-nic-monitoring	true, false	New in 8.1.1	See the option description on page 527.
sip-replaces-mode	Integer	New in 8.1.0	See the option description on page 533.
sip-pass-from-parameters	String	New in 8.1.0	See the option description on page 529.
sip-pass-refer-headers	String	New in 8.1.0	See the option description on page 529.
sip-release-call-on-disable-dn	true, false	New in 8.1.1	See the option description on page 532.
sip-referred-by-support	true, false	New in 8.1.1	See the option description on page 531.
sip-referxfer-bye-timeout	Integer	New in 8.1.0	See the option description on page 531.
sip-rel-200-retransmit	true, false	New in 8.1.1	See the option description on page 532.
sip-remote-del-from-conf	true, false	New in 8.1.1	See the option description on page 533.
sip-resubscribe-on-nonotify	true, false	New in 8.1.1	See the option description on page 535.
sip-timer-c-support	true, false	New in 8.1.0	See the option description on page 536.
sip-treatment-dtmf-interruptable	true, false	New in 8.1.0	See the option description on page 541.
sip-vip-script-down	String	New in 8.1.1	See the option description on page 542.
sip-vip-script-up	String	New in 8.1.1	See the option description on page 542.
subscription-event-allowed	String	New in 8.1.0	See the option description on page 544.

Table 117: Option Changes from Release 8.0 to Release 8.1 (Continued)

Option Name	Option Values	Type of Change	Details
timeguard-reduction	Integer	New in 8.1.1	See the option description on page 547
tlib-nic-monitoring	true, false	New in 8.1.1	See the option description on page 548.
use-propagated-call-type	String	New in 8.1.0	See the option description on page 550.
user-data-im-enabled	true, false	Moved to DN-level	See the option description on page 631.
userdata-map-all-calls	true, false	New in 8.1.0	See the option description on page 550.
userdata-map-invite-after-refer	true, false	New in 8.1.1	See the option description on page 551.
vip-state-change-timeout	Integer	New in 8.1.1	See the option description on page 552.
DN Level > AuthClient Section (New in 8.1.0)			
password	String	New in 8.1.0	See the option description on page 559.
username	String	New in 8.1.0	See the option description on page 559.
DN Level > TServer Section			
after-call-divert-destination	String	New in 8.1.0	See the option description on page 559.
agent-reject-route-point	String	New in 8.1.1	See the option description on page 560.
auto-answer-after	String	Not documented previously	See the option description on page 561.
capacity-limit-inbound	true, false	New in 8.1.1	See the option description on page 564.
clamp-dtmf-enabled	true, false	New in 8.1.1	See the option description on page 565.
connect-nailedup-on-login	String	New in 8.1.1	See the option description on page 565.
contact-list	String	New in 8.1.1	See the option description on page 567.
disable-media-before-greeting	true, false	New in 8.1.1	See the option description on page 572.
disconnect-nailedup-timeout	Integer	New in 8.1.1	See the option description on page 573.
dr-oosp-transfer-enabled	true, false	New in 8.1.1	See the option description on page 575.
emergency-backup	String	New in 8.1.0	See the option description on page 576.
emergency-callback-plan	String	New in 8.1.0	See the option description on page 576.

Table 117: Option Changes from Release 8.0 to Release 8.1 (Continued)

Option Name	Option Values	Type of Change	Details
emergency-device	String	New in 8.1.0	See the option description on page 577.
enable-extension-headers	String	New in 8.1.0	See the option description on page 579.
enable-direct-pickup	true, false	New in 8.1.1	See the option description on page 580.
enforce-privacy	id	New in 8.1.0	See the option description on page 581.
enforce-p-asserted-identity	String	New in 8.1.0	See the option description on page 581.
force-register-disable-totag	true, false	New in 8.1.0	See the option description on page 582.
hg-busy-timeout	Integer	New in 8.1.1	See the option description on page 584.
hg-members	String	New in 8.1.1	See the option description on page 584.
hg-noanswer-timeout	Integer	New in 8.1.1	See the option description on page 584.
hg-preferred-site	Integer	New in 8.1.1	See the option description on page 585.
hg-queue-limit	Integer	New in 8.1.1	See the option description on page 585.
hg-queue-timeout	Integer	New in 8.1.1	See the option description on page 585.
hg-type	fork	New in 8.1.1	See the option description on page 586.
oos-error-check	true, false	New in 8.1.1	See the option description on page 593.
override-domain-oosp	String	New in 8.1.0	See the option description on page 595.
override-domain-refer-to	String	New in 8.1.1	See the option description on page 595.
override-domain-referred-by	String	New in 8.1.1	See the option description on page 596.
override-from-on-conf	true, false	New in 8.1.1	See the option description on page 596.
peer-proxy-contact	String	New in 8.1.1	See the option description on page 598.
predictive-timerb-enabled	true, false	New in 8.0.3	See the option description on page 599.
replace-uri-contact	true, false	New in 8.1.0	See the option description on page 605.
sca-preferred-site	String	New in 8.1.1	See the option description on page 608.
session-refresh-enforced	true, false	New in 8.1.1	See the option description on page 609.
shared-line	true, false	New in 8.1.1	See the option description on page 609.
shared-line-capacity	Integer	New in 8.1.1	See the option description on page 609.

Table 117: Option Changes from Release 8.0 to Release 8.1 (Continued)

Option Name	Option Values	Type of Change	Details
shared-line-number	String	New in 8.1.1	See the option description on page 610.
sip-accept-body	String	New in 8.1.1	See the option description on page 610.
sip-alert-info	String	New in 8.1.0	See the option description on page 610.
sip-alert-info-external	String	New in 8.1.0	See the option description on page 611.
sip-answer-mode	Auto	New in 8.1.0	See the option description on page 612.
sip-disable-greeting	true, false	New in 8.1.0	See the option description on page 614.
sip-enable-diversion	true, false	New in 8.1.0	See the option description on page 615.
sip-error-overflow	String	New in 8.1.1	See the option description on page 619.
sip-filter-media	true, false	New in 8.1.1	See the option description on page 619.
sip-from-pass-through	true, false	New in 8.1.0	See the option description on page 619.
sip-pass-xfer-params-enabled	true, false	New in 8.1.0	See the option description on page 621.
sip-proxy-uri-parameters	true, false	New in 8.1.0	See the option description on page 622.
sip-rel-200-retransmit	true, false	New in 8.1.1	See the option description on page 622.
sip-request-oos-timeout	Integer	New in 8.1.0	See the option description on page 623.
sip-to-pass-through	true, false	New in 8.1.0	See the option description on page 626.
sip-transfer-complete-message	String	New in 8.1.0	See the option description on page 626.
sip-uri-params	String	New in 8.1.0	See the option description on page 627.
user-data-im-enabled	true, false	Moved from Application-level	See the option description on page 631.
userdata-map-format	String	New in 8.1.1	See the option description on page 632.
voicemail-pattern-<n>	String	New in 8.1.0	See the option description on page 633.

GENESYS

Part

2 T-Server Common Functions and Procedures

Part Two of this *SIP Server Deployment Guide* familiarizes the reader with T-Server in general. It addresses architectural, functional, and procedural information common to all T-Servers.

The information in Part Two is divided into the following chapters:

- Chapter 8, "T-Server Fundamentals," on page 649, describes T-Server, its place in the Framework 7 architecture, T-Server redundancy, and multi-site issues. It does not, however, provide configuration and installation information.
- Chapter 9, "Multi-Site Support," on page 659, describes the variations available for T-Server implementations across geographical locations.
- Chapter 10, "Common Configuration Options," on page 715, describes log configuration options common to all Genesys server applications.
- Chapter 11, "T-Server Common Configuration Options," on page 737, describes configuration options common to all T-Server types including options for multi-site configuration.

New for All T-Servers in 8.1

The following general changes that have been implemented in the 8.1 release of T-Server:

- T-Server no longer connects to applications that have disabled status in the configuration environment.
- The default value of the `background-processing` configuration option has been changed to `true`. See "background-processing" on page 738 for details.

- T-Server now supports the Unresponsive Process Detection feature. The following configuration options enable this feature:
  - "heartbeat-period" on page 733
  - "hangup-restart" on page 734

  For more information, refer to the *Framework 8.1 Management Layer User's Guide.*
- T-Server now supports IPv6. For more information, refer to the *Framework 8.1 Deployment Guide.*
- T-Server now supports vSphere 4 Hypervisor.
- T-Server now supports Acresso FLEXNet Publisher v11.9 license manager

Note: • Configuration option changes common to all T-Servers are described in "Changes from Release 8.0 to 8.1" on page 764.

GENESYS

Chapter

8 T-Server Fundamentals

This chapter provides general information about T-Server features and functionality and about its configuration and installation.

This chapter is divided into the following sections:

Learning About T-Server

The *Framework Deployment Guide* provides you with a high-level introduction to the role that T-Server plays in the Genesys Framework. If you have already looked through that guide, you may recall that T-Server is the most important component of the Framework Media Layer. The Media Layer enables Genesys solutions to communicate with various media, including traditional telephony systems, voice over IP (VoIP), e-mail, and the Web. This layer also provides the mechanism for distributing interaction-related business data, also referred to as *attached data,* within and across solutions.

Framework and Media Layer Architecture

Figure 38 illustrates the position Framework holds in a Genesys solution.

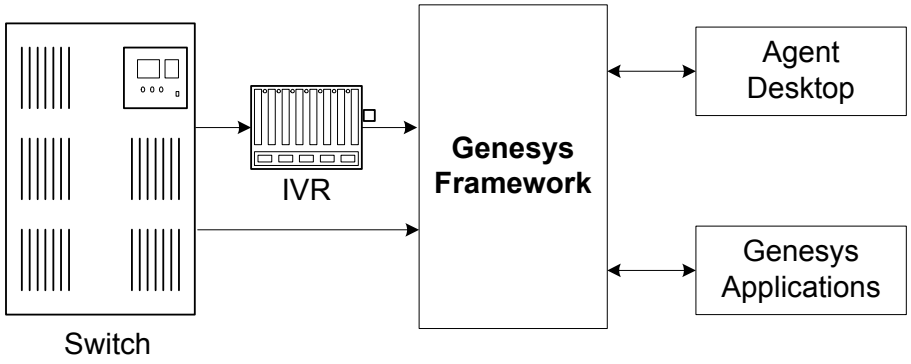

Figure 38: Framework in a Genesys Solution

Moving a bit deeper, Figure 39 presents the various layers of the Framework architecture.

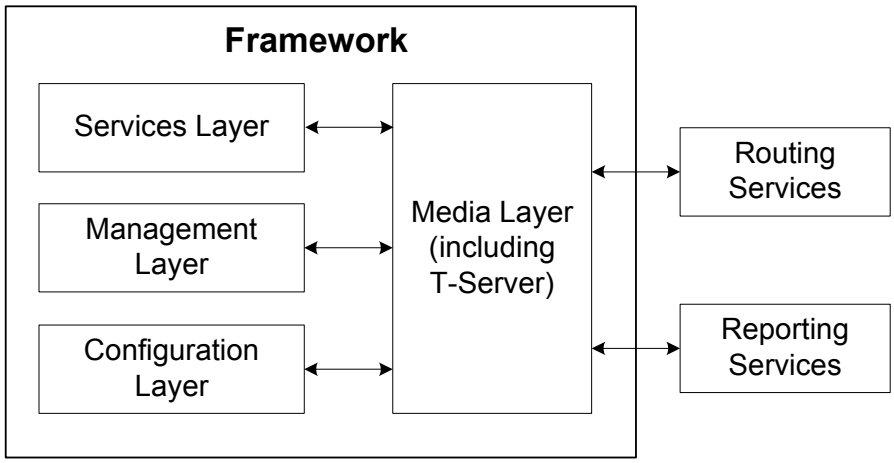

Figure 39: The Media Layer in the Framework Architecture

T-Server is the heart of the Media Layer—translating the information of the media-device realm into information that Genesys solutions can use. It enables your contact center to handle the computer-based form of the interactions that arrive and it translates the information surrounding a customer contact into reportable and actionable data.

Figure 40 presents the generalized architecture of the Media Layer.

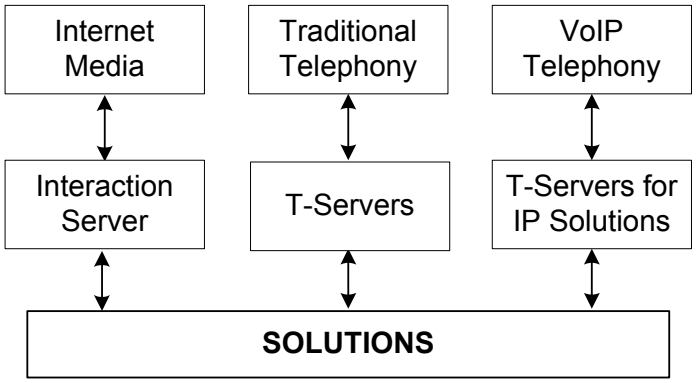

Figure 40: Media Layer Architecture

In addition to being the most important component of the Media Layer, T-Server plays the most significant role in making information about telephony traffic and its data available to Framework as a whole.

One or more components in practically every solution are T-Server clients. Solutions comprise a number of different Genesys software packages, from collections of components for various types of routing to those that allow for outbound dialing to still others. Framework in general, and T-Server in particular, enable these solutions to function in your enterprise.

T-Server has several typical clients: Stat Server, Interaction Concentrator, Universal Routing Server, and agent desktop applications. T-Server gets the information it needs about the enterprise from Configuration Server. Additionally, if you use the Management Layer, T-Server provides its ongoing status and various other log messages to server components of the Management Layer (for instance, allowing you to set alarms).

T-Server Requests and Events

This section outlines the roles that T-Server plays in a contact center. While it is possible to describe roles for all T-Servers, at a detailed level, T-Server's functionality depends on the hardware to which it is connected. (For example, when connected to a traditional switch, it performs CTI functions, but when connected to a VOIP-based telephony device, it controls IP traffic.) The CTI connection is only for the switch.

Details of T-Server Functionality

T-Server is a TCP/IP server that enables intelligent communication between media-specific protocols (such as the various CTI protocols, including CSTA and ASAI) and TCP/IP-based clients of T-Server. Applications that are clients

to T-Server use the T-Library format to transmit requests to T-Server through a TCP/IP socket. T-Server can then either translate those requests to CTI protocol for switch use or relay them directly to other TCP/IP clients.

T-Server performs three general functions in the contact center: Bridging, Messaging, and Interaction Tracking.

Bridging

T-Server acts as a platform-independent interface between media devices and business applications. In the case of a telephony device, for instance, it receives messages from and sends commands to the telephony equipment using either CTI links provided by the switch manufacturer or interface protocols provided by telephony network vendors.

On the client-application end, T-Server offers three models (call model, agent model, and device model) unified for all switches. The core functionality (such as processing an inbound call, an agent login, or a call-forwarding request) translates into a unified application programming interface (API) called T-Library, so that applications do not need to know what specific switch model they are dealing with. On the other hand, T-Library accommodates many functions that are unique to a specific switch, so that client applications are able to derive the maximum functionality offered by a particular switch.

Refer to the *Genesys Events and Models Reference Manual* for complete information on all T-Server events and call models and to the `TServer.Requests` portion of the *Platform SDK 8.x .NET* (or *Java*) *API Reference* for technical details of T-Library functions.

Messaging

In addition to translating requests and events for the client application involved in an interaction, T-Server:

- Provides a subscription mechanism that applications can use to receive notifications about interaction-related and non-interaction-related events within the contact center.
- Broadcasts messages of major importance (such as a notification that the link is down) to all clients.
- Broadcasts messages originated by a T-Server client to other T-Server clients.

The subscription mechanism consists of two parts, the DN subscription and event-type masking. Applications must register for a DN or a set of DNs to receive notifications about all events that occur in association with each registered DN. For example, when two softphone applications are registered for the same DN, and the first application initiates a call from the DN, T-Server notifies both applications that the call is initiated from the DN.

Client applications can also specify one or more types of events, and T-Server will filter out events of the non-specified types and only send events of the

requested types. For example, if agent supervisors are interested in receiving agent-related events, such as `AgentLogin` and `AgentLogout`, they have to mask `EventAgentLogin` and `EventAgentLogout`, provided that a particular T-Server supports these events.

The combination of each client's subscription for DNs and masking of event types defines what messages T-Server distributes to what client.

Interaction Tracking

T-Server maintains call information for the life of the call (or other T-Server-supported media type) and enables client applications to attach user data to the call. Call information includes:

- A unique identifier, connection ID, that T-Server assigns when creating the call.
- Automatic Number Identification (`ANI`) and Dialed Number Identification Service (`DNIS`), if reported by the CTI link.
- User data that a client application (such as an Interactive Voice Response unit or Genesys Universal Routing Server) provides.

Difference and Likeness Across T-Servers

Although Figure 40 on page 651 (and other figures) depicts T-Server that works with telephony systems as a single product, this is a simplification. Because almost every traditional telephony device has its own characteristics and communication protocols, Genesys makes different T-Servers for different telephony systems. (That means your T-Server will not work with another switch.) Thus, all T-Servers play a common role in the architecture, but their specific features differ from implementation to implementation, based on the media device in use.

Despite their switch-based differences, T-Servers for telephony systems are similar to one another in at least one important respect: they are all built with a certain amount of shared software code. This shared code is rolled into a single unit and is called T-Server Common Part (TSCP). TSCP is the central, common component for all T-Servers and has its own Release Note, which is accessible via a hyperlink from your T-Server's Release Note.

Note: This document separates common-code features based on TSCP into separate sections and chapters, such as the "T-Server Common Configuration Options" chapter. These are the options for all T-Servers that TSCP makes available for configuration.

T-Server Functional Steps During a Sample Call

The following example, Figure 41, outlines some basic steps that T-Server might take when a call arrives from outside the contact center. In this scenario,

T-Server starts tracking the call even before it is delivered to the agent. T-Server then informs the selected agent that a call has arrived. When the switch delivers the call to the agent's extension, T-Server presents account information, collected at an Interactive Voice Response (IVR) unit, to the agent at the agent desktop application.

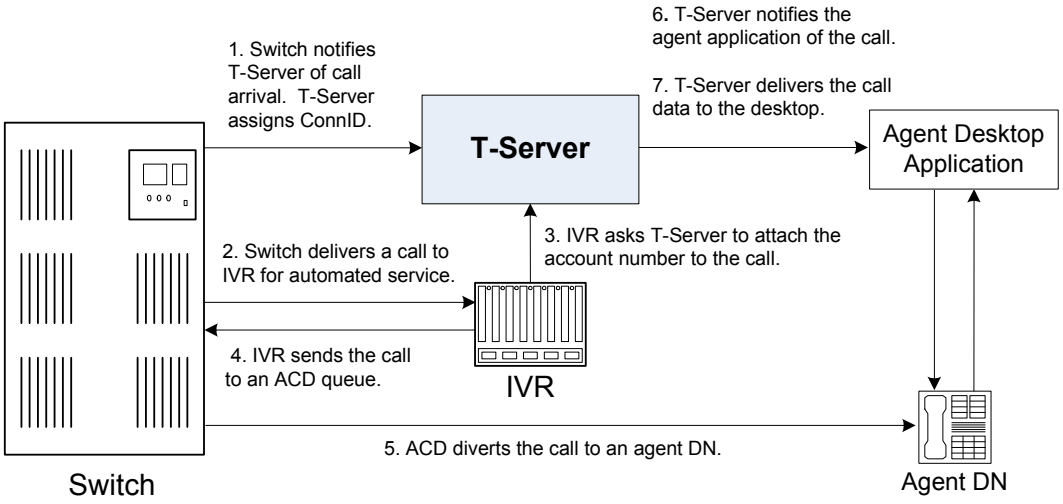

Figure 41: Functional T-Server Steps

Step 1

When the call arrives at the switch, T-Server creates a call in its internal structure. T-Server assigns the call a unique identifier, connection ID.

Step 2

The switch delivers the call to an Interactive Voice Response (IVR) unit, which begins automated interactions with the caller.

Step 3

IVR acquires user information from the caller through prompts and requests T-Server to attach that information to the call. T-Server updates the call with the user information.

Step 4

IVR sends the call to an ACD (Automated Call Distribution) queue.

Step 5

The ACD unit distributes the call to an available agent logged in to a particular DN (directory number).

Step 6

T-Server notifies the agent desktop application that the call is ringing on the agent DN. The notification event contains call data including `ANI`, `DNIS`, and account information that the IVR has collected.

Step 7

The agent desktop application presents the account information, including the name of the person whose account this is, on the agent's screen, so that the agent answering the call has all the relevant information.

These seven steps illustrate just a small part of T-Server's bridging, messaging, and interaction-processing capabilities.

Advanced Disconnect Detection Protocol

Since the 6.0 release of T-Server, the Advanced Disconnect Detection Protocol (ADDP) has replaced the Keep-Alive Protocol (KPL) as the method to detect failures for certain T-Server connections, including connections between two T-Servers and between a T-Server and its clients.

Notes: Starting with release 7.5, the KPL backward-compatibility feature is no longer supported.

ADDP applies only to connections between Genesys software components.

With ADDP, protocol activation and initialization is made on the client's side and you can change these parameters. No additional messages are sent when there is existing activity over the connection. T-Server client applications and the remote T-Server (if any) must be listening to the socket and respond promptly to the polling signal for the connection to be preserved.

If you are going to enable ADDP, you must do it using the `protocol`, `addp-timeout`, `addp-remote-timeout`, and `addp-trace` configuration options. When configuring a timeout, consider the following issues:

- The configured timeout must be at least twice as long as the maximum network latency.
- There may be an interval when T-Server does not check for network activity.
- If the link connection fails but the client is not notified (for example, because the host is turned off, or because a network cable is unplugged), the maximum reaction time to a link-connection failure is equal to double the configured timeout plus the established network latency.

Also keep in mind that the T-Server receiving the polling signal may not respond immediately, and that a delay occurs after the polling signal, while the

response travels from one T-Server to another. If you do not account for these contingencies when configuring a timeout, the connection that ADDP is monitoring will be dropped periodically.

Redundant T-Servers

T-Servers can operate in a high-availability (HA) configuration, providing you with redundant systems. The basics of each T-Server's redundant capabilities differ from T-Server to T-Server. One basic principle of redundant T-Servers is the standby redundancy type, which dictates how quickly a backup T-Server steps in when the primary T-Server goes down.

The Framework Management Layer currently supports two types of redundant configurations: `warm standby` and `hot standby`. All T-Servers offer the `warm standby` redundancy type and, starting with release 7.1, the `hot standby` redundancy type is implemented in T-Servers for most types of switches.

For detailed, up-to-date information on the subject, see the *Genesys Supported Media Interfaces Reference Manual*.

Multi-Site Support

Multi-site configuration implies the existence of two or more switches that belong to the same enterprise or service provider, and that share the Genesys Configuration Database. (In some cases this may include isolated partitions on a given switch served by different T-Servers.) The main goal of T-Server support for multi-site operations is to maintain critical information about a call as it travels from one switch to another.

For instructions on installing and configuring a multi-site environment, including information on the Inter Server Call Control (ISCC) features, please see Chapter 9, "Multi-Site Support," on page 659.

Agent Reservation

T-Server provides support for clients to invoke the agent reservation function, `TReserveAgent()`. This function allows a server application that is a client of T-Server to reserve a DN along with an agent, a `Place`, or both, so that no other T-Server client can route calls to it during a specified reservation interval. Alternatively, when clients use the ISCC feature (see "ISCC Call Data Transfer Service" on page 661), they can use an agent reservation embedded in an ISCC request. (To do so, clients have to specify a certain `Extensions` attribute in an ISCC request when initiating an ISCC transaction. See page 668 for the list of ISCC requests.)

The reservation does not currently prevent the reserved objects from receiving direct calls or calls distributed from ACD Queues; agent reservation is intended as a way of synchronizing the operation of several clients. See `RequestReserveAgent` in the *Platform SDK 8.x .NET (or Java) API Reference* for more details on this function from the client's point of view.

In addition to invoking the `TReserveAgent` function, you can customize the Agent Reservation feature by configuring options in the T-Server `Application` object. See"agent-reservation Section" on page 746 in the "T-Server Common Configuration Options" chapter for more details.

Starting with version 8.1, T-Server supports Agent Reservation failure optimization, to ensure that only agent reservation requests of the highest priority are collected. T-Server responds immediately with the `EventError` message to existing or new reservation requests of a lower priority while collecting the agent reservation requests of the highest priority only. This functionality is controlled with the `collect-lower-priority-requests` configuration option (see page 747).

Client Connections

The number of connections T-Server and SIP Server can accept from its clients depend on the operating system that T-Server runs.

Table 118 lists the number of client connections that SIP Server supports for Windows and Linux operating systems.

Table 118: Number of SIP Server's Client Connections

Operating System	Number of Connections
Linux 32-bit mode	10000 registered agents (aggregated T-Library client connection) or 4,000 registered agents (direct T-Library connections)
Linux 64-bit mode	15,000 registered agents (direct or aggregated T-Library client connections)
Windows Server 32-bit mode	10000 registered agents (aggregated T-Library client connections) or 4,000 registered agents (direct T-Library connections)
Windows 64-bit mode	15,000 registered agents (direct or aggregated T-Library client connections)

GENESYS

Chapter

9 Multi-Site Support

This chapter contains general information about multi-site environments, as well as information on deploying a multi-site environment for your T-Server.

This chapter is divided into the following sections:

Note: Each switch/T-Server combination offers different multi-site options. For details describing your specific switch/T-Server environment, refer to Chapter 11, "T-Server Common Configuration Options," on page 737.

The following instructions apply to both local and remote switches and T-Servers. Because different vendor switches can be installed at the local and remote locations, this chapter covers several, but not all, possible configurations. To help determine which sections of this chapter apply to your situation, refer to Table 119 on page 675 and Table 120 on page 678.

Multi-Site Fundamentals

A multi-site configuration has two or more switches that belong to the same enterprise or service provider and that share the Genesys Configuration Database. (In some cases, this may include isolated partitions on a given switch served by different T-Servers.) The main goal of T-Server support for

multi-site operations is to maintain critical information about a call as it travels from one switch to another.

T-Server supports multi-site operations using its *Inter Server Call Control* (*ISCC;* formerly called External Routing), which supports the following functions:

- **Call matching—**To link instances of a call distributed across multiple sites and to re-attach essential data associated with the call (`ConnID`, `UserData`, `CallType`, and `CallHistory`). The following T-Server features support this capability:
  - ISCC Call Data Transfer Service (active external routing)—when requested by a T-Server client by specifying the desired destination in the location parameter, and also with various ISCC strategies performed by direct dial or by using the Transfer Connect Service. See "ISCC Transaction Types" on page 668 and "Transfer Connect Service Feature" on page 676.
  - Inter Server Call Control/Call Overflow (ISCC/COF) feature (passive external routing)—applicable when calls are overflowed to another site either directly or manually (see page 677).
  - Number Translation feature (see page 681).
  - Network Attended Transfer/Conference (NAT/C) feature (see page 689).

  Note: When ISCC detects call instance reappearance on a given site, the call is assigned a unique `ConnID` and the user data is synchronized with the previous call instances. This ensures that `ConnID`s assigned to different instances of the same call on a given site are unique.

- **Call data synchronization between associated call instances (ISCC Event Propagation)—**To provide the most current data to call instances residing on remote T-Servers. The following T-Server features support this capability:
  - User Data propagation (see page 692)
  - Party Events propagation (see page 693)

  Note: ISCC automatically detects topology loops and prevents continuous updates.

Note: In distributed networks, Genesys recommends using call flows that prevent call topology loops and multiple reappearances of the same call instance. This approach ensures that all T-Servers involved with the call report the same `ConnID`, and also optimizes telephony trunk allocation by preventing trunk tromboning.

The T-Server configuration contains information about other T-Servers with which it will communicate. T-Server uses this information to connect with the other T-Servers. During this "handshake" process, T-Servers exchange information about the following parameters:

- Protocol type
- Switch type
- Server name
- Location name (switch name)
- T-Server role (primary or backup)

To complete the handshake process, T-Servers exchange messages about the current condition of the links to their switches. After the handshake process is complete, T-Server is ready to support a multi-site operation.

ISCC Call Data Transfer Service

Because ISCC supports active external routing, T-Servers that serve different switches (usually on different sites) can exchange call data when a call is passed from one switch to another. With this functionality, T-Server provides its clients with the following additional information about each call received from another switch:

- The connection identifier of the call (attribute `ConnID`).
- Updates to user data attached to the call at the previous site (attribute `UserData`).
- The call type of the call (attribute `CallType`)—In multi-site environments the `CallType` of the call may be different for each of its different legs. For example, one T-Server may report a call as an `Outbound` or `Consult` call, but on the receiving end this call may be reported as `Inbound`.
- The call history (attribute `CallHistory`)—Information about transferring/routing of the call through a multi-site contact center network.

Note: Load-sharing IVR Servers and Network T-Servers cannot be designated as the destination location for ISCC, except when `cast-type` is set to `dnis-pool`. Consult the *Universal Routing 8.1 Deployment Guide* for specific configuration details.

Figure 42 shows the steps that occur during a typical external routing (ISCC) transaction. Note that the location where a call is initially processed is called the *origination location,* and the location to which the call is passed is called the *destination location.*

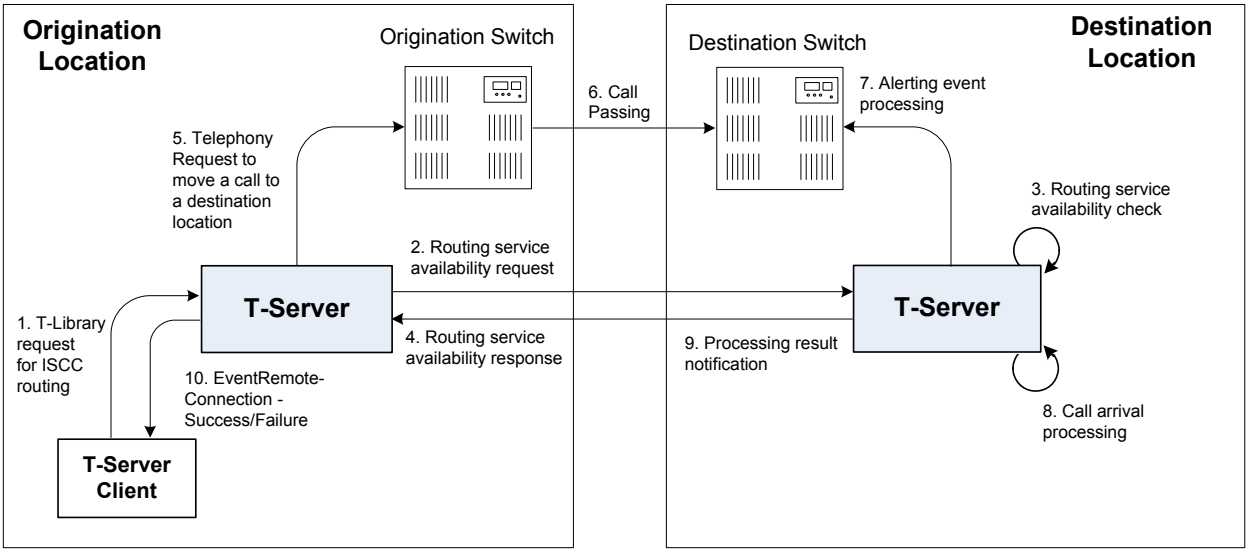

Figure 42: Steps in the ISCC Process

ISCC Call Flows

The following section identifies the steps (shown in Figure 42) that occur during an ISCC transfer of a call.

Step 1

A client connected to the T-Server at the origination location requests this T-Server to pass a call with call data to another location. For this purpose, the client must specify the `location` parameter (Attribute `Location`) when calling a corresponding T-Library function. ISCC processes the following T-Library requests:

- `TInitiateConference`
- `TInitiateTransfer`
- `TMakeCall`
- `TMuteTransfer`
- `TRouteCall`
- `TSingleStepTransfer`

Step 2

Upon receiving a client's request, the origination T-Server checks that the:

1. Connection to the destination T-Server is configured in the origination T-Server `Properties` dialog box.
2. The connection to the destination T-Server is active.

3. The destination T-Server is connected to its link.
4. The origination T-Server is connected to its link.

If these four conditions are met, the origination T-Server determines the transaction type that will be used for passing call data to another location in this transaction. The following possibilities exist:

- The client can request what *ISCC transaction type* (or simply *transaction type*) to use by specifying an appropriate key-value pair in the `Extensions` attribute of the request. The key-value pair must have a key equal to `iscc-xaction-type` and either an integer value as specified in the `TXRouteType` enumeration (see the *Platform SDK 8.x .NET (or Java) API Reference*) or a string value equal to one of the following: `default`, `route`, `direct` (or `direct-callid`), `direct-network-callid`, `direct-notoken`, `direct-ani`, `direct-uui`, `direct-digits`, `reroute`, `dnis-pool`, `pullback`, or `route-uui`.
- If the client does not specify the transaction type in the request or specifies the `default` transaction type, T-Server checks the Switch configuration for the transaction type configured in the `Access Code` (or `Default Access Code`) properties:
  - If the `Route Type` property of the `Access Code` is set to any value other than `default`, T-Server uses the specified value as the transaction type.
  - If the `Route Type` property of the `Access Code` is set to the `default` value, T-Server uses the first value from the list specified in the `cast-type` configuration option configured for the destination T-Server. If no value has been specified for the `cast-type` option, the default value of `route` is used as the transaction type.

> **Note:** For more information on Access Codes and Default Access Code, see "Switches and Access Codes" on page 702.

After the origination T-Server determines the requested transaction type, it determines if the destination T-Server supports this transaction type.

You must list the transaction types T-Server supports in the `cast-type` configuration option.

The origination T-Server issues a request for routing service availability and sends it to the destination T-Server. The T-Server request contains data that should be passed along with the call to the destination location. This data includes the transaction type, `ConnID`, `UserData`, `CallType`, and `CallHistory`.

The timer specified by the `request-tout` configuration option is set when the origination T-Server sends the request. If either the specified timeout expires or the call is abandoned before the origination T-Server receives a response from the destination T-Server, the operation is considered failed. In this scenario, the origination T-Server:

1. Generates a request to the destination T-Server to cancel the request for routing service.

2. Sends `EventError` to the client that requested the service.
3. Deletes information about the request.

Step 3

The destination T-Server receives the request for routing service availability and checks the requested type of routing. Depending on the ISCC transaction type, it stores the request information and, when appropriate, allocates access resources for the coming call. For example, an External Routing Point is allocated when the transaction type is `route`, and an Access Resource of type `dnis` is allocated when the transaction type is `dnis-pool`.

Note: The `resource-allocation-mode` and `resource-load-maximum` configuration options determine how resources are allocated. For option descriptions, refer to Chapter 11, "T-Server Common Configuration Options," on page 737 for option descriptions.

If resources are unavailable, the request is queued at the destination location until a resource is free or the origination T-Server cancels the request. If the request is canceled, the destination T-Server deletes all information about the request.

If resources are unavailable because of incorrect configuration, the destination T-Server returns an error event to the origination T-Server.

Step 4

If resources are available, the destination T-Server generates a positive response and the timer is started for the interval specified by the `timeout` configuration option of the destination T-Server.

Step 5

If the origination T-Server receives a negative response, it sends an `EventError` message to the client and clears all data about the request.

If the origination T-Server receives the confirmation about routing service availability, it processes the client's request and sends a corresponding message to the switch. The timer on the origination T-Server is also started for the interval specified by the `timeout` configuration option of the destination T-Server.

Step 6

The origination switch processes the T-Server request and passes the call to the destination switch.

Step 7

If the call arrives at the destination switch, the switch generates an alerting event.

The destination T-Server waits for the call no longer than the interval specified by the timeout configured on the destination T-Server. If the call is not received at the destination location within this interval, the destination T-Server issues a failure notification to the origination T-Server, deletes all data about the request, and, when appropriate, frees the resources previously allocated for the request.

If either the specified timeout expires or the call is abandoned before the origination T-Server receives a response from the destination T-Server, the operation is considered failed. In this case, the origination T-Server:

1. Generates a request to the destination T-Server to cancel the request for routing service.
2. Responds to the client that requested the service in one of the following ways:
   - If the origination T-Server has already sent a response to the request the client sent in Step 1, the origination T-Server supplements its response with `EventRemoteConnectionFailed`.
   - If the origination T-Server has not yet sent a response to the client, the origination T-Server sends `EventError`.
3. Deletes information about the request.

Step 8

If the destination T-Server matches the arrived call, it updates the `ConnID`, `UserData`, `CallType`, and `CallHistory` attributes with the data received in the request for routing service availability. The connection ID is updated as follows:

The arrived call is assigned the `ConnID` that is specified in the request for routing service availability, but only if this `ConnID` does not coincide with the `ConnID` of a call that has existed at the destination site. If two such `ConnID`s are identical, the arrived call is assigned a new unique `ConnID`.

For `direct-*` transaction types (where the asterisk stands for a `callid`, `uui`, `ani`, or `digits` extension), the call reaches the destination DN directly.

For the transaction types `route` and `route-uui`, the call first arrives at an External Routing Point from which it is routed to the destination DN. The call info is updated when the call reaches the External Routing Point. An External Routing Point is considered free when the first alerting event (`EventQueued` or `EventRouteRequest`) is distributed.

Keep the following issues in mind when using the ISCC feature:

- If routing from a dedicated External Routing Point to the destination DN fails, T-Server considers the transaction failed. However, the `ConnID`, `UserData`, `CallType`, and `CallHistory` attributes are updated. Then, T-Server attempts to route the call to one of the Default DNs configured for this External Routing Point.
- If the destination T-Server did not receive a request for routing service availability, but a call arrives at an External Routing Point, T-Server considers the call to be unexpected and routes the call to the DN specified by the `dn-for-unexpected-calls` configuration option. When no alternative targets are defined, the call remains at the External Routing Point until diverted by the switch or abandoned by the caller.

For `reroute` and `pullback` transaction types, the call returns to the network location. For the `dnis-pool` transaction type, the call reaches the destination DN directly.

Step 9

If, in Step 8, the call does not arrive within the configured timeout, or the transaction fails, the destination T-Server sends a notification of failure to the origination T-Server.

Otherwise, the destination T-Server notifies the origination T-Server that the routing service was successful and deletes all information about the request.

Step 10

The origination T-Server notifies the client that the routing service was successful (or failed) and deletes all information about the request.

Client-Controlled ISCC Call Flow

The following section identifies the steps that occur during a client-controlled ISCC transfer of a call.

Step 1

A client, such as Universal Routing Server (URS), that is connected to the T-Server at the origination location detects a call to be delivered to another destination location.

Step 2

The client chooses a destination location and the target DN for the call. Then, it sends the `TGetAccessNumber` request to the destination T-Server for routing service availability, indicating the target DN and other call context (`ConnID`, `UserData`, and `CallHistory` attributes).

Step 3

The destination T-Server receives the request for routing service availability. Depending on the ISCC transaction type, it stores the request information, including the call context. When appropriate, it allocates access resources for the coming call, such as External Routing Point.

If resources are unavailable, the request is queued at the destination T-Server until an appropriate ISCC resource is free or the client cancels the request. If the request is canceled, the destination T-Server deletes all information about the request.

If resources are unavailable because of incorrect configuration, the destination T-Server returns an `EventError` message to the client.

Step 4

The destination T-Server replies to the client with the `EventAnswerAccessNumber` message, which contains the allocated ISCC resource.

Step 5

The client requests that the origination T-Server delivers the call to the destination location using the allocated access resource.

Step 6

The origination T-Server receives and processes the client's request, and then sends a corresponding message to the switch.

Step 7

The call arrives at the destination switch and is reported to the destination T-Server via CTI. The call is matched by means of ISCC, based on the specified `cast-type` setting and allocated resource, and then the call is assigned a requested call context (such as `ConnID` or call data). Upon successful transaction completion, the destination T-Server notifies the client by sending `EventRemoteConnectionSuccess`.

The destination T-Server waits for the call no longer than the interval specified by the timeout that is configured on the destination T-Server. If the call is not received at the destination location within this interval, the destination T-Server issues a failure notification to the client by sending `EventRemoteConnectionFailed`, deletes all data about the request, and, when appropriate, frees the resources previously allocated for the request.

The destination T-Server notifies the client whether the routing service succeeded or failed by sending either the `EventRemoteConnectionSuccess` or `EventRemoteConnectionFailure`, respectively.

ISCC Transaction Types

As switches of different types provide calls with different sets of information parameters, a single mechanism for passing call data between the switches is not feasible in some cases. Therefore, the ISCC feature supports a number of mechanisms for passing call data along with calls between locations. This section describes ISCC transaction type principles, identifies which transaction types are supported for each T-Server, and defines each transaction type (beginning with "direct-ani" on page 669).

It is important to distinguish the two roles that T-Servers play in an external routing (ISCC) transaction—namely *origination T-Server* and *destination T-Server*.

- The origination T-Server initiates an ISCC transaction. It prepares to send the call to another T-Server and coordinates the process.
- The destination T-Server receives call data from an origination T-Server and matches this data to a call that will arrive at some time in the future.

The distinction between these roles is important because the range of telephony-hardware functionality often requires T-Servers to support two entirely different sets of ISCC transactions based on which of the two roles they play. For instance, it is very common for a particular T-Server to support many types of ISCC transactions when it takes on the origination role, but fewer when it takes on the role of a destination T-Server.

The ISCC transaction type `reroute` is a good example. Most T-Servers support `Reroute` as origination T-Servers, but very few support `Reroute` as destination T-Servers.

Determining and Configuring Transaction Type Support

You can find descriptions of these transaction types starting on page 669. Use Table 119 on page 675 to identify the transaction types your destination T-Server supports. A blank table cell indicates that T-Server does not support a certain transaction type.

You can configure the transaction types specific to your T-Server as values of the `cast-type` configuration option specified in the ISCC configuration section `extrouter`. Refer to Chapter 11, "T-Server Common Configuration Options," on page 737 for the option description.

ISCC Transaction Type General Principles

Generally, since most of the ISCC implementation is done at the T-Server Common Part (TSCP) code level, all T-Servers support certain ISCC transaction types. Any T-Server can act as the origination T-Server for the following transaction types:

- `direct-ani`, page 669
- `direct-notoken`, page 671

- dnis-pool, page 671
- pullback, page 672
- reroute, page 673
- route (aliased as route-notoken), the default transaction type, page 674

The following transaction types are unevenly supported for both the origination and destination T-Server roles:

- direct-callid (aliased as direct), page 669
- direct-digits (reserved for Genesys Engineering)
- direct-network-callid, page 670
- direct-uui, page 670
- route-uui, page 675

The reroute and pullback transaction types are supported only for selected T-Servers in the *destination* role. However, if you implement this support, other transaction types require additional configuration and testing—even those that would normally be supported by default.

direct-ani

With the transaction type direct-ani, the ANI call attribute is taken as the parameter for call matching. Properly configured switches and trunks can keep the ANI attribute when a call is transferred over the network. T-Server can use this network feature for call matching.

Warning! Depending on the switch platform, it may be possible to inherit the ANI attribute after routing a call to a remote destination, and after performing a single-step transfer and other telephone actions. However, ISCC only works properly in scenarios where the ANI attribute on the destination T-Server is represented by exactly the same digit string as on the origination T-Server.

Typically, the ANI attribute represents the original call identifier (customer phone number), which guarantees that the attribute remains unique. However, you can use the non-unique-ani resource type to block ISCC from matching calls based on an ANI that is known to be non-unique. (See "Configuring access resources for non-unique ANI" on page 711 for details.)

direct-callid

With the transaction type direct-callid, the call reaches the destination DN directly from another location, and the CallID of the call is taken as the attribute for call matching. When a call arrives at the final destination, the destination T-Server identifies its CallID, and updates the call info if the CallID matches.

Use this transaction type when the destination switch has the capability to assign to an incoming call the same network-wide unique `CallID` that the origination switch has already assigned to that call.

Notes: The `direct-callid` transaction type is used only in conjunction with the `TRouteCall` and `TSingleStepTransfer` function calls. It is applied only to the call that is in progress, and does not apply to functions that involve in the creation of a new call, such as `TMakeCall`.

For T-Server for Nortel Communication Server 2000/2100, the `direct-callid` transaction type is also applied to the `TMuteTransfer` function.

direct-network-callid

With the transaction type `direct-network-callid`, the call reaches the destination DN directly from another location, and the `NetworkCallID` of the call is taken as the attribute for call matching. When a call arrives at the final destination, the destination T-Server identifies its `NetworkCallID`, and updates the call info if the `NetworkCallID` matches.

Use this transaction type when the destination switch has the capability to assign to an incoming call the same network-wide unique `NetworkCallID` that the origination switch has already assigned to that call.

Note: To support this transaction type, you must configure `Target Type` and `ISCC Protocol Parameters` fields of the corresponding `Switch Access Code` in the Configuration Layer.

direct-uui

With the transaction type `direct-uui`, so-called user-to-user information (UUI) is taken as the attribute for call matching. Some switches make it possible to send a small data packet along with a call. T-Server can use this data to recognize a call passed from one switch to another. The destination T-Server generates a local unique value for `UUI`, and then notifies the origination T-Server. The origination T-Server uses a provided value to mark the call coming from the origination location. The destination T-Server receives a call and checks whether it is marked with an exact `UUI` value. If so, the call is considered to be matched.

On the Avaya Communication Manager and the Aspect ACD, UUI is referred to as “user-to-user information.” On the Siemens Hicom 300 switch with CallBridge, UUI is referred to as “Private User Data.” On the Alcatel A4400/OXE switch, UUI is referred to as “correlator data.”

Note: To support this transaction type, you must configure your switches to pass the UUI provided by your T-Server. You must also ensure that the trunks involved do not drop this data.

direct-notoken

With the transaction type `direct-notoken`, T-Server expects a call to arrive from another location to the destination DN specified in the request for routing service availability. When a call reaches the specified DN, T-Server processes the call as the expected externally-routed call.

Notes: This matching criterion is weak because any call that reaches the specified DN is considered to be the expected call. Genesys recommends that you use this transaction type only in a contact center subdivision that can only be reached from within the contact center (such as the second line of support, which customers cannot contact directly).

When using `direct` transaction types, Network T-Servers and load-sharing IVR Servers are not meant to act as destination T-Servers for call routing. Using Network T-Server with these transaction types requires special architecture.

dnis-pool

With the `dnis-pool` transaction type, T-Server reserves one of its DNIS access resources and waits for the call that has the same `DNIS` attribute as the name of the reserved DNIS access resource.

If the arrived call is matched successfully, the destination T-Server may update the value of the `DNIS` attribute of the call (along with `ConnID`, `UserData`, `CallType`, and `CallHistory`) with the value of the `DNIS` attribute of the original call. This occurs when the value of the `DNIS` attribute of the original call is specified as a value of the key-value pair `_ISCC_TRACKING_NUMBER_` in the `Extensions` attribute of the original client request.

The DNIS matching can be based on any number of digits out of all the digits that comprise the `DNIS` attribute. The number of digits that T-Server should use for DNIS matching is specified for the destination switch as the `ISCC Protocol Parameters` property of the Switch Access Code. The value syntax should be as follows:

`dnis-tail=<number-of-digits>`

For example, if this property is set to the `dnis-tail=7` value, ISCC matches only the last seven digits of a `DNIS`.

You must configure DNIS access resources in the switch; otherwise, ISCC fails to use this transaction type and sends `EventError` in response to the client application request.

Note: The `dnis-pool` transaction type is typically used for networks that employ a “behind the SCP” architecture, such as network IVR. Network T-Server for GenSpec and IServer are two examples of this, but other Network T-Servers might also be used in this architecture.

In Load-Balancing Mode

When T-Server uses load balancing for call routing with the `dnis-pool` transaction type, the following processes occur:

1. A client of the origination T-Server sends a request to pass a call to the location with a DNIS access resource specified in the key-value pair `iscc-selected-dnis`.
2. The origination T-Server distributes the request for a routing service to all destination T-Servers.
3. The destination T-Servers receive the request and check that the specified DNIS is not being used by another routing service request.
4. The origination T-Server expects to receive a positive response from each destination T-Server. If the origination T-Server receives a negative response from at least one T-Server, it sends an `EventError` to the client and clears all data about the request. If the origination T-Server receives the confirmation about routing service availability from all destination T-Servers, it processes the client's request and sends a corresponding message to the switch.
5. The origination switch processes the T-Server request and passes the call to the destination switch.
6. The call arrives at the destination switch, which generates an alerting event to one of the corresponding load-balanced destination T-Servers.
7. That destination T-Server processes the call and notifies the origination T-Server that the routing service was successful and deletes all information about the request.
8. The origination T-Server sends a routing service request cancellation to all other destination T-Servers.
9. The origination T-Server notifies the client that the routing service has been successful and deletes all information about the request.

pullback

`Pullback` is used in the following scenario, for those T-Servers that support it:

1. A call arrives at Site A served by a Network T-Server.
2. At Site A, a Network T-Server client requests to pass the call by means of ISCC routing to Site B served by a premise T-Server. Any transaction type except `reroute` or `pullback` can be specified in this request.
3. The call arrives at Site B and is either answered by an agent or delivered to a routing point.
4. A client of the premise T-Server at Site B sends a `TRouteCall` or `TSingleStepTransfer` request to transfer the call to the network.
5. The Site B premise T-Server notifies the Network T-Server about this request.

6. The Network T-Server receives the notification and issues an `EventRouteRequest` to obtain a new destination.
7. After receiving the new destination information, the Network T-Server disconnects the call from its current premise location at Site B and attempts to route the call to the new destination.
8. The Site B premise T-Server stops tracking the call, which has disconnected from the premise's agent DN or routing point and is delivered to the network.
9. The Network T-Server completes routing the call to its new destination.

Note: The transaction type `pullback` can only be used between SIP Servers or to return a call from a premise T-Server to the Network T-Server that serves the site from which the call was previously transferred.

reroute

`Reroute` is used in the following scenario, for those T-Servers that support it:

1. A call arrives at Site A served by a Network T-Server.
2. At Site A, a Network T-Server client requests to pass the call by means of ISCC to Site B served by a premise T-Server. Any transaction type except `reroute` or `pullback` can be specified in this request.
3. An agent at Site B answers the call.
4. A client of the premise T-Server at Site B sends a `TSingleStepTransfer` or `TRouteCall` request to transfer the call elsewhere (to a PSTN, to an agent, or to a routing point).
5. The Site B premise T-Server notifies the Network T-Server about this request and releases the call leg that resides at the agent's phone (using `TReleaseCall`) or at the Routing Point (using `TRouteCall` with the parameter `RouteTypeCallDisconnect`).
6. The Network T-Server receives the notification and reroutes the call to the requested destination by sending `EventRouteRequest` and attaching the call's user data.

Notes: The transaction type `reroute` can only be used to return a call from a premise T-Server to the Network T-Server that serves the site from which the call was previously transferred.

To perform multi-site operations that are initiated with `TRouteCall` and for which the `reroute` transaction type is requested, the origination T-Server must support the `RouteTypeCallDisconnect` subtype of `TRouteCall`.

route

With the transaction type `route` (aliased as `route-notoken`), a call from the origination location reaches a dedicated External Routing Point, and from there, it is routed to a destination DN.

To control configured External Routing Points, T-Server must register these DNs with the switch. Failure to register implies that the External Routing Point is not available for ISCC purposes. Client applications can register External Routing Points via T-Server for monitoring purposes only.

Point-to-Point (One-to-One)

In the Point-to-Point access mode, only one trunk line is used to access an External Routing Point (for example, VDN, CDN) at the destination site. See Figure 43.

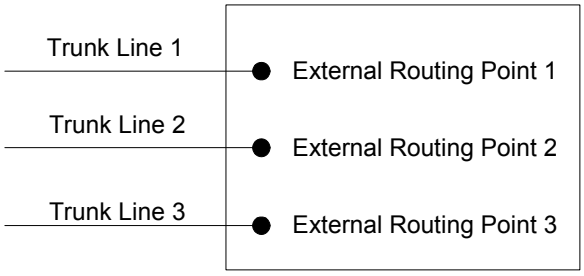

Figure 43: Point-to-Point Trunk Configuration

Note: Dedicated DNs of the `External Routing Point` type must be configured in a switch. See “Configuring Multi-Site Support” on page 700.

Multiple-to-Point (Multiple-to-One)

In the Multiple-to-Point access mode, trunk lines are assigned to the destination switch’s trunk group, from which calls are routed to the final destination. See Figure 44.

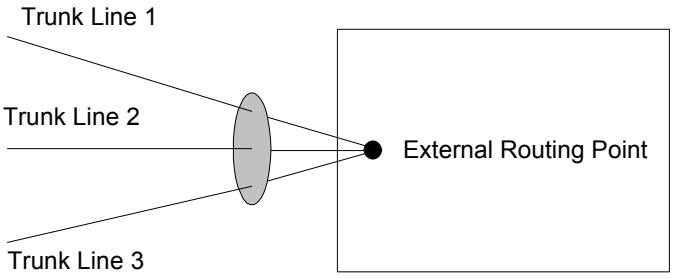

Figure 44: Multiple-to-Point Trunk Configuration

With this configuration, all calls reach the same External Routing Point. The `DNIS` attribute of a specific call differs from that of other calls and uniquely identifies the trunk from which the call arrived.

Note: To switch to this operating mode, you must configure the `route-dn` configuration option for T-Server.

route-uui

The `route-uui` transaction type employs the dedicated External Routing Point feature of the `route` transaction type (page 674) and the UUI matching feature of the `direct-uui` transaction type (page 670). This transaction type accommodates those switches that require a designated External Routing Point even though they use UUI for tracking.

Note: To support this transaction type, you must configure your switches to pass the UUI provided by your T-Server. You must also ensure that the trunks involved do not drop this data.

T-Server Transaction Type Support

Table 119 shows which transaction types are supported by a specific T-Server. Use this table to determine the transaction types that are available for use with your T-Server. This applies both to the `cast-type` you specify in the configuration options for your T-Server, and to any client-designated route-type requests specified for transfers of calls. A blank table cell indicates that T-Server does not support a certain transaction type.

Table 119: T-Server Support of Transaction Types

T-Server Type	Transaction Type										
	route		re-route	direct-callid	direct-uui / route-uui	direct-no-token	direct-ani	direct-digits	direct-network-callid	dnis-pool	pull-back
	one-to-one	multiple-to-one									
Alcatel A4400/OXE	Yes			Yes[a,b,c]	Yes[d]	Yes	Yes[a]		Yes[e]		
Avaya Communication Manager	Yes				Yes	Yes	Yes				
Avaya TSAPI	Yes				Yes	Yes	Yes				

Table 119: T-Server Support of Transaction Types (Continued)

T-Server Type	Transaction Type										
	route		re-route	direct-callid	direct-uui / route-uui	direct-no-token	direct-ani	direct-digits	direct-network-callid	dnis-pool	pull-back
	one-to-one	multiple-to-one									
Cisco Unified Communications Manager	Yes			Yes		Yes	Yes				
Mitel MiTAI	Yes					Yes	Yes		Yes		
NEC NEAX/APEX	Yes			Yes		Yes	Yes				
Siemens HiPath 4000 CSTA III	Yes				Yes[d]	Yes	Yes				
SIP Server	Yes		Yes		Yes[f]	Yes					Yes[g]

a. Not supported in the case of function `TRouteCall` on a Virtual Routing Point: a Routing Point can be simulated using a hunt group with calls being deflected or transferred from the hunt-group member when routing. When a two-step (typically mute) transfer is used on such a hunt-group member, `CallID` and `ANI` usually change; thus, the `direct-callid` and `direct-ani` types do not work.

b. Not supported in the case of function `TSingleStepTransfer` when the T-Server service is simulated using a two-step transfer to the switch. In this case, `CallID` and `ANI` change; thus, the `direct-callid` and `direct-ani` types do not work.

c. Not supported if two T-Servers are connected to different nodes.

d. There are some switch-specific limitations when assigning CSTA correlator data UUI to a call.

e. Supported only on ABCF trunks (Alcatel internal network).

f. SIP Server supports the `direct-uui` type.

g. If the `pullback` transaction does not explicitly include the destination DN, SIP Server supports the transfer only to a DN contained in the `FirstTransferOriginationLocationDN` attribute. If this attribute is empty (for example, if the first transfer transaction was ISCC/Call Overflow (COF), SIP Server fails the `pullback` transaction.

Transfer Connect Service Feature

The Transfer Connect Service (TCS) feature supports transfer connect services available on some telephony networks. When this feature is enabled, ISCC passes user data to remote locations to which calls are transferred or conferenced using transfer connect services.

Procedure:
Activating Transfer Connect Service

Start of procedure

1. In the T-Server `Application` > `Application Options` tab:
   a. Set the `tcs-use` configuration option to `always`.
   b. Set the `tcs-queue` configuration option to the number of a DN on the origination switch.

      ISCC uses this DN as an intermediate step when sending calls to the remote location. The DN that is configured as `tcs-queue` receives attached data indicating the Feature Access Code (FAC) needed to reach the remote site. After a call is directed to the DN with data, a monitoring application takes the data and generates the required DTMF (dual-tone multifrequency) tones to redirect the call through the network to the remote location.

End of procedure

Note: With T-Server for Avaya Communication Manager, you can use `RequestRouteCall` with `RouteTypeOverwriteDNIS` to initiate the playing of DTMF tones. This is done through the use of another intermediate DN (typically, an announcement port configured to give the silent treatment), to which the call is routed. When the call is established on this DN, T-Server requests that the digits sent in the `DNIS` field of the `TRequestRouteCall` be played by using the `ASAI-send-DTMF-single` procedure.

ISCC/Call Overflow Feature

The Inter Server Call Control/Call Overflow (ISCC/COF) feature of T-Server, that supports *passive external routing*, is specifically designed to handle calls delivered between sites without an explicitly defined destination location. Such scenarios include contact center overflows and manual call transfers.

An *overflow situation* occurs when a call comes into a contact center where all agents are currently busy. In this situation, the switch can transfer (overflow) the incoming call to another site where there is an available agent.

T-Server uses two methods to handle call overflow and manual transfer scenarios. The first method is based on `NetworkCallID` matching and the second method is based on `ANI/OtherDN` matching.

When connected to each other via switch-specific networks, switches of some types can pass additional information along with transferred calls. This

information may contain the NetworkCallID of a call, which is a network-wide unique identifier of the call.

When connected via a regular PSTN, switches of all types can send the ANI and/or OtherDN attributes to the destination switch during any call transfer operation.

While all T-Servers support the ISCC/COF feature using the ANI and/or OtherDN attributes, only a few support this feature using the NetworkCallID attribute. Table 120 shows the T-Server types that provide the NetworkCallID of a call.

Table 120: T-Server Support of NetworkCallID for ISCC/COF Feature

T-Server Type	Supported NetworkCallID Attribute
Alcatel A4400/OXE[a]	Yes
Avaya Communication Manager[a,b]	Yes
Avaya TSAPI[a,b]	Yes
Mitel MiTAI[a]	Yes
Nortel Communication Server 2000/2100[a]	Yes
SIP Server[a]	Yes

a. NetworkCallID is supported only if the match-flexible configuration parameter is used.

b. ISCC/COF is cross-compatible between T-Server for Avaya Communication Manager and T-Server for Avaya TSAPI.

Note: SIP Server supports ANI matching in ISCC COF scenarios.

The ISCC/COF feature can use any of the three attributes (NetworkCallID, ANI, or OtherDN) as criteria for matching the arriving call with an existing call at another location. Consequently, the attribute that is used determines what

`ConnID`, `UserData`, `CallType`, and `CallHistory` are received for the matched call from the call's previous location.

Warning! Depending on the switch platform, it may be possible to inherit the `ANI` attribute after routing a call to a remote destination, and after performing a single-step transfer and other telephone actions. However, ISCC/COF works properly only in scenarios where the `ANI` attribute on the destination T-Server is represented by exactly the same unique digit string as on the origination T-Server.

Typically, the `ANI` attribute represents the original call identifier (customer phone number), which guarantees that the attribute remains unique.

Note: When the ISCC/COF feature is in use, the Number Translation feature becomes active. For more information on feature configuration, see "Number Translation Feature" on page 681.

ISCC/COF Call Flow

Figure 45 shows the sequence of steps that occur in an ISCC/COF scenario when a call is made or transferred by an agent at Site A to a DN at Site B, or when a call is overflowed from Site A to Site B.

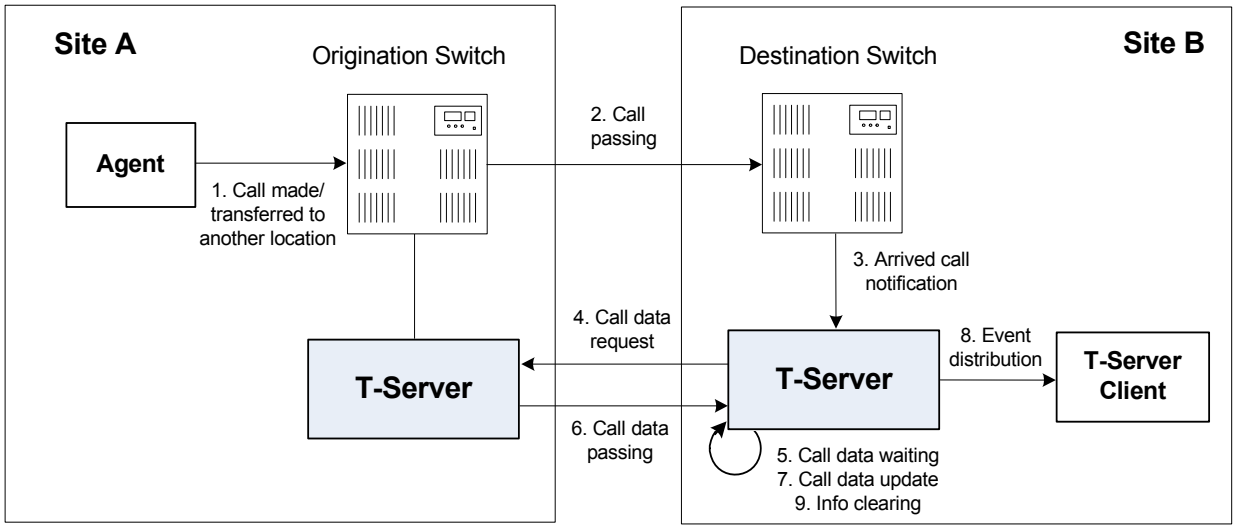

Figure 45: Steps in the ISCC/COF Process

Step 1

An agent makes or transfers a call manually to another location or a call is overflowed from Site A (origination location) to Site B (destination location).

Step 2

Switch A (the origination switch) passes the call to Switch B (the destination switch).

Step 3

The call reaches the destination switch, which notifies the destination T-Server about the arrived call.

Step 4

The destination T-Server verifies with remote locations whether the call overflowed at any of them.

To determine which calls to check as possibly having overflowed, T-Server relies on the `Switch` object and the presence of DNs on the `Switch` configured as the `Access Resource` type with the `Resource Type` set either to `cof-in` (COF-IN DNs) or to `cof-not-in` (COF-NOT-IN DNs):

T-Server skips an arriving call when one of following conditions is met:

- The call arrives at a DN configured as an Enabled COF-NOT-IN DN.
- COF-IN DNs are configured, but the call arrives at a DN other than one of the configured COF-IN DNs or to a COF-IN DN which is Disabled.

In all other cases, the call is checked for overflow.

To determine which location the call arrived from, T-Server checks the call type and checks whether the call has the `NetworkCallID`, `ANI`, or `OtherDN` attribute:

- If the call is not an inbound call, the request for call data is sent to all remote locations *except* those whose Switch Access Code has the ISCC Call Overflow `Parameters` property set to `inbound-only=true`.
- If the call of any type has the `NetworkCallID` attribute, the destination T-Server sends a request for call data to the remote locations of the same switch type as the destination location if their Switch Access Codes have the ISCC Call Overflow `Parameters` property set to `match-callid`.
- If the call of any type has the `ANI` or `OtherDN` attribute, the request for call data is sent to remote locations whose Switch Access Code has the ISCC Call Overflow `Parameters` property set to `match-ani`.

Step 5

The destination T-Server waits (suspending events related to that call) for the call data from the remote T-Server for the time interval specified in the `cof-ci-req-tout` configuration option. Within this interval, T-Server holds any events related to the call. In addition, the `cof-ci-defer-delete` option on the origination T-Server establishes the time interval only after which that T-Server deletes the call information. And the `cof-ci-wait-all`, if set to `true`,

forces the origination T-Server to wait for responses related to possible call overflow situations before updating call data.

Step 6

The T-Server at the location from which the call was transferred or overflowed sends call data to the requesting T-Server.

Step 7

If a positive response to the call-data request is received, T-Server updates `ConnID`, `UserData`, `CallType`, and `CallHistory`, distributes all suspended events related to that call, and deletes all information regarding the transaction (Step 9).

Step 8

If the timeout set by `cof-ci-req-tout` expires, T-Server distributes all suspended events, and starts the timeout specified by the `cof-rci-tout` option. If a positive response is received within the timeout set by `cof-rci-tout`, T-Server updates the `ConnID`, `UserData`, `CallType`, and `CallHistory`, and notifies client applications by distributing `EventPartyChanged`.

Step 9

T-Server deletes all information regarding the transaction when one of these results occurs:

- The first positive response to the call-data request is received.
- Negative responses from all queried locations are received.
- The timeout specified by the `cof-rci-tout` option expires.

Note: For information about configuring the ISCC/Call Overflow feature, see "Configuring Multi-Site Support" on page 700 and Table 123, "Target Type: ISCC Call Overflow Parameters," on page 706.

Number Translation Feature

The Number Translation feature of T-Server extends the ISCC/COF and `direct-ani` transaction type functions to provide more flexibility for handling calls distributed across multiple sites. T-Server translates the input string (ANI string) into a number defined by the translation rules. This processing is called number translation. T-Servers participating in handling calls at multiple sites exchange the translated numbers in order to match the call instances.

The translation process involves two algorithms, one for rule selection and the other for the actual translation. Through the first algorithm, T-Server selects a rule that will be used for number translation. Through the second algorithm,

T-Server translates the number according to the selected rule definition. See "Number Translation Rules" on page 682 for more information on configuring rules for your environment.

Number translation occurs as follows:

1. The switch reports a number, typically via `AttributeANI`.
2. T-Server evaluates all configured inbound rules to determine which one is the best fit for the received number. The best fit is determined by comparing the length of, and the specific digits in, the input number with the inbound pattern of each configured rule. See "Rule Examples" on page 687 for specific examples.
3. T-Server translates the number according to the selected rule.

To enable T-Server to translate numbers, you must perform specific configuration tasks that are associated with translation. See "Configuring Number Translation" on page 689.

Number Translation Rules

T-Server uses the number translation rules that you define in the T-Server configuration object in two ways:

- Rule selection—To determine which rule should be used for number translation
- Number translation—To transform the number according to the selected rule

Using ABNF for Rules

The number translation rules must conform to the following syntax, represented using Augmented Backus-Naur Form (ABNF) notation. For more information about ABNF, see RFC 2234, "Augmented BNF for Syntax Specifications: ABNF."

Note: The following notation explanations begin with the highest level notation. Each explanation includes the name of a component notation and a basic definition of each component that it contains. Some components require more detailed definitions, which are included later in this section.

Common Syntax Notations

Syntax notations common to many of these rules include:

- `*`—Indicates that 0 to an infinite number of the item following this symbol are acceptable.
- `1*`—Indicates that one repetition is required. For T-Server, only one instance is acceptable.

- /—Indicates that any of the items mentioned, or a combination of those items, is acceptable.

Component Notations

Component notations include:

- `dialing-plan = *dialing-plan-rule`

  where:

  - `dialing-plan-rule` represents the name of the rule. Each rule must have a unique name. There are no other naming restrictions, and you do not need to model your names according to the examples in this chapter.

  The rules are represented as separate options in the configuration. Also, fields from a rule are represented as parameters in a single option string.

- `rule = [name] in-pattern [out-pattern]`

  where:

  - `[name]` is the name for the rule option, for example, `rule-01`. In ABNF notation, the brackets `[]` indicate that 0 or 1 instance of the component is required. However, for T-Server, a name is required.
  - `in-pattern` is the part of the rule to which T-Server looks when attempting to match the input number.
  - `[out-pattern]` is the part of the rule that instructs T-Server on how to translate the input number into the required format. The brackets indicate that either 0 or 1 instance is required. You must create an `out-pattern` for number translation rules.

- `name = *( ALPHA / DIGIT / "-")`

  where:

  - `ALPHA` indicates that letters can be used in the name for the rule option.
  - `DIGIT` indicates that numbers can be used in the name for the rule option.
  - `"-"` indicates that a dash (-) can also be used in the option name, for example, `rule-01`.

- `in-pattern = 1*(digit-part / abstract-group)`

  where:

  - `digit-part` represents numbers. T-Server uses this when selecting the most appropriate rule from the entire dialing plan.
  - `abstract-group` represents one or more letters with each letter representing one or more numbers. T-Server uses this when transforming a dial string.

  For example, `[1-9]` is the `digit-part` (representing a range of numbers) and `ABBB` is the `abstract-group` for `in-pattern=[1-9]ABBB`.

- `out-pattern = 1*(symbol-part / group-identifier) *param-part`

  where:

- `symbol-part` represents digits, symbols, or a combination. Symbols are rarely used. They are not used in the United States.
- `group-identifier` are letters that represent groups of numbers. A letter in the `out-pattern` represents one or more digits, based on the number of times the letter is used in the `in-pattern`.
- `*param-part` represents an additional parameter, such as `phone-context`. Reminder: an asterisk means that 0 to an infinite number of these are acceptable.

For example, in `rule-04; in-pattern=1AAABBBCCC;out-pattern=91ABC, 91` is the `symbol-part`; `A, B,` and `C` are `group-identifiers` in the `out-pattern`, each representing three digits, since there are three instances of each in the `in-pattern`.

Note: Prefix an `out-pattern` value with a plus sign (+) for the inbound rule when the output must be in a global form (E.164 format).

- `digit-part = digits / range / sequence`

  where:
  - `digits` are numbers 0 through 9.
  - `range` is a series of digits, for example, 1-3.
  - `sequence` is a set of digits.
- `symbol-part = digits / symbols`

  where:
  - `digits` are numbers 0 through 9.
  - `symbols` include such characters as `+, -,` and so on.
- `range = "[" digits "-" digits "]" group-identifier`

  where:
  - `"[" digits "-" digits "]"` represents the numeric range, for example, `[1-2]`.
  - `group-identifier` represents the group to which the number range is applied.

    For example, `[1-2]` applies to group identifier `A` for `in-pattern=[1-2]ABBB`. When T-Server evaluates the rule to determine if it matches the number, it examines whether the first digit of the number, identified as `group-identifier A`, is `1` or `2`.
- `sequence = "[" 1*(digits [","] ) "]" group-identifier`

  where:
  - `"[" 1*(digits [","] ) "]"` represents a sequence of digits, separated by commas, and bracketed. T-Server requires that each digit set have the same number of digits. For example, in `[415,650]` the sets have three digits.

- `group-identifier` represents the group to which the number sequence is applied.

  For example, in `in-pattern=1[415,650]A*B, [415,650]` applies to `group-identifier A`. When T-Server evaluates the rule to determine if it matches the number, it examines whether the three digits (`group-identifier A`) following the 1 in the number are `415` or `650`.

- `abstract-group = fixed-length-group / flexible-length-group / entity`

  where:

  - `fixed-length-group` specifies a group composed of a specific number of digits and determined by how many times the group identifier is included in the `in-pattern`. For example, for `in-pattern=1AAABBBCCCC`, there are three digits in group `A` and `B` but four in group `C`.

    When you create an `out-pattern`, you include the group identifier only once because the `in-pattern` tells T-Server how many digits belong in that group. For example, `rule-04` (see page 687) is `in-pattern=1AAABBBCCCC; out-pattern=91ABC`.
  - `flexible-length-group` specifies a group composed of 0 or more digits in the group represented by the `group-identifier`. For example, in `in-pattern=1[415,650]A*B, *B` represents the flexible length group containing the remaining digits in the number.
  - `entity` represents digits defined for a specific purpose, for example, country code.

  The component `abstract-group` is used only for the `in-pattern`.

- `fixed-length-group = 1*group-identifier`

  See the earlier explanation under `abstract-group`.

- `flexible-length-group = "*" group-identifier`

  See the earlier explanation under `abstract-group`.

- `entity = "#" entity-identifier group-identifier`

  where:

  - "`#`" indicates the start of a Country Code `entity-identifier`.
  - `entity-identifier` must be the letter `C` which represents Country Code when preceded by a pound symbol (`#`). Any other letter following the # causes an error.
  - `group-identifier` represents the Country Code group when preceded by `#C`.

  The `entity` component is a special group that assumes some kind of predefined processing, such as the Country Code detection.

- `param-part = ";" param-name "=" param-value`

  where:

  - "`;`" is a required separator element.
  - `param-name` is the name of the parameter.
  - "`=`" is the next required element.

- `param-value` represents the value for `param-name`.
- `param-name = "ext" / "phone-context" / "dn"`

  where:
  - `"ext"` refers to extension.
  - `"phone-context"` represents the value of the `phone-context` option configured on the switch.
  - `"dn"` represents the directory number.
- `param-value = 1*ANYSYMBOL`

  where:
  - `ANYSYMBOL` represents any number, letter, or symbol with no restrictions.
- `group-identifier = ALPHA`
- `entity-identifier = ALPHA`
- `digits = 1*DIGIT`
- `symbols = 1*("-" / "+" / ")" / "("/ ".")`

Recommendations for Rule Configuration

The configuration of rules for inbound numbers usually depends on the settings in the corresponding PBX. These settings often define the form in which the PBX notifies its client applications about the number from which an inbound call is coming.

As a general guideline, configure rules that define how to process calls from:

- Internal numbers.
- External numbers within the same local dialing area.
- External numbers within the same country.
- International numbers.

Rules for inbound numbers, typically for North American locations, might look like this:

1. Two rules to transform internal numbers (extensions):

   ```
   name=rule-01;in-pattern=[1-9]ABBB;out-pattern=AB
   name=rule-02;in-pattern=[1-9]ABBBB;out-pattern=AB
   ```
2. A rule to transform local area code numbers (in `333-1234` format in this example):

   ```
   name=rule-03;in-pattern=[1-9]ABBBBBB;out-pattern=+1222AB
   ```
3. A rule to transform U.S. numbers (in `+1(222)333-4444` format):

   ```
   name=rule-04;in-pattern=1AAAAAAAAAA;out-pattern=+1A
   ```
4. A rule to transform U.S. numbers without the `+1` prefix (in `(222)333-4444` format):

   ```
   name=rule-05;in-pattern=[2-9]ABBBBBBBBB;out-pattern=+1AB
   ```

5. A rule to transform U.S. numbers with an outside prefix (in `9 +1(222)333-4444` format):

   ```
   name=rule-06;in-pattern=91AAAAAAAAAA;out-pattern=+1A
   ```

6. A rule to transform international numbers with an IDD (international dialing digits) prefix (in `011 +44(111)222-3333` format):

   ```
   name=rule-07;in-pattern=011*A;out-pattern=+A
   ```

7. A rule to transform international numbers without an IDD prefix (in `+44(111)222-3333` format):

   ```
   name=rule-08;in-pattern=[2-9]A*B;out-pattern=+AB
   ```

Rule Examples

This section provides examples of six rules that are configured as options in the Genesys Configuration Database. It also provides examples of how T-Server applies rules to various input numbers.

Rules

rule-01 `in-pattern=[1-8]ABBB;out-pattern=AB`

rule-02 `in-pattern=AAAA;out-pattern=A`

rule-03 `in-pattern=1[415,650]A*B;out-pattern=B`

rule-04 `in-pattern=1AAABBBCCCC;out-pattern=91ABC`

rule-05 `in-pattern=*A913BBBB;out-pattern=80407913B`

rule-06 `in-pattern=011#CA*B;out-pattern=9011AB`

Examples

Here are examples of how T-Server applies configured above rules to various input numbers.

Example 1 T-Server receives input number `2326`.

As a result of the rule selection process, T-Server determines that the matching rule is `rule-01`:

```
name=rule-01;in-pattern=[1-8]ABBB;out-pattern=AB
```

The matching count for this rule is `1`, because Group `A` matches the digit `2`.

As a result of the parsing process, T-Server detects two groups: Group `A = 2` and Group `B = 326`.

T-Server formats the output string as `2326`.

Example 2 T-Server receives input number `9122`.

As a result of the rule selection process, T-Server determines that the matching rule is `rule-02`:

```
name=rule-02;in-pattern=AAAA;out-pattern=A
```

The matching count for this rule is `0`; however, the overall length of the input number matches that of the `in-pattern` configuration.

As a result of the parsing process, T-Server detects one group: Group `A = 9122`.

T-Server formats the output string as `9122`.

Example 3 T-Server receives input number `16503222332`.

As a result of the rule selection process, T-Server determines that the matching rule is `rule-03`:

```
name=rule-03;in-pattern=1[415,650]A*B;out-pattern=B
```

The matching count for this rule is `4`, because the first digit matches and all three digits in Group `A` match.

As a result of the parsing process, T-Server detects two groups: Group `A = 650` and Group `B = 3222332`.

T-Server formats the output string as `3222332`.

Example 4 T-Server receives input number `19253227676`.

As a result of the rule selection process, T-Server determines that the matching rule is `rule-04`:

```
name=rule-04;in-pattern=1AAABBBCCCC;out-pattern=91ABC
```

The matching count for this rule is `1`, because the first digit matches.

As a result of parsing process, T-Server detects three groups: Group `A = 925`, Group `B = 322`, and Group `C = 7676`.

T-Server formats the output string as `919253227676`.

Example 5 T-Server receives input number `4089137676`.

As a result of rule selection process, T-Server determines that the matching rule is `rule-05`:

```
name=rule-05;in-pattern=*A913BBBB;out-pattern=80407913B
```

The matching count for this rule is `3`, because three digits match.

As a result of the parsing process, T-Server detects two groups: Group `A = 408` and Group `B = 7676`.

T-Server formats the output string as `804079137676`.

Example 6 T-Server receives input number `011441112223333`.

As a result of the rule selection process, T-Server determines that the matching rule is `rule-06`:

```
name=rule-06;in-pattern=011#CA*B;out-pattern=9011AB
```

The matching count for this rule is `3`, because three digits match.

As a result of the parsing process, T-Server detects two groups: Group `A = 44` and Group `B = 1112223333`.

T-Server formats the output string as `9011441112223333`.

Procedure:
Configuring Number Translation

Purpose: To configure the Number Translation feature in T-Server to provide more flexibility for handling calls distributed across multiple sites.

Overview

- The Number Translation feature becomes active when the ISCC/COF feature and/or the `direct-ani` transaction type are used.
- This configuration procedure must be completed within the T-Server `Application` object corresponding to your T-Server.

Start of procedure

1. Open the T-Server `Application's` Properties dialog box.
2. Click the `Options` tab.
3. Create a new section called `extrouter` or open an existing section with this name.
4. Create a new option called `inbound-translator-<n>`. This option points to another section that describes the translation rules for inbound numbers.
5. In this section, create one configuration option for each rule. Specify the rule name as the option name. The values of these options are the rules for the number translation.

   For the option description and its valid values, see Chapter 11, "T-Server Common Configuration Options," on page 737.
6. When you are finished, click `Apply`.
7. Click `OK` to save your changes and exit the `Properties` dialog box.

End of procedure

Network Attended Transfer/Conference Feature

The Network Attended Transfer/Conference (NAT/C) feature is designed to enable agents working in multi-site contact centers to consult with each other before making call transfers or conferences, regardless of whether both agents work at the same or different sites. It also enables the agent who requests a consultation to maintain his or her conversation with the customer while the system is looking for an available agent and setting up the consultation call.

The NAT/C feature does not rely on the call transfer capabilities of the local switch.

There are two modes in which the network attended transfer/conference can be performed: *direct* and *URS-controlled*. Figure 46 shows the sequence of steps that occur in *URS-controlled* mode, when Agent A, who is handling a customer call, requests a consultation with another agent, and URS (Universal Routing Server) selects Agent B, who is working at another site. The *direct* mode is similar to the *URS-controlled* mode, with the difference that URS is not involved in the process (Step 2 and Step 3 are omitted).

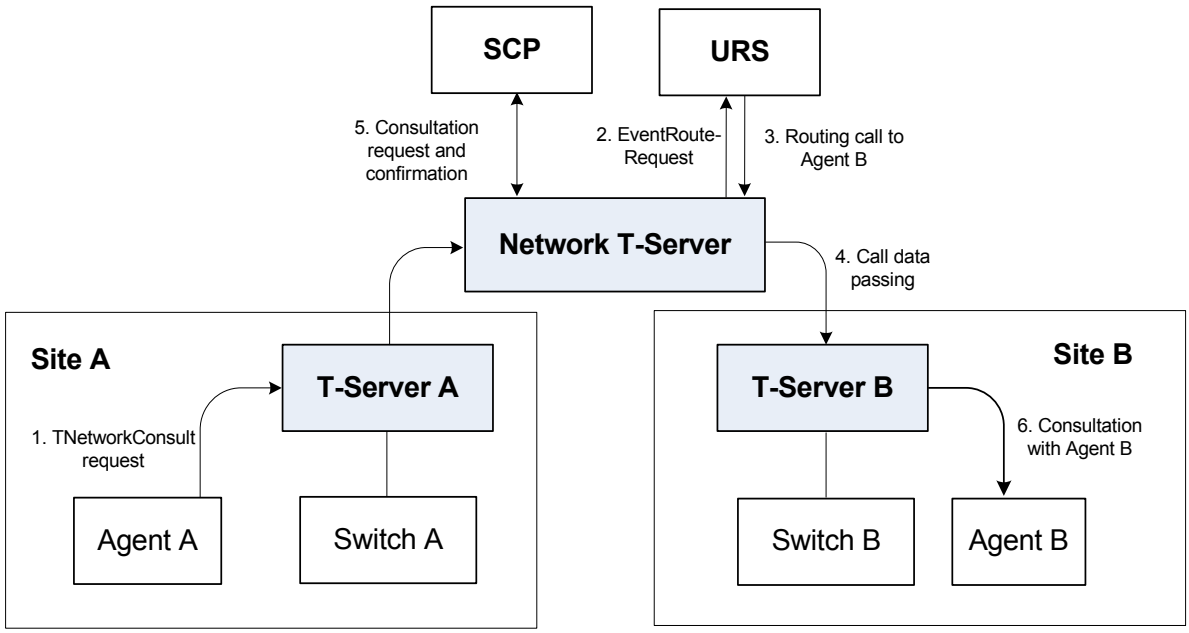

Figure 46: Steps in the NAT/C Process in URS-Controlled Mode

Step 1

Agent A makes a request for a consultation with another agent. A `TNetworkConsult` request is relayed to the Network T-Server. Depending on the parameter settings of the `TNetworkConsult` request, the NAT/C feature will operate in either *direct* or *URS-controlled* mode. For more information, see the *Platform SDK 8.x .NET (or Java) API Reference.*

Step 2

(*URS-controlled* mode only.) The Network T-Server sends `EventRouteRequest` to URS.

Step 3

(*URS-controlled* mode only.) URS locates an available agent at Site B and instructs the Network T-Server to route the call to Agent B. The Network T-Server confirms the initiation of the network transfer by sending `EventNetworkCallStatus` to T-Server A, which then relays it to Agent A.

Step 4

The Network T-Server proceeds to obtain the access number from T-Server B, and passes the call data to T-Server B. (See "ISCC Call Data Transfer Service" on page 661 for details.)

Step 5

The Network T-Server instructs the Service Control Point (SCP) to initiate a new voice path with Agent B. Once the connection is confirmed, the Network T-Server distributes `EventNetworkCallStatus` to both T-Server A and T-Server B, which then relay it to Agent A and Agent B respectively, to indicate that the consultation call is being established.

The Network T-Server also distributes `EventRouteUsed` to URS to confirm successful routing of the call to the selected agent.

Step 6

At this point, the customer is on hold, and Agent A is consulting with Agent B. Agent A can do one of the following:

- End the consultation and retrieve the original customer call
- Alternate between Agent B and the customer
- Set up a conference call with Agent B and the customer
- Transfer the customer call to Agent B

Note: All T-Servers support NAT/C requests with `AttributeHomeLocation` provided that this attribute identifies a network location that is capable of processing such requests. Refer to the *Network T-Server Deployment Guides* to determine whether a specific Network T-Server can process these requests.

Event Propagation Feature

The Event Propagation feature complements the ISCC and ISCC/COF features by distributing updated user data and party-related events to remote T-Servers. This feature is used when a call is being made, transferred, or conferenced to another location, and when, as a result, one or more instances of the call reside at one location while other call instances reside at another location. In this scenario, when a client at one location makes changes to user data, updated user data is passed *(propagated)* to T-Servers at other locations.

The Event Propagation feature consists of User Data update propagation and Party Events propagation.

User Data Propagation

User data propagation takes place when a client at one location makes changes to user data associated with a call that was made, transferred, conferenced, or routed to other locations. The remote clients involved with the call are notified about the changes with `EventAttachedDataChanged`.

When T-Server receives a local update to user data (that is, when a client of this T-Server has changed the call's user data), T-Server determines if parties at remote locations are involved with the call and, if so, sends (propagates) the updated user data to the T-Servers at remote locations.

When T-Server receives a remote update to user data (that is, when a client of a remote T-Server has changed the call's user data and the remote T-Server has used the Event Propagation feature to send the updated user data), T-Server:

1. Updates the user data of the corresponding local call.
2. Determines if parties at other remote locations are involved with the call and, if so, propagates the updated user data to T-Servers at other remote locations.

The locations to which user data is propagated are selected based on a call distribution topology. That is, the updated user data is passed directly to the location to which a call was sent and to the location from which the call was received, excluding the location from which the update was received.

For example, consider a call made from location A to location B, and then conferenced from location B to location C. The three instances of the call reside at different locations: the first instance is at location A, the second instance is at location B, and the third instance is at location C. The Event Propagation feature is employed in the following scenarios:

- When T-Server at location A receives a local update to user data, it notifies T-Server at location B (to which it sent the call) about changes to the call's user data. Thus, T-Server at location B receives a remote update to user data and, in turn, notifies T-Server at location C (to which it sent the call) about these changes.

  Although T-Server at location C receives a remote update to user data, it does not pass the notification to any other T-Servers, because it did not send the call to any other locations. As mentioned earlier, T-Servers at locations B and C update the user data of the corresponding local calls and notify their clients about the changes with `EventAttachedDataChanged`.
- When T-Server at location B receives a local update to user data, it notifies T-Server at location C (to which it sent the call) and T-Server at location A (from which it received the call) about changes to the call's user data. Thus, T-Servers at locations C and A receive a remote update to user data.

  Because T-Server at location C did not send the call to any other locations, and T-Server at location A originated the call, neither of these T-Servers passes the notification to any other T-Servers. T-Servers at locations C and

A update the user data of the corresponding local calls and notify their clients about the changes with `EventAttachedDataChanged`.

- When T-Server at location C receives a local update to user data, it notifies T-Server at location B (from which it received the call) about changes to the call's user data. Thus, T-Server at location B receives a remote update to user data and, in turn, notifies T-Server at location A (from which it received the call) about these changes.

  Although T-Server at location A receives a remote update to user data, it does not pass the notification to any other T-Servers, because it originated the call. T-Servers at locations B and A update the user data of the corresponding local calls and notify their clients about the changes with `EventAttachedDataChanged`.

When a call is distributed between location A and location C using location B, and is then deleted on location B, propagation between locations A and C still occurs through the transit node at location B.

Party Events Propagation

Party events propagation takes place when a transfer or a conference is completed for a call that was made to or from one or more remote locations, or when a conference party is removed from the conference.

In these cases, the Event Propagation feature distributes party events, such as `EventPartyChanged`, `EventPartyAdded`, and `EventPartyDeleted`, to remote locations involved with the call, according to appropriate call model scenarios.

For example, consider a call made from DN 1 to DN 2 on location A. A `TInitiateConference` request is then issued for DN 2 to transfer the call to external DN 3 on location B. That transfer is made by means of ISCC routing. When this conference is completed on location A, the Event Propagation feature sends `EventPartyChanged` to location B and distributes this event to involved client applications that are connected to location B and registered for DN 3. After that, if a party of the conference is removed from the conference (for example, a party on DN 2), the Event Propagation feature sends `EventPartyDeleted` to location B and distributes this event to client applications registered for DN 3.

If a call involved in the propagation has no local parties but has two or more remote parties, the party events propagation is processed in the same manner as the propagation of user data updates.

For a complete event flow in such scenarios, refer to the *Genesys Events and Models Reference Manual.*

Switch Partitioning

A multi-site environment with switch partitioning or intelligent trunks can be defined as a configuration of multiple virtual switches (or `Switch` objects) that are defined in GAX under a single `Switching Office` object representing a physical switch. Each `Switch` object has its own instance of a T-Server application. All T-Server applications connect to the switch via the same or different CTI link or a gateway. (See Figure 47.)

When the Event Propagation feature is active, updated user data and party-related events—`EventPartyChanged`, `EventPartyDeleted`, and `EventPartyAdded`—are propagated to T-Servers that are involved in call transactions, such as transfer or conference. However, with switch partitioning, the call instances may reside at one partition or at different partitions.

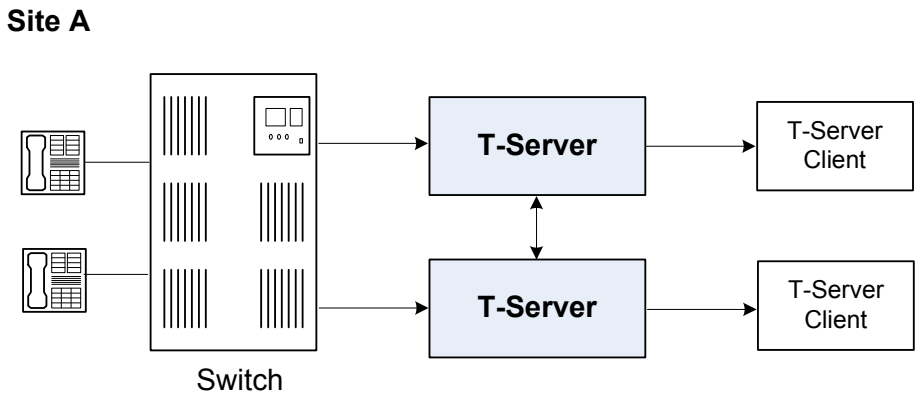

Figure 47: Switch Partitioning Architecture

Starting with version 8.0, in addition to `ConnIDs` and `UserData`, T-Server can synchronize the `CallType` attribute. Each T-Server is required to register all DNs it monitors. In a multi-partitioned environment, when configured, calls between partitions are reported as internal `(CallTypeInternal)`. In a non-partitioned environment, such calls are reported as inbound `(CallTypeInbound)` and/or outbound `(CallTypeOutbound)`, depending on the direction of a call. In order for T-Servers to report calls between specified partitions as internal, registered DNs of these partitions must be assigned to a Switch (T-Server), Switching Office, or Tenant, using the `dn-scope` configuration option. If DNs that are involved in calls are not in the T-Server scope, those DNs will be reported as inbound or outbound.

In addition, T-Server supports `LocalCallType` and `PropagatedCallType` attributes, which depend on the `propagated-call-type` configuration option setting for reporting. See the option description on page 742.

To control race conditions that may occur in the switch-partitioned environment, use the `epp-tout` configuration option (see page 758).

Notes: Because of possible delays in TCP/IP connections, a sequence of events sent for the same call by two or more T-Servers to clients may appear in an unexpected order. For example, in a simple call scenario with two partitions, `EventRinging` and `EventEstablished` messages may both arrive before `EventDialing`.

Genesys switch partitioning does not apply to hardware partitioning functionality that is supported on some switches.

Table 121 shows the T-Server types that support switch partitioning.

Table 121: T-Server Support for Switch Partitioning

T-Server Type	Supported
Alcatel A4400/OXE	Yes
Avaya Communication Manager	Yes
Avaya TSAPI	Yes
Cisco Unified Communications Manager	Yes
SIP Server	Yes

ISCC Path Optimization

The ISCC event propagation distributes events between call parties and is carried through an ISCC connection between SIP Severs. The ISCC signaling path might not precisely coincide with the SIP signaling path.

Figure 48 illustrates a two-server multi-site call topology. The ISCC event propagation triggered by party A is delivered to party C by transiting SIP Server 2.

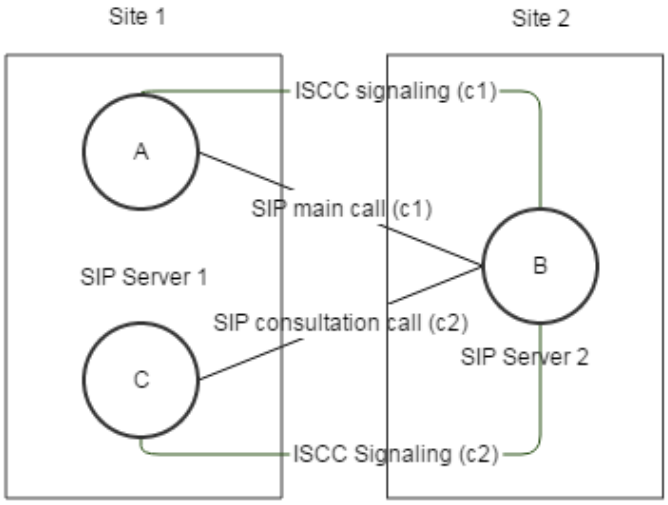

Figure 48: Multi-Site Consultation Call Loop

For trunk optimization, when party B completes a transfer, SIP Server 2 goes out of the SIP signaling path (see "Trunk Optimization for Multi-Site Transfers" on page 376). Even though SIP Server 2 exited the SIP signaling path, without ISCC path optimization, event propagation is still carried through SIP Server 2. (See Figure 49.)

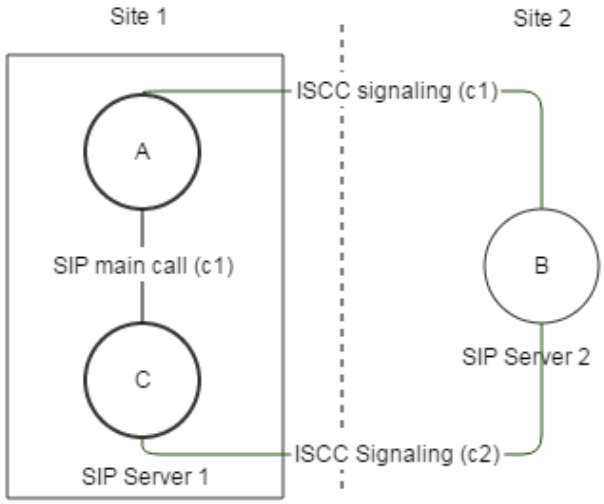

Figure 49: Multi-Site Transfer Without ISCC Path Optimization

The ISCC path optimization feature optimizes the ISCC signaling path by dropping transit sites from the path. Figure 50 illustrates the ISCC path after optimization.

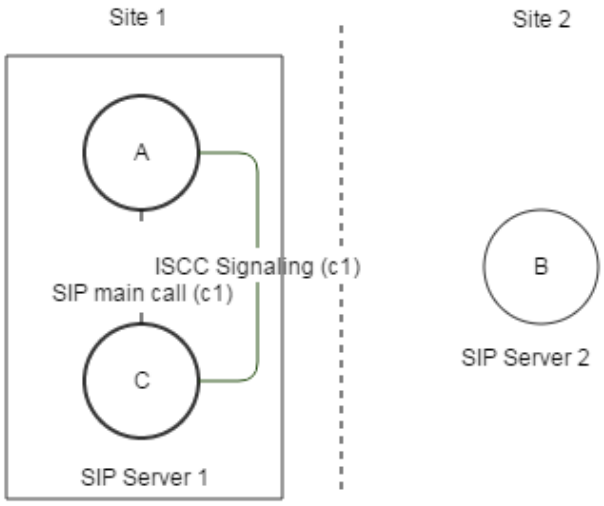

Figure 50: Multi-Site Transfer With ISCC Path Optimization

The ISCC Path Optimization feature is controlled by the `path-optimization` parameter that must be configured in the `ISCC Protocol Parameters` field of the `Switch Access Code`. See Procedure: Configuring Access Codes, on page 704. See also the `ipo-tout` option on page 748.

This feature applies only to two-site configurations.

Note: The ISCC Path Optimization feature is supported only between SIP Servers. It is not currently supported in other Genesys T-Servers.

Event Propagation Configuration

The basic Event Propagation feature configuration includes a setting of specific configuration options at a T-Server `Application` level. The advanced feature configuration allows you to customize the feature at a `Switch` level.

When determining whether to notify other T-Servers of changes to user data, or to distribute party events, T-Server checks:

1. Call topology (what location a call came from and to what location the call was then transferred or conferenced).
2. Outbound parameters of the `Switch` this T-Server relates to (whether propagation parameters are configured for the access codes this switch uses to reach the switch at the location a call came from and the switch at the location to which the call was then transferred or conferenced).

Warning! The direction of user-data or party-events propagation does not necessarily match the direction of call distribution. Therefore, the access code used to deliver the call can differ from the access code used for the purpose of Event Propagation.

If one of the T-Servers along the call distribution path has the Event Propagation feature disabled, that T-Server does not distribute events to remote locations.

Procedure: Activating Event Propagation: basic configuration

Purpose: To activate the Event Propagation feature for User Data updates and call-party–associated events (Party Events) distribution.

Start of procedure

1. In the T-Server `Application > Application Options` tab, click the `extrouter` section.
2. Set the `event-propagation` option to the `list` value.

   This setting enables User Data propagation. If you need to enable Party Events propagation, perform Step 5.
3. Set the `use-data-from` option to the `current` value.

   This setting enables Party Events propagation.

   For the option description and its valid values, see Chapter 11, "T-Server Common Configuration Options," on page 737.

End of procedure

Next Steps

- For advanced feature configuration, do the following procedure:

  Procedure: Modifying Event Propagation: advanced configuration, on page 699

Procedure:
Modifying Event Propagation: advanced configuration

Purpose: To modify access codes for advanced Event Propagation configuration.

Prerequisites

- Procedure: Activating Event Propagation: basic configuration, on page 698

Overview

You can set Event Propagation parameters using:

- The `Default Access Code` properties of the `Switch` that receives an ISCC-routed call (the destination switch).
- The `Access Code` properties of the `Switch` that passes an ISCC-routed call (the origination switch).

If you do not set up Event Propagation parameters for a given `Access Code`, T-Server uses corresponding settings configured for the `Default Access Code` of the destination switch.

The procedures for modifying `Default Access Codes` and `Access Codes` are very similar to each other.

Start of procedure

1. Among configured `Switches`, select the `Switch` that the configured T-Server relates to.
2. Open the `Switch's Properties` dialog box and click either the `Default Access Codes` tab or the `Access Codes` tab.
3. Select a configured `Default Access Code` or configured `Access Code` and click `Edit`.

> **Note:** If no `Default Access Code` is configured, see page 703 for instructions. If no `Access Codes` are configured, see page 704 for instructions.

4. In the `Switch Access Code Properties` dialog box that opens, specify a value for the `ISCC Protocol Parameters` field as follows:
   - To enable distribution of both user data associated with the call and call-party–associated events[2], type:

     ```
     propagate=yes
     ```

2. The following are call-party–associated events: `EventPartyChanged`, `EventPartyDeleted`, and `EventPartyAdded`.

which is the default value.

- To enable distribution of user data associated with the call and disable distribution of call-party–associated events, type:

  `propagate=udata`
- To disable distribution of user data associated with the call and enable distribution of call-party–associated events, type:

  `propagate=party`
- To disable distribution of both user data associated with the call and call-party–associated events, type:

  `propagate=no`
- To enable the ISCC Call Optimization feature, configure this parameter:

  `path-optimization=<cpo, true, false>`

  See Procedure: Configuring Access Codes, on page 704.

End of procedure

ISCC Transaction Monitoring Feature

This feature allows T-Server clients to monitor ISCC transactions that occur during the call data transfer between T-Servers in a multi-site environment.

In order to be able to monitor ISCC messaging, a T-Server client must subscribe to the ISCC Transaction Monitoring. Once a subscription request is confirmed, a client will receive updates about all multi-site operations of this T-Server.

The `TTransactionMonitoring` request is used to instruct T-Server to start, stop, or modify a client's subscription to Transaction Monitoring feature notifications by setting the `TSubscriptionOperationType` parameter to `SubscriptionStart`, `SubscriptionStop`, or `SubscriptionModify` respectively. The transaction status is reported in `EventTransactionStatus` messages to the subscribed clients.

To determine whether the Transaction Monitoring feature is supported by a specific T-Server, a T-Server client may query T-Server's capabilities. For more information about support of this feature, see *Genesys Events and Models Reference Manual* and *Platform SDK 8.x .NET (or Java) API Reference*.

Configuring Multi-Site Support

Prior to configuring T-Server to support multi-site operation, you must read the "Licensing Requirements" on page 56, as well as previous sections of this chapter on multi-site deployment. In particular, Table 119 on page 675 shows which transaction types are supported by a specific T-Server, while Table 120

on page 678 shows whether your T-Server supports the NetworkCallID attribute for the ISCC/COF feature. Use this information as you follow the instructions in this chapter.

Note: Before attempting to configure a multi-site environment, Genesys recommends that you plan the changes you want to make to your existing contact centers. You should then gather the configuration information you will need (such as the name of each T-Server application, port assignments, and switch names), and use GAX to create and partially configure each T-Server object. Review multi-site option values in the "extrouter Section" on page 748 and determine what these values need to be, based on your network topology.

For T-Server to support multi-site operation, you must create and configure three types of objects in the Configuration Layer:

1. Applications
2. Switches, including Access Codes
3. DNs

You must configure these objects for origination and destination locations. Multi-site support features activate automatically at T-Server startup. See "DNs" on page 708 for details.

Applications

Ensure that T-Server Application objects, and their corresponding Host objects, exist and are configured for origination and destination locations.

Once you've done that, use GAX to add this configuration to a T-Server Application.

Procedure: Configuring T-Server Applications

Purpose: To configure T-Server Application objects for multi-site operation support.

Start of procedure

1. In the T-Server Application, click the Connections tab, and click Add to add a connection to the appropriate T-Server. The Connection Info Properties dialog box displays.
2. Use the Browse button to search for the T-Server you want to connect to, and fill in the following values:
   - Port ID

- Connection Protocol
- Local Timeout
- Remote Timeout
- Trace Mode

3. Click the Application Options tab. Create a new section called extrouter or open an existing section with this name.

Note: If you do not create the extrouter section, T-Server uses the default values of the corresponding configuration options.

4. Open the extrouter section. Configure the options used for multi-site support.

Note: For a list of options and valid values, see "extrouter Section" on page 748, in the "T-Server Common Configuration Options" chapter.

5. Repeat this procedure for all T-Servers for origination and destination locations that are used for multi-site operations.

End of procedure

Next Steps

- See "Switches and Access Codes."

Switches and Access Codes

Ensure that Switching Office and Switch objects are configured for both origination and destination locations.

You configure Access Codes to a destination switch in the origination Switch's Properties dialog box. The only exception is the Default Access Code, which is configured at the destination Switch's Properties dialog box.

You can configure two types of switch Access Codes in the Switch's Properties dialog box:

- A Default Access Code (for inbound calls)—Specifies the access code that other switches can use to access this switch when they originate a multi-site transaction.
- An Access Code (for outbound calls)—Specifies the access code that this switch can use when it originates a multi-site transaction to access another switch.

When the origination T-Server processes a multi-site transaction, it looks for an access code to the destination switch. First, T-Server checks the Access Code of the origination Switch:

- If an access code to the destination switch is configured with the target type Target ISCC and with any transaction type except Forbidden, T-Server uses this access code to dial the destination switch.
- If the access code to the destination switch is not configured on the Access Code tab of the origination switch, the origination T-Server checks the Default Access Code tab of the destination switch. If an access code is configured there with the target type Target ISCC and with any transaction type except Forbidden, T-Server uses this access code to dial the destination switch.
- If no access code with the required properties is found, T-Server rejects the transaction.

Note: When migrating from previous releases of T-Servers to 8.1, or when using T-Servers of different releases (including 8.1) in the same environment, see "Compatibility Notes" on page 707.

Procedure: Configuring Default Access Codes

Purpose: To configure the Default Access Codes (one per Switch object) to be used by other switches to access this switch when they originate a multi-site transaction.

Prerequisites

- Ensure that Switching Office and Switch objects are configured for both origination and destination locations.

Start of procedure

1. Among configured Switches, select the Switch that the configured T-Server relates to.
2. Open the Switch Properties dialog box and click the Default Access Codes tab.
3. Click Add to open the Access Code Properties dialog box.

4. In the Code field, specify the access code used by remote switches to reach a DN at this switch. An access code is used as a prefix to the remote switch numbers.

   Note: If no prefix is needed to dial to the configured switch, you can leave the Code field blank.

5. In the Target Type field, select Target ISCC.
6. In the Route Type field, select a value corresponding to the transaction type you want to use (given that it is supported for your switch type).

End of procedure

Next Steps

- See "Configuring Access Codes."

Procedure: Configuring Access Codes

Purpose: To configure the Access Codes (one or more per Switch object) that this switch can use when it originates a multi-site transaction to access another switch.

Prerequisites

- Ensure that Switching Office and Switch objects are configured for both origination and destination locations.

Start of procedure

1. Among configured Switches, select the Switch that the configured T-Server relates to.
2. Open the Switch Properties dialog box and click the Access Codes tab.
3. Click Add to open the Access Code Properties dialog box.
4. In the Switch field, specify the switch that this switch can reach using this access code. Use the Browse button to locate the remote switch.
5. In the Code field, specify the access code used to reach a DN at the remote switch from this switch. An access code is used as a prefix to the remote switch numbers.

   Note: If no prefix is needed to dial from one switch to another, you can leave the Code field blank.

6. In the Target Type field, select Target ISCC.

   When you select Target ISCC as your target type, the Properties dialog box changes its lower pane to the Sources pane. It is here that you enter the extended parameters for your access codes, by specifying the ISCC Protocol and ISCC Call Overflow Parameters.

   To set these parameters, locate the two drop-down boxes that appear below the Target Type field in the Sources pane of that Properties dialog box.

   a. In the ISCC Protocol Parameters drop-down box, enter the appropriate ISCC Protocol parameter, as a comma-separated list of one or more of the following items shown in Table 122:

Table 122: Target Type: ISCC Protocol Parameters

ISCC Protocol Parameters	Description
dnis-tail=<number-of-digits>	Where number-of-digits is the number of significant DNIS digits (last digits) used for call matching. 0 (zero) matches all digits.
propagate=<yes, udata, party, no>	Default is yes. For more information, see "Modifying Event Propagation: advanced configuration" on page 699.
direct-network-callid=<>	Use Table 120 on page 678 to determine if your T-Server supports the direct-network-callid transaction type.
path-optimization=<cpo, true, false>	Default is cpo. When set to false, ISCC does not optimize the Event Propagation path (backward-compatible behavior). When set to cpo, SIP Server aligns the ISCC call path with the SIP signaling path. For example, if SIP Server gets out of the SIP signaling path, so does ISCC. If this parameter is set to true, ISCC makes a decision to optimize the call path based on its internal analysis and without direct relation to the SIP signaling path. **Note:** There can be two Switch Access Codes for any pair of switches. One Switch Access Code is for transfers from switch A to switch B, and the other one for transfers from switch B to switch A. When those Switch Access Codes contain different values of the path-optimization parameter, the effective value is the weaker of the two values. Therefore, if one of them is set to false, the effective value is false; if one of them is set to cpo, the effective value is cpo. Otherwise, the effective value is true. See "ISCC Path Optimization" on page 695 for details.
alternate-route-cof=<true, false>	SIP Server-specific. When set to true, SIP Server uses the ISCC Call Overflow (COF) feature for alternate routing. See "Alternate Routing for Unresponsive URS/ORS" on page 109.

b. In the `ISCC Call Overflow Parameters` drop-down box, enter call overflow parameters, as a comma-separated list of one or more of the following items shown in Table 123:

Table 123: Target Type: ISCC Call Overflow Parameters

ISCC Call Overflow Parameters	Description
`match-callid`	Matches calls using network `CallID`.
`match-ani`	Matches calls using ANI. **Note:** When using `match-ani`, the `match-flexible` parameter must be set to `false`.
`match-flexible`	Supports flexible call matching based on the following values: Default Value: `true` Valid Values: `true`, `false`, and `[matching-context-type]`, where `[matching-context-type]` is the switch-specific value, which must be the same as the value of the `default-network-call-id-matching` configuration option of the corresponding T-Server.
`inbound-only=<boolean>`	Default is `true`. Setting `inbound-only` to `true` disables COF on consultation and outbound calls.

7. In the `Route Type` field, select a value corresponding to the transaction type you want to use (given that it is supported for your switch type). Table 124 contains cross-reference information on transaction types that the Configuration Layer and T-Server use.

Table 124: Route Type and ISCC Transaction Type Cross-Reference

Route Type Field Value	ISCC Transaction Type
Default	The first value from the list of values specified in the `cast-type` option for the T-Server at the destination site
Direct	`direct-callid`
Direct ANI	`direct-ani`
Direct Digits	`direct-digits`
Direct DNIS and ANI	Reserved
Direct Network Call ID	`direct-network-callid`
Direct No Token	`direct-notoken`
Direct UUI	`direct-uui`
DNIS Pooling	`dnis-pooling`
Forbidden	External routing to this destination is not allowed
ISCC defined protocol	Reserved
PullBack	`pullback`
Re-Route	`reroute`
Route	`route`

End of procedure

Next Steps

- After configuring a switch for multi-site support, proceed with the configuration of DNs assigned to this switch.

Compatibility Notes

When migrating from previous releases of T-Servers to 8.1, or when using T-Servers of different releases (including 8.1) in the same environment, keep in mind the following compatibility issues:

- The `Target External Routing Point` value of the `Target Type` field is obsolete and provided only for backward compatibility with T-Servers of releases 5.1 and 6.0. When two access codes for the same switch are

configured, one with the Target ISCC target type and the other with the Target External Routing Point target type, T-Servers of releases 8.x, 7.x, 6.5, and 6.1:

- Use the Target ISCC access code for transactions with T-Servers of releases 8.x, 7.x, 6.5, and 6.1.
- Use the Target External Routing Point access code for transactions with T-Servers of releases 5.1 and 6.0.

When the only access code configured for a switch has the Target External Routing Point target type, T-Server uses this access code for all transactions.

- When the Target External Routing Point value of the Target Type field is configured, you must set the Route Type field to one of the following:
  - Default to enable the route transaction type
  - Label to enable the direct-ani transaction type
  - Direct to enable the direct transaction type

Note: The direct transaction type in releases 5.1 and 6.0 corresponds to the direct-callid transaction type in releases 6.1 and later.

  - UseExtProtocol to enable the direct-uui transaction type
  - PostFeature to enable the reroute transaction type

  These values are fully compatible with the transaction types supported in T-Server release 5.1.
- For successful multi-site operations between any two locations served by release 5.1 T-Servers, identical Route Type values must be set in the Switch's Access Code Properties dialog boxes for both the origination and destination switches.

DNs

Use the procedures from this section to configure access resources for various transaction types.

Procedure: Configuring access resources for the route transaction type

Purpose: To configure dedicated DNs required for the route transaction type.

Prerequisites

- Ensure that Switching Office and Switch objects are configured for both origination and destination locations.

Start of procedure

1. Under a configured Switch, select the DNs folder. From the main menu, select File > New > DN to create a new DN object.
2. On the General tab of the DN's Properties dialog box, specify the number of the configured DN as the value of the Number field. This value must correspond to the Routing Point number on the switch.
3. Select External Routing Point as the value of the Type field.
4. If a dialable number for that Routing Point is different from its DN name, specify the number in the Association field.
5. Click the Access Numbers tab. Click Add and specify these access number parameters:
   - Origination switch.
   - Access number that must be dialed to reach this DN from the origination switch.

   In determining an access number for the Routing Point, T-Server composes it of the values of the following properties (in the order listed):
   a. Access number (if specified).
   b. Switch access code from the switch of the origination party to the switch to which the Routing Point belongs, concatenated with its Association (if the Association value is specified).
   c. Switch access code from the switch of the origination party to the switch to which the Routing Point belongs, concatenated with the number for the DN.
   d. Default access code of the switch to which the Routing Point belongs, concatenated with its Association (if the Association value is specified).
   e. Default access code of the switch to which the Routing Point belongs, concatenated with the number for the DN.

   Note: If option use-implicit-access-numbers is set to true, the access number composed of switch access code and DN can be used for external transfers of calls originating at switches for which an access number is not specified.

6. When you are finished, click Apply.

End of procedure

Procedure: Configuring access resources for the dnis-pool transaction type

Purpose: To configure dedicated DNs required for the `dnis-pool` transaction type.

Start of procedure

1. Under a configured `Switch`, select the `DNs` folder. From the main menu, select `File` > `New` > `DN` to create a new `DN` object.
2. On the `General` tab of the DN's `Properties` dialog box, specify the number of the configured DN as the value of the `Number` field. This value must be a dialable number on the switch.
3. Select `Access Resource` as the `Type` field and type `dnis` as the value of the `Resource Type` field on the `Advanced` tab.
4. Click the `Access Numbers` tab. Click `Add` and specify these `Access Number` parameters:
   - Origination switch.
   - Access number that must be dialed to reach this DN from the origination switch.

   An access number for the access resource is determined in the same manner as for the `route` access resource.
5. When you are finished, click `Apply`.

End of procedure

Procedure: Configuring access resources for direct-* transaction types

Start of procedure

You can use any configured DN as an access resource for the `direct-*` transaction types. (The `*` symbol stands for any of the following: `callid`, `uui`, `notoken`, `ani`, or `digits`.)

You can select the `Use Override` check box on the `Advanced` tab to indicate whether the override value should be used instead of the number value to dial to the DN. You must specify this value if the DN has a different DN name and dialable number. In fact, this value is required for T-Servers for some switch

types—such as Aspect ACD, Nortel Communication Server 2000/2100, and Spectrum.

End of procedure

Procedure: Configuring access resources for ISCC/COF

Purpose: To configure dedicated DNs required for the ISCC/COF feature.

Note: Use Table 120 on page 678 to determine if your T-Server supports the ISCC/COF feature.

Start of procedure

1. Under a configured `Switch`, select the `DNs` folder. From the main menu, select `File > New > DN` to create a new `DN` object.
2. On the `General` tab of the DN `Properties` dialog box, enter the name of the configured DN in the `Number` field.

   Note: The name of a DN of type `Access Resource` must match the name of a DN in your configuration environment (typically, a DN of type `Routing Point` or `ACD Queue`), so T-Server can determine whether the calls arriving at this DN are overflowed calls.

3. Select `Access Resource` as the value for the `Type` field.
4. On the `Advanced` tab, type `cof-in` or `cof-not-in` as the value for the `Resource Type` field.

   Note: Calls coming to DNs with the `cof-not-in` value for the `Resource Type` are never considered to be overflowed.

5. When you are finished, click `Apply`.

End of procedure

Procedure: Configuring access resources for non-unique ANI

Purpose: To configure dedicated DNs required for the `non-unique-ani` resource type.

The non-unique-ani resource type is used to block direct-ani and COF/ani from relaying on ANI when it matches configured/enabled resource digits. Using non-unique-ani, T-Server checks every ANI against a list of non-unique-ani resources.

Start of procedure

1. Under a configured Switch, select the DNs folder. From the main menu, select File > New > DN to create a new DN object.
2. On the General tab of the DN Properties dialog box, specify the ANI digits that need to be excluded from normal processing.
3. Select Access Resource as the value for the Type field.
4. On the Advanced tab, specify the Resource Type field as non-unique-ani.
5. When you are finished, click Apply.

End of procedure

Procedure: Modifying DNs for isolated switch partitioning

Purpose: To modify DNs that belong to a particular partition where switch partitioning is used.

This configuration instructs T-Server to select an External Routing Point that has the same partition as the requested destination DN.

Note: When a target DN is not configured or has no configured partition name, T-Server allocates a DN of the External Routing Point type that belongs to any partition.

Start of procedure

1. Under a Switch object, select the DNs folder.
2. Open the Properties dialog box of a particular DN.
3. Click the Annex tab.
4. Create a new section named TServer.
5. Within that section, create a new option named epn. Set the option value to the partition name to which the DN belongs.
6. Repeat Steps 1–5 for all DNs, including DNs of the External Routing Point type, that belong to the same switch partition.
7. When you are finished, click Apply.

End of procedure

Configuration Examples

This section provides two configuration examples and describes how the configuration settings affect T-Server's behavior.

Multiple Transaction Types

This example demonstrates the difference in how ISCC directs a call when you specify two different transaction types (`route` and `direct-ani`). In this example, you configure an origination and a destination switch for as described in "Switches and Access Codes" on page 702.

1. Among configured `Switches`, select the origination `Switch`.
2. Open the `Switch Properties` dialog box and click the `Default Access Codes` tab.
3. Click `Add` to open the `Access Code Properties` dialog box.
4. Set the `Access Code` field to `9`.
5. When you are finished, click `Apply`.
6. Among configured `Switches`, select the destination `Switch`.
7. Under the destination `Switch`, configure a DN as described in "Configuring access resources for the route transaction type" on page 708.
8. Set the DN `Number` field to `5001234567`.
9. Click the `Advanced` tab of this DN's `Properties` dialog box.
10. Select the `Use Override` check box and enter `1234567` in the `Use Override` field.
11. When you are finished, click `Apply` or `Save`.
12. Use a T-Server client application to register for this new DN with the destination T-Server and, therefore, with the switch.
13. Request to route a call from any DN at the origination switch to the destination DN you have just configured:
    - If you are using the `route` ISCC transaction type, the client requests that T-Server deliver a call to a destination location using the DN number `5001234567`. ISCC requests that the switch dial one of the external routing points at the destination location, using the value either of the `Access Number` field or of the `Access Code` field, which is `9`, concatenated with the external routing point at the destination location. The call is routed to the DN number `5001234567`.
    - If you are using the `direct-ani` ISCC transaction type, the client requests that T-Server deliver a call to a destination location using the DN number `1234567`, which is the `Use Override` value. ISCC requests that the switch dial `91234567`, which is a combination of the `Switch Access Code` value and the `Use Override` value. The destination T-Server is waiting for the call to directly arrive at DN number `5001234567`.

Call Overflow Methods

This section demonstrates how to indicate which overflow methods a switch supports.

In this example, for T-Server to use ANI/OtherDN matching in call overflow and manual transfer scenarios, set the `ISCC Call Overflow Parameters` to:

```
match-ani, inbound-only=true
```

when configuring `Switch Access Codes` as described on page 704.

With this setting, the switch's location is queried for call data each time the destination T-Server receives an inbound call with the `ANI` or `OtherDN` attribute.

For T-Server to use `NetworkCallID` matching in call overflow and manual transfer scenarios, set the `ISCC Call Overflow Parameters` to (for example):

```
match-callid, inbound-only=false
```

when configuring `Switch Access Codes` as described on page 704.

With this setting, the switch's location is queried for call data each time the destination T-Server receives a call of any type (including inbound) with the `NetworkCallID` attribute.

GENESYS

Chapter

10 Common Configuration Options

Unless otherwise noted, the common configuration options that this chapter describes are common to all Genesys server applications and applicable to any Framework server component. This chapter includes the following sections:

Note: Some server applications also support log options that are unique to them. For descriptions of a particular application's unique log options, refer to the chapter/document about that application.

Setting Configuration Options

Unless specified otherwise, set common configuration options, using one of the following navigation paths:

- In Genesys Administrator Extension (GAX) > `Application` object > `Application Options` tab
- (Obsolete) In Genesys Administrator > `Application` object > `Options` tab > `Advanced View (Options)`

- (Obsolete) In Configuration Manager—`Application` object > `Properties` dialog box > `Options` tab

Warning! Configuration section names, configuration option names, and predefined option values are case-sensitive. Type them in GAX exactly as they are documented in this chapter.

Mandatory Options

You do not have to configure any common options to start Server applications.

log Section

This section must be called `log`.

verbose

Default Value: `all`
Valid Values:

`all`	All log events (that is, log events of the Standard, Trace, Interaction, and Debug levels) are generated.
`debug`	The same as `all`.
`trace`	Log events of the Trace level and higher (that is, log events of the Standard, Interaction, and Trace levels) are generated, but log events of the Debug level are not generated.
`interaction`	Log events of the Interaction level and higher (that is, log events of the Standard and Interaction levels) are generated, but log events of the Trace and Debug levels are not generated.
`standard`	Log events of the Standard level are generated, but log events of the Interaction, Trace, and Debug levels are not generated.
`none`	No output is produced.

Changes Take Effect: Immediately

Determines whether a log output is created. If it is, specifies the minimum level of log events generated. The log events levels, starting with the highest priority level, are Standard, Interaction, Trace, and Debug. See also "Log Output Options" on page 722.

Note: For definitions of the Standard, Interaction, Trace, and Debug log levels, refer to the *Framework Management Layer User's Guide* and *Genesys Administrator Extension Help*.

buffering

Default Value: `true`
Valid Values:

`true`	Enables buffering.
`false`	Disables buffering.

Changes Take Effect: Immediately

Turns on/off operating system file buffering. The option is applicable only to the `stderr` and `stdout` output (see page 722). Setting this option to `true` increases the output performance.

Note: When buffering is enabled, there might be a delay before log messages appear at the console.

segment

Default Value: `false`
Valid Values:

`false`	No segmentation is allowed.
`<number> KB` or `<number>`	Sets the maximum segment size, in kilobytes. The minimum segment size is `100 KB`.
`<number> MB`	Sets the maximum segment size, in megabytes.
`<number> hr`	Sets the number of hours for the segment to stay open. The minimum number is `1` hour.

Changes Take Effect: Immediately

Specifies whether there is a segmentation limit for a log file. If there is, sets the mode of measurement, along with the maximum size. If the current log segment exceeds the size set by this option, the file is closed and a new one is created. This option is ignored if log output is not configured to be sent to a log file.

expire

Default Value: `false`
Valid Values:

`false`	No expiration; all generated segments are stored.
`<number> file` or `<number>`	Sets the maximum number of log files to store. Specify a number from `1`–`1000`.
`<number> day`	Sets the maximum number of days before log files are deleted. Specify a number from `1`–`100`.

Changes Take Effect: Immediately

Determines whether log files expire. If they do, sets the measurement for determining when they expire, along with the maximum number of files

(segments) or days before the files are removed. This option is ignored if log output is not configured to be sent to a log file.

Note: If an option's value is set incorrectly—out of the range of valid values— it will be automatically reset to `10`.

keep-startup-file

Default Value: `true`

Valid Values:

`false`	No startup segment of the log is kept.
`true`	A startup segment of the log is kept. The size of the segment equals the value of the `segment` option.
`<number> KB`	Sets the maximum size, in kilobytes, for a startup segment of the log.
`<number> MB`	Sets the maximum size, in megabytes, for a startup segment of the log.

Changes Take Effect: After restart

Specifies whether a startup segment of the log, containing the initial T-Server configuration, is to be kept. If it is, this option can be set to `true` or to a specific size. If set to `true`, the size of the initial segment will be equal to the size of the regular log segment defined by the `segment` option. The value of this option will be ignored if segmentation is turned off (that is, if the `segment` option set to `false`).

Note: In SIP Server multi-threaded logging, the default value of `true` applies only to the T-Server thread log file, and is not reflected in other logs that are generated by running threads.

messagefile

Default Value: As specified by a particular application

Valid Values: `<string>.lms` (message file name)

Changes Take Effect: Immediately, if an application cannot find its `*.lms` file at startup

Specifies the file name for application-specific log events. The name must be valid for the operating system on which the application is running. The option value can also contain the absolute path to the application-specific `*.lms` file. Otherwise, an application looks for the file in its working directory.

Warning! An application that does not find its `*.lms` file at startup cannot generate application-specific log events and send them to Message Server.

message_format

Default Value: `short`
Valid Values:

`short`	An application uses compressed headers when writing log records in its log file.
`full`	An application uses complete headers when writing log records in its log file.

Changes Take Effect: Immediately

Specifies the format of log record headers that an application uses when writing logs in the log file. Using compressed log record headers improves application performance and reduces the log file's size.

With the value set to `short`:

- A header of the log file or the log file segment contains information about the application (such as the application name, application type, host type, and time zone), whereas single log records within the file or segment omit this information.
- A log message priority is abbreviated to `Std`, `Int`, `Trc`, or `Dbg`, for Standard, Interaction, Trace, or Debug messages, respectively.
- The message ID does not contain the prefix `GCTI` or the application type `ID`.

A log record in the full format looks like this:

```
2002-05-07T18:11:38.196 Standard localhost cfg_dbserver GCTI-00-05060
Application started
```

A log record in the short format looks like this:

```
2002-05-07T18:15:33.952 Std 05060 Application started
```

Note: Whether the full or short format is used, time is printed in the format specified by the `time_format` option.

time_convert

Default Value: `Local`
Valid Values:

`local`	The time of log record generation is expressed as a local time, based on the time zone and any seasonal adjustments. Time zone information of the application's host computer is used.
`utc`	The time of log record generation is expressed as Coordinated Universal Time (UTC).

Changes Take Effect: Immediately

Specifies the system in which an application calculates the log record time when generating a log file. The time is converted from the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970).

time_format

Default Value: `time`
Valid Values:

`time`	The time string is formatted according to the `HH:MM:SS.sss` (hours, minutes, seconds, and milliseconds) format.
`locale`	The time string is formatted according to the system's locale.
`ISO8601`	The date in the time string is formatted according to the ISO 8601 format. Fractional seconds are given in milliseconds.

Changes Take Effect: Immediately

Specifies how to represent, in a log file, the time when an application generates log records.

A log record's time field in the ISO 8601 format looks like this:

```
2001-07-24T04:58:10.123
```

print-attributes

Default Value: `false`
Valid Values:

`true`	Attaches extended attributes, if any exist, to a log event sent to log output.
`false`	Does not attach extended attributes to a log event sent to log output.

Changes Take Effect: Immediately

Specifies whether the application attaches extended attributes, if any exist, to a log event that it sends to log output. Typically, log events of the Interaction log level and Audit-related log events contain extended attributes. Setting this option to `true` enables audit capabilities, but negatively affects performance. Genesys recommends enabling this option for Solution Control Server and Configuration Server when using audit tracking. For other applications, refer to *Genesys 8.1 Combined Log Events Help* to find out whether an application generates Interaction-level and Audit-related log events; if it does, enable the option only when testing new interaction scenarios.

check-point

Default Value: `1`
Valid Values: `0`–`24`
Changes Take Effect: Immediately

Specifies, in hours, how often the application generates a `check point` log event, to divide the log into sections of equal time. By default, the application generates this log event every hour. Setting the option to `0` prevents the generation of `check-point` events.

memory

Default Value: No default value
Valid Values: `<string>` (memory file name)
Changes Take Effect: Immediately

Specifies the name of the file to which the application regularly prints a snapshot of the memory output, if it is configured to do this (see "Log Output Options" on page 722). The new snapshot overwrites the previously written data. If the application terminates abnormally, this file will contain the latest log messages. Memory output is not recommended for processors with a CPU frequency lower than 600 MHz.

Note: If the file specified as the `memory` file is located on a network drive, an application does not create a snapshot file (with the extension `*.memory.log`).

---

memory-storage-size

Default Value: `2 MB`
Valid Values:

`<number> KB` or `<number>`	The size of the memory output, in kilobytes. The minimum value is `128 KB`.
`<number> MB`	The size of the memory output, in megabytes. The maximum value is `64 MB`.

Changes Take Effect: When memory output is created

Specifies the buffer size for log output to the memory, if configured. See also "Log Output Options" on page 722.

no-memory-mapping

Default Value: `0`
Valid Values: `0, 1`
Changes Take Effect: At restart

Specifies if a `.snapshot.log` file is disabled. By default, SIP Server generates a `.snapshot.log` file. It is recommended to set this option to `1` if log files are generated in network storage rather than on the local disk.

spool

Default Value: The application's working directory
Valid Values: `<path>` (the folder, with the full path to it)
Changes Take Effect: Immediately

Specifies the folder, including full path to it, in which an application creates temporary files related to network log output. If you change the option value while the application is running, the change does not affect the currently open network output.

compatible-output-priority

Default Value: `false`
Valid Values:

`true`	The log of the level specified by "Log Output Options" is sent to the specified output.
`false`	The log of the level specified by "Log Output Options" and higher levels is sent to the specified output.

Changes Take Effect: Immediately

Specifies whether the application uses 6.x output logic. For example, you configure the following options in the `log` section for a 6.x application and for a 7.x application:

```
[log]
verbose = all
debug = file1
standard = file2
```

The log file content of a 6.x application is as follows:

- `file1` contains Debug messages only.
- `file2` contains Standard messages only.

The log file content of a 7.x application is as follows:

- `file1` contains Debug, Trace, Interaction, and Standard messages.
- `file2` contains Standard messages only.

If you set `compatible-output-priority` to `true` in the 7.x application, its log file content will be the same as for the 6.x application.

Warning! Genesys does not recommend changing the default value of this option unless you have specific reasons to use the 6.x log output logic—that is, to mimic the output priority as implemented in releases 6.x. Setting this option to `true` affects log consistency.

Log Output Options

To configure log outputs, set log level options (`all`, `alarm`, `standard`, `interaction`, `trace`, and/or `debug`) to the desired types of log output (`stdout`, `stderr`, `network`, `memory`, and/or `[filename]`, for log file output).

You can use:

- One log level option to specify different log outputs.
- One log output type for different log levels.
- Several log output types simultaneously, to log events of the same or different log levels.

You must separate the log output types by a comma when you are configuring more than one output for the same log level. See "Examples" on page 727.

Warnings!
- If you direct log output to a file on the network drive, an application does not create a snapshot log file (with the extension `*.snapshot.log`) in case it terminates abnormally.
- 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.

Note: The log output options are activated according to the setting of the `verbose` configuration option.

all

Default Value: No default value
Valid Values (log output types):

`stdout`	Log events are sent to the Standard output (`stdout`).
`stderr`	Log events are sent to the Standard error output (`stderr`).
`network`	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. Setting the `all` log level option to the `network` output enables an application to send log events of the `Standard`, `Interaction`, and `Trace` levels to Message Server. `Debug`-level log events are neither sent to Message Server nor stored in the Log Database.
`memory`	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends all log events. The log output types must be separated by a comma when more than one output is configured. For example:

```
all = stdout, logfile
```

Note: To ease the troubleshooting process, consider using unique names for log files that different applications generate.

alarm

Default Value: No default value
Valid Values (log output types):

`stdout`	Log events are sent to the Standard output (`stdout`).
`stderr`	Log events are sent to the Standard error output (`stderr`).
`network`	Log events are sent to Message Server, which resides anywhere on the network, and Message Server stores the log events in the Log Database.
`memory`	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the Alarm level. The log output types must be separated by a comma when more than one output is configured. For example:

```
standard = stderr, network
```

standard

Default Value: No default value
Valid Values (log output types):

`stdout`	Log events are sent to the Standard output (`stdout`).
`stderr`	Log events are sent to the Standard error output (`stderr`).
`network`	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
`memory`	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Standard` level. The log output types must be separated by a comma when more than one output is configured. For example:

```
standard = stderr, network
```

interaction

Default Value: No default value
Valid Values (log output types):

`stdout`	Log events are sent to the Standard output (`stdout`).
`stderr`	Log events are sent to the Standard error output (`stderr`).
`network`	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
`memory`	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Interaction` level and higher (that is, log events of the `Standard` and `Interaction` levels). The log outputs must be separated by a comma when more than one output is configured. For example:

```
interaction = stderr, network
```

trace

Default Value: No default value
Valid Values (log output types):

`stdout`	Log events are sent to the Standard output (`stdout`).
`stderr`	Log events are sent to the Standard error output (`stderr`).
`network`	Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
`memory`	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Trace` level and higher (that is, log events of the `Standard`, `Interaction`, and `Trace` levels). The log outputs must be separated by a comma when more than one output is configured. For example:

```
trace = stderr, network
```

debug

Default Value: No default value
Valid Values (log output types):

`stdout`	Log events are sent to the Standard output (`stdout`).
`stderr`	Log events are sent to the Standard error output (`stderr`).
`memory`	Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Debug` level and higher (that is, log events of the `Standard`, `Interaction`, `Trace`, and `Debug` levels). The log output types must be separated by a comma when more than one output is configured—for example:

```
debug = stderr, /usr/local/genesys/logfile
```

Note: `Debug`-level log events are never sent to Message Server or stored in the Log Database.

Log File Extensions

You can use the following file extensions to identify log files that an application creates for various types of output:

- `*.log`—Assigned to log files when you configure output to a log file. For example, if you set `standard = confservlog` for Configuration Server, it prints log messages into a text file called `confservlog.<time_stamp>.log`.
- `*.qsp`—Assigned to temporary (spool) files when you configure output to the network but the network is temporarily unavailable. For example, if you set `standard = network` for Configuration Server, it prints log messages into a file called `confserv.<time_stamp>.qsp` during the time the network is not available.
- `*.snapshot.log`—Assigned to files that contain the output snapshot when you configure output to a log file. The file contains the last log messages that an application generates before it terminates abnormally. For example, if you set `standard = confservlog` for Configuration Server, it prints the last log message into a file called `confserv.<time_stamp>.snapshot.log` in case of failure.

Note: Provide `*.snapshot.log` files to Genesys Customer Care when reporting a problem.

- `*.memory.log`—Assigned to log files that contain the memory output snapshot when you configure output to memory and redirect the most recent memory output to a file. For example, if you set `standard = memory` and `memory = confserv` for Configuration Server, it prints the latest memory output to a file called `confserv.<time_stamp>.memory.log`.

Examples

This section presents examples of a `log` section that you might configure for an application when that application is operating in production mode and in two lab modes, debugging and troubleshooting.

Production Mode Log Section

```
[log]
verbose = standard
standard = network, logfile
```

With this configuration, an application only generates the log events of the `Standard` level and sends them to Message Server, and to a file named `logfile`, which the application creates in its working directory. Genesys recommends that you use this or a similar configuration in a production environment.

Warning! 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.

Lab Mode Log Section

```
[log]
verbose = all
all = stdout, /usr/local/genesys/logfile
trace = network
```

With this configuration, an application generates log events of the `Standard`, `Interaction`, `Trace`, and `Debug` levels, and sends them to the standard output and to a file named `logfile`, which the application creates in the `/usr/local/genesys/` directory. In addition, the application sends log events of the `Standard`, `Interaction`, and `Trace` levels to Message Server. Use this configuration to test new interaction scenarios in a lab environment.

Failure-Troubleshooting Log Section

```
[log]
verbose = all
standard = network
all = memory
```

```
memory = logfile
memory-storage-size = 32 MB
```

With this configuration, an application generates log events of the `Standard` level and sends them to Message Server. It also generates log events of the `Standard`, `Interaction`, `Trace`, and `Debug` levels, and sends them to the memory output. The most current log is stored to a file named `logfile`, which the application creates in its working directory. Increased memory storage allows an application to save more of the log information generated before a failure.

Note: If you are running an application on UNIX, and you do not specify any files in which to store the memory output snapshot, a core file that the application produces before terminating contains the most current application log. Provide the application's core file to Genesys Customer Care when reporting a problem.

Debug Log Options

The options in this section enable you to generate `Debug` logs containing information about specific operations of an application.

x-conn-debug-open

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about "open connection" operations of the application.

Warning! Use this option only when requested by Genesys Customer Care.

x-conn-debug-select

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about "socket select" operations of the application.

Warning! Use this option only when requested by Genesys Customer Care.

x-conn-debug-timers

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about the timer creation and deletion operations of the application.

Warning! Use this option only when requested by Genesys Customer Care.

x-conn-debug-write

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about "write" operations of the application.

Warning! Use this option only when requested by Genesys Customer Care.

x-conn-debug-security

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about security-related operations, such as Transport Layer Security and security certificates.

Warning! Use this option only when requested by Genesys Customer Care.

x-conn-debug-api

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about connection library function calls.

Warning! Use this option only when requested by Genesys Customer Care.

x-conn-debug-dns

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about DNS operations.

Warning! Use this option only when requested by Genesys Customer Care.

x-conn-debug-all

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
`1`	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about open connection, socket select, timer creation and deletion, write, security-related, and DNS operations, and connection library function calls. This option is the same as enabling or disabling all of the previous `x-conn-debug-<op type>` options.

Warning! Use this option only when requested by Genesys Customer Care.

log-extended Section

This section must be called `log-extended`.

level-reassign-<eventID>

Default Value: Default value of log event `<eventID>`
Valid Values:

`alarm`	The log level of log event `<eventID>` is set to `Alarm`.
`standard`	The log level of log event `<eventID>` is set to `Standard`.
`interaction`	The log level of log event `<eventID>` is set to `Interaction`.
`trace`	The log level of log event `<eventID>` is set to `Trace`.
`debug`	The log level of log event `<eventID>` is set to `Debug`.
`none`	Log event `<eventID>` is not recorded in a log.

Changes Take Effect: Immediately

Specifies a log level for log event `<eventID>` that is different than its default level, or disables log event `<eventID>` completely. If no value is specified, the

log event retains its default level. This option is useful when you want to customize the log level for selected log events.

These options can be deactivated with the option `level-reassign-disable`.

Warning! Use caution when making these changes in a production environment.

Depending on the log configuration, changing the log level to a higher priority may cause the log event to be logged more often or to a greater number of outputs. This could affect system performance.

Likewise, changing the log level to a lower priority may cause the log event to be not logged at all, or to be not logged to specific outputs, thereby losing important information. The same applies to any alarms associated with that log event.

In addition to the preceding warning, take note of the following:

- Logs can be customized only by release 7.6 or later applications.
- When the log level of a log event is changed to any level except `none`, it is subject to the other settings in the `[log]` section at its new level. If set to `none`, it is not logged and is therefore not subject to any log configuration.
- Using this feature to change the log level of a log changes only its priority; it does not change how that log is treated by the system. For example, increasing the priority of a log to `Alarm` level does not mean that an alarm will be associated with it.
- Each application in a High Availability (HA) pair can define its own unique set of log customizations, but the two sets are not synchronized with each other. This can result in different log behavior depending on which application is currently in primary mode.
- This feature is not the same as a similar feature in Universal Routing Server (URS) release 7.2 or later. In this Framework feature, the priority of log events are customized. In the URS feature, the priority of debug messages only are customized. Refer to the *Universal Routing Reference Manual* for more information about the URS feature.
- You cannot customize any log event that is not in the unified log record format. Log events of the `Alarm`, `Standard`, `Interaction`, and `Trace` levels feature the same unified log record format.

Example

This is an example of using customized log level settings, subject to the following log configuration:

```
[log]
verbose=interaction
all=stderr
```

```
interaction=log_file
standard=network
```

Before the log levels of the log are changed:

- Log event 1020, with default level `standard`, is output to `stderr` and `log_file`, and sent to Message Server.
- Log event 2020, with default level `standard`, is output to `stderr` and `log_file`, and sent to Message Server.
- Log event 3020, with default level `trace`, is output to `stderr`.
- Log event 4020, with default level `debug`, is output to `stderr`.

Extended log configuration section:

```
[log-extended]
level-reassign-1020=none
level-reassign-2020=interaction
level-reassign-3020=interaction
level-reassign-4020=standard
```

After the log levels are changed:

- Log event 1020 is disabled and not logged.
- Log event 2020 is output to `stderr` and `log_file`.
- Log event 3020 is output to `stderr` and `log_file`.
- Log event 4020 is output to `stderr` and `log_file`, and sent to Message Server.

level-reassign-disable

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When this option is set to `true`, the original (default) log level of all log events in the `[log-extended]` section are restored. This option is useful when you want to use the default levels, but not delete the customization statements.

log-filter Section

The `log-filter` section contains configuration options used to define the default treatment of filtering data in log output. This section contains one configuration option, `default-filter-type`. Refer to the chapter "Hide Selected Data in Logs" in the *Genesys 8.1 Security Deployment Guide* for complete information about this option.

log-filter-data Section

The `log-filter-data` section contains configuration options used to define the treatment of filtering data in log output on a key-by-key basis. This section contains one configuration option in the form of `<key name>`. Refer to the chapter "Hide Selected Data in Logs" in the *Genesys 8.1 Security Deployment Guide* for complete information about this option.

security Section

The `security` section contains configuration options used to specify security elements for your system. In addition to other options that may be required by your application, this section contains the configuration option `disable-rbac`, which is used to enable or disable Role-Based Access Control for an application. Refer to the chapter "Role-Based Access Control" in the *Genesys Security Deployment Guide* for complete information about this option.

sml Section

This section must be called `sml`.

Options in this section are defined as follows:

- In Genesys Administrator Extension (GAX) > `Application` object > `Options` tab
- (Obsolete) In Genesys Administrator—`Application` object > `Options` tab > `Advanced View (Annex)`
- (Obsolete) In Configuration Manager—`Application` object > `Properties` dialog box > `Annex` tab

Warning! Use the first three options in this section (`heartbeat-period`, `heartbeat-period-thread-class-<n>`, and `hangup-restart`) with great care, and only with those applications of which support for this functionality has been announced. Failure to use these options properly could result in unexpected behavior, from ignoring the options to an unexpected restart of the application.

heartbeat-period

Default Value: None
Valid Values:

`0`	This method of detecting an unresponsive application is not used by this application.
`3-604800`	Length of timeout, in seconds; equivalent to 3 seconds–7 days.

Changes Take Effect: Immediately

Specifies the maximum amount of time, in seconds, in which heartbeat messages are expected from an application. If Local Control Agent (LCA) does not receive a heartbeat message from the application within this period, it assumes the application is not responding and carries out corrective action.

This option can also be used to specify the maximum heartbeat interval for threads registered with class zero (0). This thread class is reserved for use by the Management Layer only.

If this option is not configured or is set to zero (`0`), heartbeat detection is not used by this application.

heartbeat-period-thread-class-<n>

Default Value: None
Valid Values:

`0`	Value specified by `heartbeat-period` in application is used.
`3-604800`	Length of timeout, in seconds; equivalent to 3 seconds–7 days.

Changes Take Effect: Immediately

Specifies the maximum amount of time, in seconds, in which heartbeat messages are expected from a thread of class <n> registered by an application. If a heartbeat message from the thread is not received within this period, the thread is assumed to be not responding, and therefore, the application is unable to provide service.

If this option is not configured or is set to zero (`0`), but the application has registered one or more threads of class <n>, the value specified by the value of `heartbeat-period` for the application will also be applied to these threads.

Refer to application-specific documentation to determine what thread classes, if any, are used.

hangup-restart

Default Value: `true`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

If set to `true` (the default), specifies that LCA is to restart the unresponsive application immediately, without any further interaction from Solution Control Server.

If set to `false`, specifies that LCA is only to generate a notification that the application has stopped responding.

suspending-wait-timeout

Default Value: `10`
Valid Values: `5-600`
Changes Take Effect: Immediately

Specifies a timeout (in seconds) after the Stop Graceful command is issued to an application during which the status of the application should change to `Suspending` if the application supports graceful shutdown. If the status of the application does not change to `Suspending` before the timeout expires, it is assumed that the application does not support graceful shutdown, and it is stopped ungracefully.

Use this option if you are unsure whether the Application supports graceful shutdown.

Note: Genesys recommends that you do not set this option for any Management Layer component (Configuration Server, Message Server, Solution Control Server, or SNMP Master Agent) or any DB Server. These components by definition do not support graceful shutdown, so this option is not required.

common Section

This section must be called `common`.

enable-async-dns

Default Value: `0`-for standalone deployments, `1`-for SIP Cluster deployments
Valid Values:

`0`	Disables asynchronous processing of DNS requests.
`1`	Enables asynchronous processing of DNS requests.

Changes Take Effect: Immediately

Enables the asynchronous processing of DNS requests such as, for example, host-name resolution.

Warnings!
- Use this option only when requested by Genesys Customer Care.
- Use this option only with T-Servers.

rebind-delay

Default Value: `10`
Valid Values: `0`–`600`
Changes Take Effect: After restart

Specifies the delay, in seconds, between socket-bind operations that are being executed by the server. Use this option if the server has not been able to successfully occupy a configured port.

Warning! Use this option only when requested by Genesys Customer Care.

GENESYS

Chapter

11 T-Server Common Configuration Options

This chapter describes the configuration options that are generally common to all T-Server types, with some exceptions noted. It contains the following sections:

T-Server also supports common log options described in Chapter 10, "Common Configuration Options," on page 715.

Setting Configuration Options

Unless specified otherwise, set T-Server common configuration options in the `Application` object, using one of the following navigation paths:

- In Genesys Administrator Extension (GAX) > `Application` object > `Application Options` tab
- (Obsolete) In Genesys Administrator > `Application` object > `Options` tab > `Advanced View (Options)`

- (Obsolete) In Configuration Manager > `Application` object > `Properties` > `Options` tab

Mandatory Options

Except as noted for certain environments, the configuration of common options is not required for basic T-Server operation.

TServer Section

The `TServer` section contains the configuration options that are used to support the core features common to all T-Servers.

This section must be called `TServer`.

ani-distribution

Default Value: `inbound-calls-only`
Valid Values: `inbound-calls-only, all-calls, suppressed`
Changes Take Effect: Immediately

Controls the distribution of the ANI information in `TEvent` messages. When this option is set to `all-calls`, the `ANI` attribute will be reported for all calls for which it is available. When this option is set to `suppressed`, the `ANI` attribute will not be reported for any calls. When this option is set to `inbound-calls-only`, the `ANI` attribute will be reported for inbound calls only.

background-processing

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

When set to `true`, T-Server processes all client requests in the background, giving higher priority to the rest of the messages. This ensures that it processes these messages without any significant delay.

With Background Processing functionality enabled, T-Server processes all switch messages immediately and waits until there are no switch messages before processing the message queue associated with T-Server client requests. T-Server reads all connection sockets immediately and places client requests in the input buffer, which prevents T-Server clients from disconnecting because of configured timeouts.

When T-Server processes client requests from the message queue, requests are processed in the order in which T-Server received them.

When set to `false`, T-Server processes multiple requests from one T-Server client before proceeding to the requests from another T-Server client, and so on.

background-timeout

Default Value: `60 msec`
Valid Values: See "Timeout Value Format" on page 763.
Changes Take Effect: Immediately

Specifies the time interval that T-Server waits before processing client requests in background mode. You must set the `background-processing` option to `true` in order for this option to take effect.

check-tenant-profile

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: For the next connected client

When set to `true`, T-Server only allows a client to register if the client provides the correct name and password of a T-Server Tenant. If the client provides the Tenant name concatenated with a slash (/) and the Tenant password for the Tenant to which T-Server belongs as the value of `AttributeApplicationPassword` in the `TRegisterClient` request, T-Server allows that client to register DNs that are included in the switch configuration in the Configuration Database, but it does not allow the client to register DNs that are *not* included in the switch configuration.

consult-user-data

Default Value: `separate`
Valid Values:

`separate`	Stores user data for original and consultation calls in separate structures. The data attached to the original call is available for review or changes only to the parties of that call. The data attached to the consultation call is available only to the parties of the consultation call.
`inherited`	Copies user data from an original call to a consultation call when the consultation call is created; thereafter, stores user data separately for the original and the consultation call. Changes to the original call's user data are not available to the parties of the consultation call, and vice versa.
`joint`	Stores user data for an original call and a consultation call in one structure. The user data structure is associated with the original call, but the parties of both the original and consultation calls can see and make changes to the common user data.

Changes Take Effect: For the next consultation call created

Specifies the method for handling user data in a consultation call.

Note: A T-Server client can also specify the `consult-user-data` mode in the `Extensions` attribute `ConsultUserData` key for a conference or transfer request. If it is specified, the method of handling user data is based on the value of the `ConsultUserData` key-value pair of the request and takes precedence over the T-Server `consult-user-data` option. If it is not specified in the client request, the value specified in the `consult-user-data` option applies.

customer-id

Default Value: No default value. (A value must be specified for a multi-tenant environment.)
Valid Values: Any character string
Changes Take Effect: Immediately

Identifies the T-Server customer. You must set this option to the name of the tenant that is using this T-Server. You must specify a value for this option if you are working in a multi-tenant environment.

Note: Do not configure the `customer-id` option for single-tenant environments.

dn-scope

Default Value: `undefined`
Valid Values: `undefined, switch, office, tenant`
Changes Take Effect: Immediately
Related Feature: "Switch Partitioning" on page 694

Specifies whether DNs associated with the `Switch, Switching Office,` or `Tenant` objects will be considered in the T-Server monitoring scope, enabling T-Server to report calls to or from those DNs as internal.

With a value of `tenant,` all DNs associated with the switches that are within the `Tenant` will be in the T-Server monitoring scope. With a value of `office,` all DNs associated with the switches that are within the `Switching Office` will be in the T-Server monitoring scope. With a value of `switch,` all DNs associated with the `Switch` will be in the T-Server monitoring scope.

With a value of `undefined` (the default), pre-8.x T-Server behavior applies and the switch partitioning is not turned on.

Note: Setting the option to a value of `office` or `tenant,` which requires T-Server to monitor a large set of configuration data, may negatively affect T-Server performance.

log-trace-flags

Default Value: `+iscc, +cfg$dn, -cfgserv, +passwd, +udata, -devlink, -sw, -req, -callops, -conn, -client`

Valid Values (in any combination):

`+/-iscc`	Turns on/off the writing of information about Inter Server Call Control (ISCC) transactions.
`+/-cfg$dn`	Turns on/off the writing of information about DN configuration.
`+/-cfgserv`	Turns on/off the writing of messages from Configuration Server.
`+/-passwd`	Turns on/off the writing of `AttributePassword` in TEvents.
`+/-udata`	Turns on/off the writing of attached data.
`+/-devlink`	Turns on/off the writing of information about the link used to send CTI messages to the switch (for multilink environments).
`+/-sw`	Reserved by Genesys Engineering.
`+/-req`	Reserved by Genesys Engineering.
`+/-callops`	Reserved by Genesys Engineering.
`+/-conn`	Reserved by Genesys Engineering.
`+/-client`	Turns on/off the writing of additional information about the client's connection.

Changes Take Effect: Immediately

Specifies—using a space-, comma- or semicolon-separated list—the types of information that are written to the log files.

management-port

Default Value: `0`

Valid Values: `0` or any valid TCP/IP port

Changes Take Effect: After T-Server is restarted

Specifies the TCP/IP port that management agents use to communicate with T-Server. If set to `0` (zero), this port is not used.

merged-user-data

Default Value: `main-only`
Valid Values:

`main-only`	T-Server attaches user data from the remaining call only.
`merged-only`	T-Server attaches user data from the merging call.
`merged-over-main`	T-Server attaches user data from the remaining and the merging call. In the event of equal keys, T-Server uses data from the merging call.
`main-over-merged`	T-Server attaches data from the remaining and the merging call. In the event of equal keys, T-Server uses data from the remaining call.

Changes Take Effect: Immediately

Specifies the data that is attached to the resulting call after a call transfer, conference, or merge completion.

Note: The option setting does not affect the resulting data for merging calls if the `consult-user-data` option is set to `joint`. (See "consult-user-data" on page 739.)

propagated-call-type

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Switch Partitioning" on page 694

Determines what T-Server reports as the value of the `CallType` attribute in events related to calls that have been synchronized with another site via ISCC, as follows:

- When set to `false`, T-Server reports in events related to calls that have been synchronized with another site via ISCC the same value for the `CallType` attribute as it did in pre-8.0 releases and adds the new `PropagatedCallType` attribute with the value of the `CallType` attribute at the origination site. This provides backward compatibility with existing T-Server clients.

When set to `true`, T-Server reports in events related to calls that have been synchronized with another site via ISCC the same value for the `CallType` attribute as at the origination site, and adds the new `LocalCallType` attribute with the same value as `CallType` in pre-8.0 releases.

server-id

Default Value: An integer equal to the value `ApplicationDBID` as reported by Configuration Server
Valid Values: Any integer from `0`–`16383`
Changes Take Effect: Immediately

Specifies the `Server ID` that T-Server uses to generate `Connection IDs` and other unique identifiers. In a multi-site environment, you must assign each T-Server a unique `Server ID`, in order to avoid confusion in reporting applications and T-Server behavior.

Configuration of this option is necessary for Framework environments in which there are two or more instances of the Configuration Database.

Note: If you do not specify a value for this option, T-Server populates it with the `ApplicationDBID` as reported by Configuration Server. Each data object in the Configuration Database is assigned a separate DBID that maintains a unique Server ID for each T-Server configured in the database.

Warning! Genesys does not recommend using multiple instances of the Configuration Database.

user-data-limit

Default Value: `16000`
Valid Values: `0`–`65535`
Changes Take Effect: Immediately

Specifies the maximum size (in bytes) of user data in a packed format.

Note: When T-Server works in mixed 8.x/7.x/6.x environment, the value of this option must not exceed the default value of `16000` bytes; otherwise, 6.x T-Server clients might fail.

license Section

The License section contains the configuration options that are used to configure T-Server licenses. They set the upper limit of the seat-related DN licenses `(tserver_sdn)` that T-Server tries to check out from a license file. See “License Checkout” on page 745.

This section must be called `license`.

Note: T-Server also supports the `license-file` option described in the *Genesys Licensing Guide.*

If you use two or more T-Servers, and they share licenses, you must configure the following options in the `license` section of the T-Servers.

num-of-licenses

Default Value: `0` or `max`
Valid Values: String `max` or any integer
Changes Take Effect: Immediately

Specifies how many DN licenses T-Server checks out. The values `max` or `0` (zero) check out exactly 9999 licenses. (The value `max=9999` remains for backward compatibility.) To check out any other number of licenses, specify the value as an integer, up to the number of seats supportable in your environment. The sum of all `num-of-licenses` values for all concurrently deployed T-Servers must not exceed the number of seat-related DN licenses (`tserver_sdn`) in the corresponding license file. The primary and backup T-Servers share the same licenses, and therefore they need to be counted only once. T-Server checks out the number of licenses indicated by the value for this option, regardless of the number actually in use.

num-sdn-licenses

Default Value: `0` or `max` (all DN licenses are seat-related)
Valid Values: String `max` (equal to the value of `num-of-licenses`), or any integer
Changes Take Effect: Immediately

Specifies how many seat-related licenses T-Server checks out. A value of `0` (zero) means that T-Server does not grant control of seat-related DNs to any client, and it does not look for seat-related DN licenses at all.
(The value `max= 9999` remains for backward compatibility.)

The sum of all `num-sdn-licenses` values for all concurrently deployed T-Servers must not exceed the number of seat-related DN licenses (`tserver_sdn`) in the corresponding license file. The primary and backup T-Servers share the same licenses, and therefore they need to be counted only once. T-Server checks out the number of licenses indicated by the value for this option, regardless of the number actually in use.

Notes:
- For Network T-Servers, Genesys recommends setting this option to `0`.
- Be sure to configure in the Configuration Database all the DNs that agents use (Extensions and ACD Positions) and that T-Server should control.

License Checkout

Table 125 shows how to determine the number of seat-related DN licenses that T-Server attempts to check out. See the examples on page 745.

Table 125: License Checkout Rules

Options Settings[a]			License Checkout[b]
num-of-licenses	num-sdn-licenses		Seat-related DN licenses
max (or 0)	max		9999
max (or 0)	x		x
max (or 0)	0		0
x	max		x
x	y		min (y, x)
x	0		0

a. In this table, the following conventions are used: `x` and `y` - are positive integers; `max`=9999; `min (y, x)` is the lesser of the two values defined by `y` and `x`, respectively.

b. The License Checkout column shows the number of licenses that T-Server attempts to check out. The actual number of licenses will depend on the licenses' availability at the time of checkout.

Examples

This section presents examples of option settings in the `license` section.

Table 126: Example 1

If...		Then...
Options Settings	License File Settings	License Checkout
num-of-licenses = max	tserver_sdn = 500	500 seat-related DNs
num-sdn-licenses = max		

Table 127: Example 2

If...		Then...
Options Settings	**License File Settings**	**License Checkout**
num-of-licenses = 1000	tserver_sdn = 500	500 seat-related DNs
num-sdn-licenses = max		

Table 128: Example 3

If...		Then...
Options Settings	**License File Settings**	**License Checkout**
num-of-licenses = 1000	tserver_sdn = 600	400 seat-related DNs
num-sdn-licenses = 400		

Table 129: Example 4

If...		Then...
Options Settings	**License File Settings**	**License Checkout**
num-of-licenses = max	tserver_sdn = 5000	1000 seat-related DNs
num-sdn-licenses = 1000		

agent-reservation Section

The `agent-reservation` section contains the configuration options that are used to customize the T-Server Agent Reservation feature. See "Agent Reservation" on page 656 section for details on this feature.

This section must be called `agent-reservation`.

Note: The Agent Reservation functionality is currently a software-only feature that is used to coordinate multiple client applications. This feature does not apply to multiple direct or ACD-distributed calls.

collect-lower-priority-requests

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether an agent reservation request is collected, depending on its priority during the time interval specified by the `request-collection-time` configuration option. When set to `false`, during the `request-collection-time` interval T-Server collects reservation requests of the highest priority only, rejecting newly submitted requests that have a lower priority or rejecting all previously submitted requests if a request with a higher priority arrives. When set to `true` (the default), agent reservation requests are collected as they were in pre-8.x releases.

reject-subsequent-request

Default Value: `true`
Valid Values:

`true`	T-Server rejects subsequent requests.
`false`	A subsequent request prolongs the current reservation made by the same client application for the same agent.

Changes Take Effect: Immediately

Specifies whether T-Server rejects subsequent requests from the same client application, for an agent reservation for the same `Agent` object that is currently reserved.

Note: Genesys does not recommend setting this option to `false` in a multi-site environment in which remote locations use the Agent-Reservation feature.

request-collection-time

Default Value: `100 msec`
Valid Values: See “Timeout Value Format” on page 763.
Changes Take Effect: Immediately

Specifies the interval that agent reservation requests are collected before a reservation is granted. During this interval, agent reservation requests are delayed, in order to balance successful reservations between client applications (for example, Universal Routing Servers).

reservation-time

Default Value: `10000 msec`
Valid Values: See “Timeout Value Format” on page 763.
Changes Take Effect: Immediately

Specifies the default interval for which a an Agent DN is reserved. During this interval, the agent cannot be reserved again.

extrouter Section

The `extrouter` section contains the configuration options that are used to support multi-site environments with the Inter Server Call Control (ISCC) feature. The configuration options in this section of the document are grouped with related options that support the same functionality, as follows:

This configuration section must be called `extrouter`.

For a description of the ways in which T-Server supports multi-site configurations and for an explanation of the configuration possibilities for a multi-site operation, see the "Multi-Site Support" chapter.

Note: In a multi-site environment, you must configure the `timeout`, `cast-type`, and `default-dn` options with the same value for both the primary and backup T-Servers. If you do not do this, the value specified for the backup T-Server overrides the value specified for the primary T-Server.

ipo-tout

Default Value: `5 sec`
Valid Values: See "Timeout Value Format" on page 763.
Changes Take Effect: Immediately
Related Feature: "ISCC Path Optimization" on page 695
Related Parameter: `path-optimization` in Table 122 on page 705

Specifies the number of seconds that SIP Server waits for the ISCC Path Optimization transaction to complete. The transaction fails when optimization is not completed within the specified time period. When the transaction fails, the ISCC signaling path does not match the SIP Signaling path, resulting in incorrect data propagation. Usually, path optimization takes less than 1 sec.

Note: When the `ipo-tout` setting is too high, it might lead to a memory leak, resulting in incorrect data propagation.

Genesys does not recommend to change the default value unless instructed by Genesys Customer Care.

match-call-once

Default Value: `true`
Valid Values:

`true`	ISCC does not process (match) an inbound call that has already been processed (matched).
`false`	ISCC processes (attempts to match) a call as many times as it arrives at an ISCC resource or multi-site-transfer target.

Changes Take Effect: Immediately

Specifies how many times ISCC processes an inbound call when it arrives at an ISCC resource. When set to `false`, ISCC processes (attempts to match) the call even if it has already been processed.

Note: To support multi-site tromboning scenarios, this option can be set to `false`. Contact Genesys Customer Care to confirm the option setting.

reconnect-tout

Default Value: `5 sec`
Valid Values: See "Timeout Value Format" on page 763.
Changes Take Effect: At the next reconnection attempt

Specifies the time interval after which a remote T-Server attempts to connect to this T-Server after an unsuccessful attempt or a lost connection. The number of attempts is unlimited. At startup, T-Server immediately attempts the first connection, without this timeout.

report-connid-changes

Default Value: `false`
Valid Values:

`true`	`EventPartyChanged` is generated.
`false`	`EventPartyChanged` is not generated.

Changes Take Effect: Immediately

Specifies whether the destination T-Server generates `EventPartyChanged` for the incoming call when the resulting `ConnID` attribute is different from the `ConnID` attribute of an instance of the same call at the origination location.

use-data-from

Default Value: `current`
Valid Values:

`active`	The values of `UserData` and `ConnID` attributes are taken from the consultation call.
`original`	The values of `UserData` and `ConnID` attributes are taken from the original call.
`active-data-original-call`	The value of the `UserData` attribute is taken from the consultation call and the value of `ConnID` attribute is taken from the original call.
`current`	If the value of `current` is specified, the following occurs: • Before the transfer or conference is completed, the `UserData` and `ConnID` attributes are taken from the consultation call. • After the transfer or conference is completed, `EventPartyChanged` is generated, and the `UserData` and `ConnID` are taken from the original call.

Changes Take Effect: Immediately

Specifies the call from which the values for the `UserData` and `ConnID` attributes are taken for a consultation call that is routed or transferred to a remote location.

Note: For compatibility with the previous T-Server releases, you can use the values `consult`, `main`, and `consult-user-data` for this option. These are aliases for `active`, `original`, and `current`, respectively.

ISCC Transaction Options

cast-type

Default Value: `route, route-uui, reroute, direct-callid, direct-uui, direct-network-callid, direct-notoken, direct-digits, direct-ani, dnis-pool, pullback`

Valid Values: `route, route-uui, reroute, direct-callid, direct-uui, direct-network-callid, direct-notoken, direct-digits, direct-ani, dnis-pool, pullback`

Changes Take Effect: For the next request for the remote service

Specifies—using a space-, comma- or semicolon-separated list—the routing types that can be performed for this T-Server.

The valid values provide for a range of mechanisms that the ISCC feature can support with various T-Servers, in order to pass call data along with calls between locations.

Because switches of different types provide calls with different sets of information parameters, some values might not work with your T-Server. See Table 119 on page 675 for information about supported transaction types by a

specific T-Server. The "Multi-Site Support" chapter also provides detailed descriptions of all transaction types.

Notes: For compatibility with the previous T-Server releases, you can use the `direct` value for this option. This is an alias for `direct-callid`.

An alias, `route-notoken`, has been added to the `route` value.

default-dn

Default Value: No default value
Valid Values: Any DN
Changes Take Effect: For the next request for the remote service

Specifies the DN to which a call is routed when a Destination DN (`AttributeOtherDN`) is not specified in the client's request for routing. If neither this option nor the client's request contains the destination DN, the client receives `EventError`.

Note: This option is used only for requests with route types `route`, `route-uui`, `direct-callid`, `direct-network-callid`, `direct-uui`, `direct-notoken`, `direct-digits`, and `direct-ani`.

direct-digits-key

Default Value: `CDT_Track_Num`
Valid Values: Any valid key name of a key-value pair from the `UserData` attribute
Changes Take Effect: For the next request for the remote service

Specifies the name of a key from the `UserData` attribute that contains a string of digits that are used as matching criteria for remote service requests with the `direct-digits` routing type.

Note: For compatibility with the previous T-Server releases, this configuration option has an alias value of `cdt-udata-key`.

dn-for-unexpected-calls

Default Value: No default value
Valid Values: Any DN
Changes Take Effect: Immediately

Specifies a default DN for unexpected calls arriving on an External Routing Point.

network-request-timeout

Default Value: `20 sec`
Valid Values: See "Timeout Value Format" on page 763.
Changes Take Effect: For the next network request

For a premise T-Server, this option specifies the time interval that the premise T-Server waits for a response, after relaying a `TNetwork<...>` request to the Network T-Server. For a Network T-Server, this option specifies the time interval that the Network T-Server waits for a response from an SCP (Service Control Point), after initiating the processing of the request by the SCP.

When the allowed time expires, the T-Server cancels further processing of the request and generates `EventError`.

register-attempts

Default Value: `5`
Valid Values: Any positive integer
Changes Take Effect: For the next registration

Specifies the number of attempts that T-Server makes to register a dedicated External Routing Point.

register-tout

Default Value: `2 sec`
Valid Values: See "Timeout Value Format" on page 763.
Changes Take Effect: For the next registration

Specifies the time interval after which T-Server attempts to register a dedicated External Routing Point. Counting starts when the attempt to register a Routing Point fails.

request-tout

Default Value: `20 sec`
Valid Values: See "Timeout Value Format" on page 763.
Changes Take Effect: For the next request for remote service

Specifies the time interval that a T-Server at the origination location waits for a notification of routing service availability from the destination location. Counting starts when the T-Server sends a request for remote service to the destination site.

resource-allocation-mode

Default Value: `circular`
Valid Values:

`home` T-Server takes an alphabetized (or numerically sequential) list of configured DNs and reserves the first available DN from the top of the list for each new request. For example, if the first DN is not available, the second DN is allocated for a new request. If the first DN is freed by the time the next request comes, the first DN is allocated for this next request.

`circular` T-Server takes the same list of configured DNs, but reserves a subsequent DN for each subsequent request. For example, when the first request comes, T-Server allocates the first DN; when the second request comes, T-Server allocates the second DN; and so on. T-Server does not reuse the first DN until reaching the end of the DN list.

Changes Take Effect: Immediately

Specifies the manner in which T-Server allocates resources (that is, DNs of the `External Routing Point` type and Access Resources with the `Resource Type` set to `dnis`) for multi-site transaction requests.

resource-load-maximum

Default Value: `0`
Valid Values: Any positive integer
Changes Take Effect: Immediately

Specifies the maximum number of ISCC routing transactions that can be concurrently processed at a single DN of the `External Routing Point` route type. After a number of outstanding transactions at a particular DN of the `External Routing Point` type reaches the specified number, T-Server considers the DN not available. Any subsequent request for this DN is queued until the number of outstanding transactions decreases. A value of `0` (zero) means that no limitation is set to the number of concurrent transactions at a single External Routing Point. In addition, the `0` value enables T-Server to perform load balancing of all incoming requests among all available External Routing Points, in order to minimize the load on each DN.

route-dn

Default Value: No default value
Valid Values: Any DN
Changes Take Effect: Immediately

Specifies the DN that serves as a Routing Point for the `route` transaction type in the multiple-to-one access mode.

timeout

Default Value: `60 sec`
Valid Values: See "Timeout Value Format" on page 763.
Changes Take Effect: For the next request for remote service

Specifies the time interval that the destination T-Server waits for a call routed from the origination location. Counting starts when this T-Server notifies the requesting T-Server about routing service availability. The timeout must be long enough to account for possible network delays in call arrival.

transaction-state

Default Value: `default`
Valid Values: `default, by-islink`
Changes Take Effect: On the next ISCC origination transaction

Enables improved historical reporting of data for multi-site scenarios where a call is successfully delivered to the destination site but is not answered by the target agent.

- The `default` value supports backward compatibility when T-Server reports the origination transaction and the IS-Link as unsuccessful, and stops event propagation (if configured). As a result, in most cases, historical reporting applications do not take into account the calls on different sites that are part of the same interaction.
- The `by-islink` value forces T-Server to report both the transaction and the IS-Link as successful and to maintain event propagation in those multi-site scenarios.

Keep this option's value consistent between all T-Servers participating in multi-site transactions.

Limitation: If No-Answer Supervision is used, the agent-no-answer timeout must not exceed the value of the `timeout` option in the `extrouter` section (the `default` value of this option is `60 sec`).

use-implicit-access-numbers

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After T-Server is restarted

Determines whether an External Routing Point in which at least one access number is specified is eligible for use as a resource for calls coming from switches for which an access number is not specified in the External Routing Point. If this option is set to `false`, the External Routing Point is not eligible for use as a resource for calls coming from such switches. If this option is set to `true`, an implicit access number for the External Routing Point, composed of the switch access code and the DN number of the External Routing Point, will be used.

Note: If an External Routing Point does not have an access number specified, this option will not affect its use.

Transfer Connect Service Options

tcs-queue

Default Value: No default value
Valid Values: Any valid DN number
Changes Take Effect: For the next request for the remote service

Specifies the TCS DN number to which a call, processed by the TCS feature, is dialed after the originating external router obtains an access number. This option applies only if the `tcs-use` option is activated.

tcs-use

Default Value: `never`
Valid Values:

`never`	The TCS feature is not used.
`always`	The TCS feature is used for every call.
`app-defined`	In order to use the TCS feature for a multi-site call transfer request, a client application must add a key-value pair with a `TC-type` key and a nonempty string value to either the `UserData` or Extensions attribute of the request.

Changes Take Effect: Immediately

Specifies whether the Transfer Connect Service (TCS) feature is used.

Note: For compatibility with the previous T-Server releases, you can use the value `up-app-depended` for this option. This is an alias for `app-defined`.

ISCC/COF Options

cof-ci-defer-create

Default Value: `0`
Valid Values: See “Timeout Value Format” on page 763.
Changes Take Effect: Immediately

Specifies the time interval that T-Server waits for call data from the switch before generating a negative response for a call data request from a remote T-Server. If T-Server detects the matching call before this timeout expires, it sends the requested data. This option applies only if the `cof-feature` option is set to `true`.

cof-ci-defer-delete

Default Value: `0`
Valid Values: See "Timeout Value Format" on page 763.
Changes Take Effect: Immediately

Specifies the time interval that T-Server waits before deleting call data that might be overflowed. If set to `0`, deletion deferring is disabled. This option applies only if the `cof-feature` option is set to `true`.

cof-ci-req-tout

Default Value: `500 msec`
Valid Values: See "Timeout Value Format" on page 763.
Changes Take Effect: For the next COF operation

Specifies the time interval during which T-Server will wait for call data requested with respect to a call originated at another site. After T-Server sends the call data request to remote T-Servers, all events related to this call will be suspended until either the requested call data is received or the specified timeout expires. This option applies only if the `cof-feature` option is set to `true`.

cof-ci-wait-all

Default Value: `false`
Valid Values:

`true`	T-Server waits for responses from all T-Servers that might have the requested call data before updating the call data with the latest information.
`false`	T-Server updates the call data with the information received from the first positive response.

Changes Take Effect: Immediately

Specifies whether T-Server, after sending a request for matching call data, waits for responses from other T-Servers before updating the call data (such as `CallHistory`, `ConnID`, and `UserData`) for a potentially overflowed call. The waiting period is specified by the `cof-ci-req-tout` and `cof-rci-tout` options. This option applies only if the `cof-feature` option is set to `true`.

cof-feature

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Enables or disables the Inter Server Call Control/Call Overflow (ISCC/COF) feature.

cof-rci-tout

Default Value: `10 sec`
Valid Values: See "Timeout Value Format" on page 763.
Changes Take Effect: For the next COF operation

Specifies the time interval that T-Server waits for call data from other T-Servers' transactions. Counting starts when `cof-ci-req-tout` expires. This option applies only if the `cof-feature` option is set to `true`.

local-node-id

Default Value: `0`
Valid Values: `0` or any positive integer
Changes Take Effect: Immediately

This option, if enabled, checks all networked calls against the specified `NetworkNodeID` (the identity of the switch to which the call initially arrived). If the `NetworkNodeID` is the same as the value of this option, the request for call information is *not* sent. The default value of `0` disables the functionality of this option. To establish an appropriate `NetworkNodeID`, specify a value other than the default. This option applies only if the `cof-feature` option is set to `true`.

Note: This option applies only to T-Server for Nortel Communication Server 2000/2100.

default-network-call-id-matching

Default Value: No default value
Valid Values: See the "T-Server-Specific Configuration Options" chapter for an option description for your T-Server
Changes Take Effect: Immediately

When a value for this option is specified, T-Server uses the `NetworkCallID` attribute for the ISCC/COF call matching.

To activate this feature, the `cof-feature` option must be set to `true`.

Note: SIP Server and several T-Servers support the `NetworkCallID` attribute for the ISCC/COF call matching in a way that requires setting this option to a specific value. For information about the option value that is specific for your T-Server, see the "T-Server-Specific Configuration Options" chapter of your *T-Server Deployment Guide.*

Event Propagation Options

compound-dn-representation

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies which format T-Server uses to represent a DN when reporting an `OtherDN` or `ThirdPartyDN` attribute in event propagation messages.

When set to `true`, the `<switch>::DN` (compound) format is used. This option value supports backward compatibility for pre-8.x T-Server ISCC/EPP functionality and is provided for multi-site deployments where the same DNs are configured under several switches.

When set to `false`, the `DN` (non-compound) format is used. This option value ensures more transparent reporting of `OtherDN` or `ThirdPartyDN` attributes and is recommended for all single-site deployments, as well as for multi-site deployments that do not have the same DNs configured under several switches. This option applies only if the `event-propagation` option is set to `list`.

Note: Local DNs are always represented in the non-compound `(DN)` form.

epp-tout

Default Value: `0`
Valid Values: See "Timeout Value Format" on page 763.
Changes Take Effect: Immediately

Specifies the time interval, in seconds, during which T-Server attempts to resolve race conditions that may occur in deployments that use switch partitioning or intelligent trunks. This option applies only if the `event-propagation` option is set to `list`.

Note: If the time interval is not long enough to account for possible network switching delays, T-Server may produce duplicated events, such as events that are propagated by the ISCC and generated locally.

event-propagation

Default Value: `list`
Valid Values:

`list`	Changes in user data and party events are propagated to remote locations through call distribution topology.
`off`	The feature is disabled. Changes in user data and party events are not propagated to remote locations.

Changes Take Effect: Immediately

Specifies whether the Event Propagation feature is enabled.

Number Translation Option

inbound-translator-<*n*>

Default Value: No default value
Valid Value: Any valid name
Changes Take Effect: Immediately

Specifies the name of another configuration section as the value for the `inbound-translator` option. For example,
`inbound-translator-1 = ani-translator`

where `ani-translator` is the name of the configuration that describes the translation rules for inbound numbers.

GVP Integration Option

handle-vsp

Default Value: `no`
Valid Values:

`requests`	ISCC will process and adjust requests related to this DN and containing a `Location` attribute before submitting them to the service provider.
`events`	ISCC will process and adjust events received from the service provider and containing a `Location` attribute before distributing them to T-Server clients.
`all`	ISCC will process and adjust both events and requests.
`no`	No ISCC processing of such requests and events takes place.

Changes Take Effect: Immediately

Specifies the way ISCC handles events from, and requests to, an external service provider registered for a DN using the `AddressType` attribute set to `VSP`.

backup-sync Section

The backup-synchronization section contains the configuration options that are used to support a high-availability (`hot standby` redundancy type) configuration.

This section must be called `backup-sync`.

Note: These options apply only to T-Servers that support the `hot standby` redundancy type.

addp-remote-timeout

Default Value: `0`
Valid Values: Any integer from `0`–`3600`
Changes Take Effect: Immediately

Specifies the time interval, in seconds, that the redundant T-Server waits for a response from this T-Server after sending a polling signal. The default value of `0` (zero) disables the functionality of this option. To establish an appropriate timeout, specify a value other than the default. This option applies only if the `protocol` option is set to `addp`.

addp-timeout

Default Value: `0`
Valid Values: Any integer from `0`–`3600`
Changes Take Effect: Immediately

Specifies the time interval, in seconds, that this T-Server waits for a response from another T-Server after sending a polling signal. The default value of `0` (zero) disables the functionality of this option. To establish an appropriate timeout, specify a value other than the default. This option applies only if the `protocol` option is set to `addp`.

addp-trace

Default Value: `off`
Valid Values:

`off, false, no`	No trace (default).
`local, on, true, yes`	Trace on this T-Server side only.
`remote`	Trace on the redundant T-Server side only.
`full, both`	Full trace (on both sides).

Changes Take Effect: Immediately

Specifies whether `addp` messages are traced in a log file, to what level the trace is performed, and in which direction. This option applies only if the `protocol` option is set to `addp`.

protocol

Default Value: `default`
Valid Values:

`default`	The ADDP feature is not active.
`addp`	Activates the Advanced Disconnect Detection Protocol.

Changes Take Effect: When the next connection is established

Specifies the name of the method used to detect connection failures. If you specify the `addp` value, you must also specify a value for the `addp-timeout`, `addp-remote-timeout`, and `addp-trace` options.

For secure TLS connections, you must set this option to `addp`.

sync-reconnect-tout

Default Value: `20 sec`
Valid Values: See "Timeout Value Format" on page 763.
Changes Take Effect: Immediately

Specifies the time interval after which the backup T-Server attempts to reconnect to the primary server (for a synchronized link).

call-cleanup Section

The call-cleanup section contains the configuration options that are used to control detection and cleanup of stuck calls in T-Server. For more information on stuck call handling, refer to the *Framework Management Layer User's Guide*.

This section must be called `call-cleanup`.

cleanup-idle-tout

Default Value: `0`
Valid Values: See "Timeout Value Format" on page 763.
Changes Take Effect: Immediately

Specifies the time interval, in seconds, that T-Server waits for a call to be 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 the call information from memory unconditionally. The default value of `0` disables the stuck calls cleanup.

Note: If the call-cleanup functionality is enabled in T-Server for Avaya Communication Manager, the UCID (Universal Call ID) feature must be enabled on the switch as well. This allows the UCID to be generated and passed to T-Server.

notify-idle-tout

Default Value: `0`
Valid Values: See "Timeout Value Format" on page 763.
Changes Take Effect: Immediately

Specifies the time interval, in seconds, that T-Server waits for a call to be 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. The default value of `0` disables the stuck calls notification.

periodic-check-tout

Default Value: `10 min`
Valid Values: See "Timeout Value Format" on page 763.
Changes Take Effect: Immediately

Specifies the time interval for periodic checks for stuck calls. These checks affect both notification and cleanup functionality, and are made 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 check.

Note: Setting this option to a value of less than a few seconds can affect T-Server performance.

Examples

This section presents examples of option settings in the `call-cleanup` section.

Example 1

```
cleanup-idle-tout = 0
notify-idle-tout = 0
periodic-check-tout = 10
```

With these settings, T-Server will not perform any checks for stuck calls.

Example 2

```
cleanup-idle-tout = 0
notify-idle-tout = 5 min
periodic-check-tout = 10 min
```

With these settings, T-Server performs checks every 10 minutes and sends notifications about all calls that have been idle for at least 5 minutes.

Example 3

```
cleanup-idle-tout = 20 min
notify-idle-tout = 5 min
periodic-check-tout = 10 min
```

With these settings, T-Server performs checks every 10 minutes, sends notifications about all calls that have been idle for at least 5 minutes, and attempts to clean up all calls that have been idle for more than 20 minutes.

Translation Rules Section

The section name is specified by the `inbound-translator-<n>` option. It contains options that define translation rules for inbound numbers.

You can choose any name for this section, provided that it matches the value of the section. Every option in this section corresponds to a rule and must conform to the format described below. You can configure as many rules as necessary to accommodate your business needs.

rule-<*n*>

Default Value: No default value
Valid Value: Any valid string in the following format:
`in-pattern=<input pattern value>;out-pattern=<output pattern value>`
Changes Take Effect: Immediately

Defines a rule to be applied to an inbound number. The two parts of the option value describe the input and output patterns in the rule. When configuring the pattern values, follow the syntax defined in "Using ABNF for Rules" on page 682. See "Configuring Number Translation" on page 689 for examples of these rules as well as detailed instructions for creating rules for your installation. For example, a value for this configuration option might look like this:

```
rule-01 = in-pattern=0111#CABBB*ccD;out-pattern=ABD
```

security Section

The `security` section contains the configuration options that are used to configure secure data exchange between T-Servers and other Genesys components. Refer to the *Genesys Security Deployment Guide* for complete information on the security configuration.

Timeout Value Format

This section of the document describes the values to use for those T-Server common options that set various timeouts. The current format allows you to use fractional values and various time units for timeout settings.

For timeout-related options, you can specify any value that represents a time interval, provided that it is specified in one of the following formats:

`[[[`*hours*`:]`*minutes*`:]`*seconds*`][`*milliseconds*`]`

or

`[`*hours* `hr][`*minutes* `min][`*seconds* `sec][`*milliseconds* `msec]`

Where a time unit name in italic (such as *hours*) is to be replaced by an integer value for this time unit.

Integer values with no measuring units are still supported, for compatibility with previous releases of T-Server. When you do not specify any measuring units, the units of the default value apply. For example, if the default value equals `60 sec`, specifying the value of `30` sets the option to `30` seconds.

Example 1

The following settings result in a value of 1 second, 250 milliseconds:

```
sync-reconnect-tout = 1.25
sync-reconnect-tout = 1 sec 250 msec
```

Example 2

The following settings result in a value of 1 minute, 30 seconds:

```
timeout = 1:30
timeout = 1 min 30 sec
```

Changes from Release 8.0 to 8.1

Table 130 lists the configuration options that:

- Are new or changed in the 8.1 release of T-Server
- Have been added or changed since the most recent 8.0 release of this document

If a configuration option has been replaced with another that enables the same functionality, the new option name and its location in this chapter are noted.

Table 130: Option Changes from Release 8.0 to 8.1

Option Name	Option Values	Type of Change	Details
TServer Section			
background-processing	`true, false`	See Details	Default value changed to `true`. See the option description on page 738.

GENESYS

Supplements

Related Documentation Resources

The following resources provide additional information that is relevant to this software. Consult these additional resources as necessary.

SIP Server Solution

- The *SIP Server 8.1 High-Availability Deployment Guide*, which contains reference information related to SIP Server high-availability deployment options, workflows, and deployment procedures for each supported operating system.
- The *SIP Server 8.1 Integration Reference Manual*, which contains reference information related to integrating SIP Server with SIP softswitches and gateways.
- The *Genesys Media Server Deployment Guide*, which will help you configure, install, and use Genesys Media Server.
- Release Notes and Product Advisories for this product, which are available on the Genesys Documentation website.

Management Framework

- The *Framework Deployment Guide*, which will help you configure, install, start, and stop Framework components.
- The *Framework Configuration Options Reference Manual*, which will provide you with descriptions of configuration options for other Framework components.

Genesys

- The *Genesys Events and Models Reference Manual*, which contains the T-Library API, information on `TEvents`, and an extensive collection of call models.

- *Genesys Technical Publications Glossary,* which provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms used in this document.
- *Genesys Migration Guide*, which provides documented migration strategies for Genesys product releases. Contact Genesys Customer Care for more information.

Information about supported operating systems and third-party software is available on the Genesys Documentation website in the following documents:

- *Genesys Supported Operating Environment Reference Guide*
- *Genesys Supported Media Interfaces Reference Manual*

Consult the following additional resources as necessary:

- *Genesys Hardware Sizing Guide,* which provides information about Genesys hardware sizing guidelines for the Genesys 8.x releases.
- *Genesys Interoperability Guide,* which provides information on the compatibility of Genesys products with various Configuration Layer Environments; Interoperability of Reporting Templates and Solutions; and Gplus Adapters Interoperability.
- *Genesys Licensing Guide,* which introduces you to the concepts, terminology, and procedures that are relevant to the Genesys licensing system.

For additional system-wide planning tools and information, see the release-specific listings of System-Level Documents on the Genesys Documentation website.

Genesys product documentation is available on the:

- Genesys Customer Care website.
- Genesys Documentation website.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesys.com.

Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

81fr_ref_06-2018_v8.1.001.00

You will need this number when you are talking with Genesys Customer Care about this product.

Screen Captures Used in This Document

Screen captures from the product graphical user interface (GUI), as used in this document, may sometimes contain minor spelling, capitalization, or grammatical errors. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Type Styles

Table 131 describes and illustrates the type conventions that are used in this document.

Table 131: Type Styles

Type Style	Used For	Examples
Italic	• Document titles • Emphasis • Definitions of (or first references to) unfamiliar terms • Mathematical variables Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on page 768).	Please consult the *Genesys Migration Guide* for more information. Do *not* use this value for this option. A *customary and usual* practice is one that is widely accepted and used within a particular industry or profession. The formula, $x + 1 = 7$ where x stands for . . .
Monospace font (Looks like `teletype` or `typewriter text`)	All programming identifiers and GUI elements. This convention includes: • The *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages. • The values of options. • Logical arguments and command syntax. • Code samples. Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.	Select the `Show variables on screen` check box. In the `Operand` text box, enter your formula. Click `OK` to exit the `Properties` dialog box. T-Server distributes the error messages in `EventError` events. If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls. Enter `exit` on the command line.
Square brackets ([ ])	A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.	`smcp_server -host [/flags]`
Angle brackets (< >)	A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise. **Note:** In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.	`smcp_server -host <confighost>`

GENESYS

Index

Symbols

Numerics

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X