Framework 8.0

SIP Server

Deployment Guide

The information contained herein is proprietary and confidential and cannot be disclosed or duplicated without the prior written consent of Genesys Telecommunications Laboratories, Inc.

Copyright © 2004–2010 Genesys Telecommunications Laboratories, Inc. All rights reserved.

About Genesys

Alcatel-Lucent's Genesys solutions feature leading software that manages customer interactions over phone, Web, and mobile devices. The Genesys software suite handles customer conversations across multiple channels and resources—self-service, assisted-service, and proactive outreach—fulfilling customer requests and optimizing customer care goals while efficiently using resources. Genesys software directs more than 100 million customer interactions every day for 4000 companies and government agencies in 80 countries. These companies and agencies leverage their entire organization, from the contact center to the back office, while dynamically engaging their customers. Go to www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support 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, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. 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.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the regional numbers provided on page 17. For complete contact information and procedures, refer to the ***Genesys Technical Support Guide***.

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.genesyslab.com

Document Version: 80fr_dep-sip_12-2010_v8.0.401.00

Table of Contents

List of Procedures 13

Preface 15

About SIP Server 15
Intended Audience 16
Reading Prerequisites 16
Making Comments on This Document 16
Contacting Genesys Technical Support 17

Part 1 **SIP Server Deployment** 19

New Features in Release 8.0 of SIP Server 19

Chapter 1 **SIP Server Fundamentals** 25

Overview 25
SIP Server Architecture 26
SIP Server Deployment Modes 26
Media Server Deployment Architecture 29
Redundant SIP Servers (High Availability) 30
Load Balancing 30
Multi-Threaded Architecture 31
Multi-Site Support 32
Next Steps 32

Chapter 2 **SIP Server General Deployment** 33

Prerequisites 33
Software Requirements 34
Hardware and Network Environment Requirements 34
Licensing Requirements 35
About Configuration Options 37
Network Considerations 38
Voice Quality 38

Bandwidth Requirements 39
Remote Agent Configuration 39
Deployment Sequence 40
Wizard Deployment of SIP Server 41
Wizard Configuration of SIP Server 41
Wizard Installation of SIP Server 42
Manual Deployment of SIP Server 44
Manual Configuration of Telephony Objects 44
Manual Configuration of SIP Server 46
Manual Installation of SIP Server 47
Next Steps 50

Chapter 3 Starting and Stopping SIP Server 51

Command-Line Parameters 51
Starting and Stopping with the Management Layer 53
Starting with Startup Files 54
Starting Manually 55
Verifying Successful Startup 56
Stopping Manually 57
Starting and Stopping with Windows Services Manager 57
Next Steps 58

Chapter 4 SIP Devices Support 59

Overview 59
About Trunk and Trunk Group DNs 61
Configuring Devices and Services 62
Configuring ACD Queues 63
Configuring MCUs 63
Configuring Endpoints 65
Configuring Gateways 68
Configuring Music Servers 70
Configuring Routing Points 72
Configuring Softswitches 72
Configuring an Application Service 74
Configuring a Recording Service 75
Configuring a Treatment Service 77
Configuring an MSML Service 77
Configuring Agent Logins 80
Configuring Stream Manager 80
Location of Media Files 81
Load Balancing of Media Services 81
Determining Codecs 81

Configuring Genesys Media Server 81
About Genesys Media Server 82
SIP Server and Media Server Integration 82
Genesys Media Server Integration 82

Chapter 5 SIP Server Feature Support 85

Alternate Routing 86
Alternate Routing for Stranded Calls 86
Alternate Routing for Unresponsive DNs 89
Alternate Routing for Unresponsive URS 90
Alternate Routing for Calls to an External Destination 91
Application Failure Detection 92
How Failure Detection Works 92
Feature Configuration 93
Feature Limitation 93
Associating an ACD Queue with a Routing Point 93
Asterisk Voice Mail Integration 94
Genesys Voice Mail Configuration Adapter for Asterisk 95
Message Waiting Indicator Functionality 95
Feature Configuration 96
Automatic Agent Logout 97
Feature Configuration 97
Call Recording 97
Regular Call Recording 97
Feature Configuration 99
Emergency (Manual) Call Recording 100
Call Supervision 101
Overview 101
Feature Configuration 104
Multi-Site Supervision 109
Remote Supervision 112
Call Transfer and Conference 120
Conference Calls 123
Consultation Transfers and Conferences 124
Feature Limitations 126
Class of Service 126
Outbound Dialing Rules 127
Ring-Through Rules 134
Feature Configuration 134
Contact Header Handling Options 137
Sample Call Flow Scenario 138
Feature Configuration 138
Control of SIP Response Code from within a Routing Strategy 139

Feature Configuration 139
Dial Plan 141
Dial Plan Configuration Overview 141
Dial Plan Call Flow 142
The Dial-Plan Rule 142
About Privilege Levels 147
Feature Configuration 148
DTMF Tones Generation 156
DTMF Parameters for Play Application Treatments 156
Feature Configuration 157
Dummy SDP 157
How It Works 158
Feature Configuration 160
Early Media for Inbound Calls 160
Feature Configuration 161
Emulated Agents 162
Business-Call Handling 162
Emulated Agents Support 164
Endpoint Service Monitoring 167
Passive Out-of-Service Detection 167
Active Out-of-Service Detection 168
IMS Integration 169
Genesys Contact Center in the IMS Network 169
Feature Configuration 170
Instant Messaging 171
Instant Messaging Transcript 171
Supported Call Operations 172
Feature Configuration 176
Mapping SIP Headers and SDP Messages 177
From SIP Messages to T-Library Messages 177
From T-Library Messages to SIP Messages 180
SDP Message Mapping 182
Dynamic DN Replacement 183
Media Server Reliability 183
How SIP Server Detects a Media Server Failure 184
Reliability for Conference Calls 185
Reliability for Supervisor Features 185
Reliability for Voice Call Recording 185
Feature Configuration 185
Feature Limitation 186
Music and Announcements 186
Announcement Treatments on Routing Points 186
Music Treatments on Routing Points 188

Other Treatments on Routing Points 189
Feature Limitation 189
Nailed-Up Connection 189
How SIP Server Establishes the Nailed-Up Connection 190
Processing Contact Center Calls for the Nailed-Up Connection 191
Feature Configuration 191
Feature Limitations 192
Network Asserted Identity 192
How the Mechanism Works 192
How SIP Server Supports the Mechanism 193
Inbound Calls 194
Outbound Calls 196
Internal Call (CLIR) 197
Feature Configuration 198
No-Answer Supervision 200
Business and Private Calls 200
Agent No-Answer Supervision 201
Extension No-Answer Supervision 201
Position No-Answer Supervision 202
Device-Specific Overrides 202
Extension Attributes for Overrides for Individual Calls 202
Feature Limitations 202
Outbound IP Solution Integration 203
SIP Server Features for Outbound IP Solution 203
Feature Configuration 206
Feature Limitation 207
Overload Control 207
How Overload Control Works 208
Feature Configuration 210
Personal Greeting 211
Feature Configuration 212
Playing Multiple Media Files 214
Presence from Switches and Endpoints 216
Subscription to SIP Server 217
SIP Server Subscription to Endpoints Behind a Switch 217
Endpoint Sends PUBLISH Requests to SIP Server 218
Mapping Polycom Agent-Status Updates to T-Library Events 219
Presence Integration with Microsoft Office Communications Server 2007 220
Preview Interactions 226
Preview Interaction for IM 226
Private Headers 227
Providing Call Participant Info 227

Feature Configuration 227
Providing a Caller ID 228
Quality of Service 228
Remote Media on Genesys SIP Endpoint SDK 8.0 229
Feature Configuration 229
Remote Server Registration 230
Remote Talk 230
SIP Traffic Monitoring 231
How it Works 231
Feature Configuration 231
Feature Limitations 232
Transport Layer Security for SIP Traffic 232
About TLS 232
Feature Configuration 233
Trunk Capacity Support 235
Tromboning Control 235
Duplicated DN Names 236
Bounced Calls Between T-Servers 237
Feature Configuration 237
Trunk Optimization for Multi-Site Transfers 238
Feature Configuration 241
Feature Limitation 242
Video Support 242
Push Video 242
Other Supported Scenarios 244
Feature Configuration 244
Working with Multiple Devices 247
Device Selection Procedure 247
Selection Based on Geo-Location 249
Feature Configuration 249
Genesys Voice Platform Integration 253
Genesys Media Server 253

Chapter 6 T-Library Functionality Support 255

Using T-Library Functions 255
Using the Extensions Attribute 264
Error Messages 272
Known Limitations 274
Third-Party Equipment—Known Limitations 275

Chapter 7 SIP Server Configuration Options 277

Application-Level Options 277

T-Server Section 277
UPDATE, INVITE, INFO, and REFER Sections 327
Log Section 328
Multi-Site Support Section 328
SIP Error Map Section 328
Agent Login–Level and DN-Level Options 330
GVP Integration Options 368
Reserved Options 369
Changes from Release 7.6 to Release 8.0 370

Part 2 T-Server Common Functions and Procedures 375

New for All T-Servers in 8.0 375

Chapter 8 T-Server Fundamentals 377

Learning About T-Server 377
Framework and Media Layer Architecture 377
T-Server Requests and Events 379
Advanced Disconnect Detection Protocol 383
Redundant T-Servers 384
Multi-Site Support 388
Agent Reservation 388
Client Connections 389

Chapter 9 Multi-Site Support 391

Multi-Site Fundamentals 391
ISCC Call Data Transfer Service 393
ISCC Call Flows 394
ISCC Transaction Types 400
T-Server Transaction Type Support 408
Transfer Connect Service Feature 412
ISCC/Call Overflow Feature 413
Number Translation Feature 417
Number Translation Rules 418
Network Attended Transfer/Conference Feature 425
Event Propagation Feature 427
User Data Propagation 428
Party Events Propagation 429
Switch Partitioning 430
Event Propagation Configuration 431
ISCC Transaction Monitoring Feature 434
Configuring Multi-Site Support 434

Applications 435
Switches and Access Codes 436
DNs 442
Configuration Examples 447

Chapter 10 Common Configuration Options 449

Setting Configuration Options 450
Mandatory Options 450
Log Section 450
Log Output Options 455
Examples 459
Debug Log Options 460
Log-Extended Section 463
Log-Filter Section 465
Log-Filter-Data Section 465
Security Section 465
SML Section 466
Common Section 467
Changes from 7.6 to 8.0 467

Chapter 11 T-Server Common Configuration Options 469

Setting Configuration Options 469
Mandatory Options 470
T-Server Section 470
License Section 475
Agent-Reservation Section 478
Multi-Site Support Section 479
ISCC Transaction Options 481
Transfer Connect Service Options 485
ISCC/COF Options 486
Event Propagation Options 488
Number Translation Option 489
Translation Rules Section 489
Backup-Synchronization Section 490
Call-Cleanup Section 491
Security Section 493
Timeout Value Format 493
Changes from Release 7.6 to 8.0 494

Supplements Related Documentation Resources 495

Document Conventions 497

Index 499

List of Procedures

Installing SIP Server on UNIX using the wizard 42
Installing SIP Server on Windows using the wizard 43
Configuring SIP Server manually 46
Installing SIP Server on UNIX manually 47
Installing SIP Server on Windows manually 49
Verifying the manual installation of SIP Server 49
Configuring SIP Server to start with the Management Layer 53
Starting SIP Server on UNIX with a startup file 54
Starting SIP Server on Windows with a startup file 55
Starting SIP Server on UNIX manually 56
Starting SIP Server on Windows manually 56
Stopping SIP Server on UNIX manually 57
Stopping SIP Server on Windows manually 57
Configuring an ACD Queue 63
Configuring an MCU 64
Configuring endpoints 65
Configuring a gateway 68
Configuring a Music Server 70
Configuring Routing Points 72
Configuring softswitches 73
Configuring an application service 74
Configuring a recording service 75
Configuring a treatment service 77
Configuring an MSML Service 78
Configuring remote supervision 113
Configuring a COS DN 135
Controlling SIP Response Codes from a Routing Strategy 140
Configuring a Dial Plan DN 149
Configuring Class of Service for a Dial Plan 151

Including additional dial-plans 152
Assigning the dial plan to a device 153
Assigning COS to a device 155
Assigning the dial plan and COS globally 155
Configuring a personal greeting 213
Enabling presence subscription 217
Configuring a gcti::video device 245
Configuring a video service 246
Setting the geo-location for a call 251
Activating Transfer Connect Service 412
Configuring Number Translation 425
Activating Event Propagation: basic configuration 432
Modifying Event Propagation: advanced configuration 432
Configuring T-Server Applications 435
Configuring Default Access Codes 437
Configuring Access Codes 438
Configuring access resources for the route transaction type 442
Configuring access resources for the dnis-pool transaction type 444
Configuring access resources for direct-* transaction types 444
Configuring access resources for ISCC/COF 445
Configuring access resources for non-unique ANI 445
Modifying DNs for isolated switch partitioning 446

Preface

Welcome to the *Framework 8.0 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 8.0 Deployment Guide,* and the Release Note for SIP Server.

This document is valid only for the 8.0 release of this product

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

This preface contains the following sections:

- About SIP Server, page 15
- Intended Audience, page 16
- Making Comments on This Document, page 16
- Contacting Genesys Technical Support, page 17

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on page 495.

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 8.0 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 8.0 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 8.0. If you take that approach, please also read Release Notes and refer to other related resources, such as the *Genesys 7 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.
- Configuration Manager 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 8.0 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@genesyslab.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 Technical Support 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 Technical Support

If you have purchased support directly from Genesys, contact Genesys Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North and Latin America	+888-369-5555 (toll-free) +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-1276-45-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Malaysia	1-800-814-472 (toll-free) +61-7-3368-6868	support@genesyslab.com.au
India	000 800 100 7136 (toll-free) +91-(022)-3918-0537	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp
Before contacting technical support, refer to the *Genesys Technical Support Guide* for complete contact information and procedures.		

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 25, provides information about SIP Server architectures and deployment considerations.
- Chapter 2, "SIP Server General Deployment," on page 33, presents configuration and installation procedures for SIP Server.
- Chapter 3, "Starting and Stopping SIP Server," on page 51, 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 59, 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 85, describes which features SIP Server supports and how to configure them.
- Chapter 6, "T-Library Functionality Support," on page 255, describes the T-Library functionality that SIP Server supports, known limitations, and error messages.
- Chapter 7, "SIP Server Configuration Options," on page 277, describes configuration options specific to SIP Server.

New Features in Release 8.0 of SIP Server

SIP Server introduces a variety of new features across several 8.0.x releases.

8.0.4 Features

The following new features were introduced in release 8.0.4 of SIP Server.

- **SIP traffic monitoring.** SIP Server can now actively monitor SIP traffic in order to initiate a switchover after a configured length of time with no SIP messages. See “SIP Traffic Monitoring” on page 231.
- **Video contact center with Genesys Media Server.** SIP Server now supports video contact center capability using the Genesys Media Server. This includes Video on Hold/Queue, Push Video, and Video Conference with active speaker detection. See “Video Support” on page 242.
- **Enhanced SIP message mapping.** SIP Server now supports SIP message mapping for the Unicode character set. For details, see the option description for “encoding” on page 290.
- **Dummy SDP support.** SIP Server can now include a “dummy SDP” in the initial SIP `INVITE` message that it sends in certain call routing scenarios. See “Dummy SDP” on page 157.
- **Dynamic DN replacement.** SIP Server can now dynamically replace the DN parameter within SIP headers, improving integration with other SIP elements. See “Dynamic DN Replacement” on page 183.
- **Staggering of Busy Lamp Field (BLF) `SUBSCRIBE` messages.** SIP Server can now stagger the time interval it uses when sending `SUBSCRIBE` messages for the BLF feature. For details, see the option description for “subscription-delay” on page 324.
- **Routing outbound calls to external destinations using `REFER`.** SIP Server can now route outbound calls to en external destination using the SIP `REFER` method, placing SIP Server out of the signaling path. See “Routing to External Destinations using REFER—Outbound Calls” on page 122.
- **Preview interaction for IM.** SIP Server now supports the preview mechanism for IM-enabled DNs, allowing the IM user to accept or reject incoming interaction before they arrive. See “Preview Interaction” on page 172.
- **Tromboning Control.** SIP Server supports anti-tromboning for ISCC transfers between SIP and T-Server instances. See “Tromboning Control” on page 235.
- **Dial plan on Trunk DNs.** SIP Server now supports assigning dial plans to `Trunk` DNs for inbound calls. See “Dial Plan” on page 141.

8.0.3 Features

The following new features are now available in release 8.0.3 of SIP Server:

- **Overload Control.** SIP Server now supports an overload control mechanism, which can gracefully handle situations where the load on the SIP Server exceeds its configured rate capacity threshold(s). See “Overload Control” on page 207.
- **Alternate Routing for Calls to an External Destination.** SIP Server can now route inbound 1pcc calls to a specified DN when the `INVITE` is addressed to an external destination. To support this feature, a new

Application-level configuration option, `default-route-point`, has been added. See "Alternate Routing for Calls to an External Destination" on page 91.

- **Support for Genesys Media Server.** SIP Server supports integration with the Genesys Media Server for providing media services including call treatments, call recording, call supervision and basic IVR. See "Configuring Genesys Media Server" on page 81.
- **Enhanced Support for Dial Plan.** SIP Server supports additional functionality for the Dial Plan feature:
  - Ability to configure a dial plan for forwarded calls.
  - Alternate routing when the dial plan destination is busy.
  - Alternate routing when the dial plan destination is set to do-not-disturb.

  For more information, see
- **Network Asserted Identity.** SIP Server supports the network asserted identity mechanism defined by RFC 3323, RFC 3324, and RFC 3325. See "Network Asserted Identity" on page 192.
- **Control of SIP Response Code from within a Routing Strategy.** SIP Server supports user-specified error code responses for rejected calls, as applied in the `Extensions` attribute in call requests that are made from a routing strategy. See "Control of SIP Response Code from within a Routing Strategy" on page 139.
- **Multi-Threaded Architecture.** SIP Server supports multi-threading for enhanced scalability for machines with multiple CPUs. See "Multi-Threaded Architecture" on page 31.
- **Application Failure Detection.** SIP Server supports Management Layer monitoring of the SIP Server application, in order to take corrective action if the application or one of its threads fails. See "Application Failure Detection" on page 92.
- **Remote Media on Genesys SIP Endpoint SDK 8.0.** SIP Server supports remote 3pcc Beep Tones Control and DTMF Tones Control on custom endpoints that are built from the Genesys SIP Endpoint SDK 8.0. See "Remote Media on Genesys SIP Endpoint SDK 8.0" on page 229.
- **IMS Integration.** SIP Server supports integration with IP Multimedia Subsystem (IMS) networks. See "IMS Integration" on page 169.
- **Transport Layer Security for SIP Traffic.** SIP Server supports secure communication using TLS for the SIP listening port. See "Transport Layer Security for SIP Traffic" on page 232.
- **Presence Integration for Microsoft OCS 2007 R2.** SIP Server now supports the ability to subscribe to the presence status for Microsoft Office Communications Server 2007 R2 users. See "Presence Integration with Microsoft Office Communications Server 2007" on page 220.

- **Enhanced Support for Personal Greeting.** SIP Server now enables you to configure how SIP Server provides personal greetings in different scenarios. For example, enable greetings after transfers or conferences. See "Personal Greeting" on page 211.

8.0.2 Features The following new features were introduced in release 8.0.2 of SIP Server.

- **Network Attended Transfer/Conference (NAT/C).** SIP Server now supports the NAT/C feature, which allows agents working in multi-site contact centers to consult with each other before making call transfers or conferences. See "Network Attended Transfer/Conference Feature" on page 425.
- **Call supervision for transferred calls.** SIP Server now supports supervisor monitoring for DNs that are moved from a consultation call to the main call as a result of a two-step transfer. See "Call Supervision For Transferred Calls" on page 108.
- **Stranded Call Routing.** SIP Server now offers alternate routing for stranded calls (calls left waiting in a queue after the last agent logs out) and stranded-on-arrival calls (calls that arrive at a queue that has no remaining logged-in agents). See "Alternate Routing" on page 86
- **Alternate routing for unresponsive DNs.** SIP Server now supports alternate routing for new calls to Genesys SIP endpoints that fail to respond to the `INVITE` request. See "Alternate Routing" on page 86.
- **Enhanced active out-of-service-detection.** SIP Server now 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. See "Forwarding OPTIONS Through a Proxy" on page 168.
- **Enhanced control over SIP INFO messages.** SIP Server now includes a new option, `info-pass-through`, that lets you control whether SIP Server will pass SIP INFO messages to a remote device. For details, see "info-pass-through" on page 298.
- **Support for Outbound IP Solution.** SIP Server includes several new features to support the Outbound IP Solution. See "Outbound IP Solution Integration" on page 203.

8.0.1 Features The following new features were introduced in the 8.0.1 release of SIP Server.

- **Playing multiple media files.** SIP Server can now play different media files for various contact center music-on-hold and video-on-hold requirements, from a single instance of SIP Server. See "Playing Multiple Media Files" on page 214.
- **Contact header handling options.** SIP Server supports two methods for handling the `Contact` header in `REGISTER` requests: it can preserve information contained in the header; or it can replace the user-name part of the URI with the DN name. See "Contact Header Handling Options" on page 137.

- **Recording for consultation calls.** SIP Server now supports regular call recording on consultation calls. See “Consultation Calls” on page 98.
- **Busy tone for more than five seconds.** SIP Server can now play a busy tone for five seconds or longer. See “busy-tone-duration” on page 284.
- **Mapping agent actions to T-Library events.** SIP Server can now support agent login and logout functionality from endpoints by using `SUBSCRIBE` and `NOTIFY` requests. See “Mapping Polycom Agent-Status Updates to T-Library Events” on page 219.
- **Enhanced reliability for media services.** SIP Server now immediately reinvites the next available media server, in cases where a `BYE` message is received from the media server while playing music-on-hold or ringback tone, or when the HTTP stream is disconnected.

  SIP Server now supports media reliability for conference, call-recording, and supervisor-feature services. See “Media Server Reliability” on page 183.

8.0.0 Features

The following features were added in the initial 8.0.0 release of SIP Server.

- **`TCompleteTransfer` using `REFER` or `REFER` with `Replaces` header.** SIP Server now supports the `TCompleteTransfer` operation by using either the SIP `REFER` method or the SIP `REFER` method with the `Replaces` header. See “TCompleteTransfer using REFER or REFER with Replaces” on page 125.
- **Automatic Agent Logout.** SIP Server can now automatically log out an agent after a specified period of inactivity, ensuring accurate reporting of agent activity. See “Automatic Agent Logout” on page 97.
- **Enhanced support for geo-location.** SIP Server now supports the ability to assign the `geo-location` for a call from the routing strategy, which takes precedence over the `geo-location` that is configured at the DN level in Configuration Manager. See “Selection Based on Geo-Location” on page 249.
- **Support for `TClearCall` requests.** `TClearCall` now instructs SIP Server to delete all parties from any type of call. This includes call operations that are currently underway or currently queued (for example, a `HoldCall` operation that is delayed while waiting for a response to an `INVITE` request).
- **Enhanced support for high availability (HA).** SIP Server now supports several additional capabilities related to high availability deployments:
  - SIP Server now supports high-availability configurations where both primary and backup instances of SIP Server reside on a single host server.
  - SIP Server now synchronizes the SIP registration `Contact` header for a particular device across both primary and backup instances of SIP Server.

  - SIP Server now supports the use of the F5 Networks® BIG-IP® Local Traffic Manager™ to manage the virtual IP address of a SIP Server HA pair.

  For more information about deploying SIP Server in a high-availability configuration, see the *Framework 8.0 SIP Server High-Availability Deployment Guide*.

- **Nailed-up connection.** SIP Server can now provide a persistent SIP session for agents who require a dedicated connection to the contact center—typically for time-division multiplexing (TDM) agents dialing in to the contact center from the public switched telephone network (PSTN). See "Nailed-Up Connection" on page 189.
- **Enhanced Agent No-Answer Supervision.** SIP Server can now apply Agent No-Answer Supervision functionality (alternate routing or forced agent logout) in cases where an agent DN returns a `4xx` rejection response to an `INVITE` request. See "No-Answer Supervision" on page 200.
- **Support for Quality of Service (QoS).** SIP Server can now set QoS bits to a user-defined value in order to prioritize SIP signaling traffic. See "Quality of Service" on page 228.
- **Stuck calls cleanup.** SIP Server now supports the detection and cleanup of stuck calls.
- **Support for early media.** SIP Server now 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). See "Early Media for Inbound Calls" on page 160.
- **Support for ITU-T recommendation E.164.** SIP Server now supports the E.164 recommendation for the international public telecommunication numbering plan.

Notes: For a list of new features that are common to all T-Servers, see "New for All T-Servers in 8.0" on page 375 of this document.

For a list of configuration-option changes that apply to SIP Server, see "Changes from Release 7.6 to Release 8.0" on page 370.

Chapter

1 SIP Server Fundamentals

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

- Overview, page 25
- SIP Server Architecture, page 26
- Redundant SIP Servers (High Availability), page 30
- Load Balancing, page 30
- Multi-Threaded Architecture, page 31
- Multi-Site Support, page 32
- Next Steps, page 32

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.

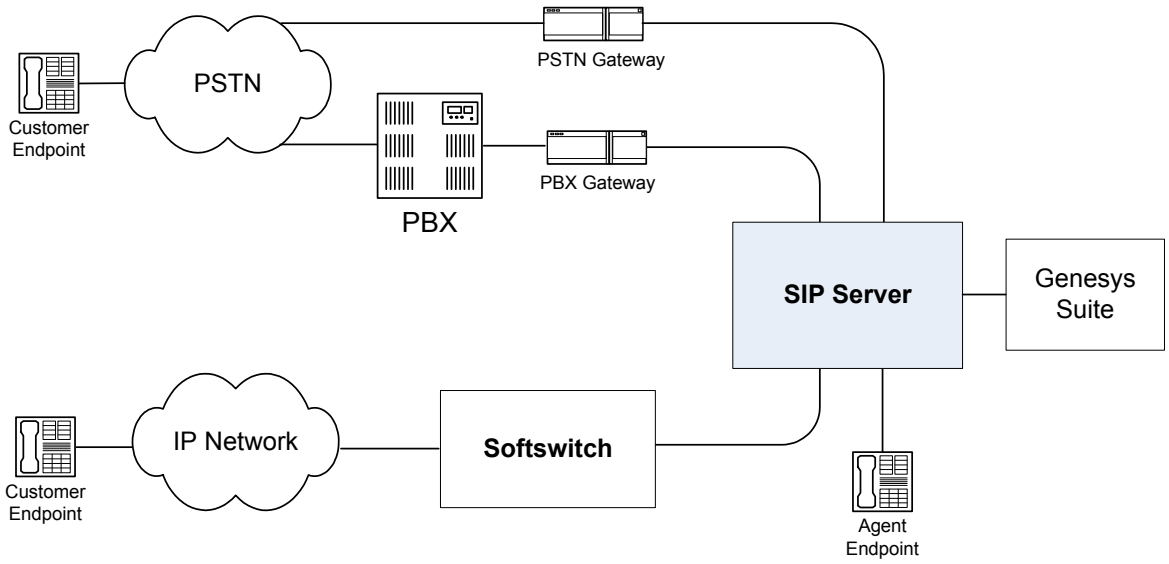

Figure 1: SIP Network Architecture

SIP Server provides all SIP signaling and T-Server functions. Stream Manager 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 59 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:

- Stand-alone mode
- Application Server mode

- Customer-side proxy mode

Stream Manager is used as an MCU in these scenarios. For more information, see the *Framework 7.6 Stream Manager Deployment Guide*.

Stand-alone Mode

Figure 2 illustrates SIP Server in the Stand-alone mode.

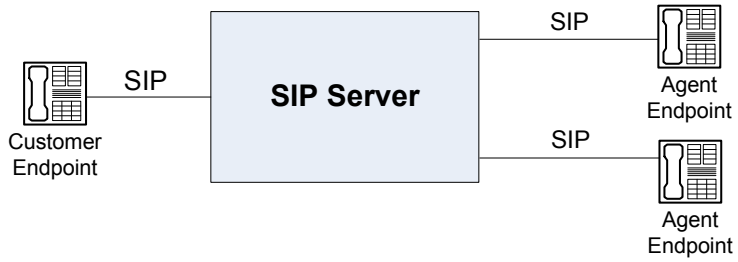

Figure 2: Stand-alone Mode

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 Manager.

  When, for each agent DN, an IP address is configured on the DN `Annex` tab. For example, for DN `1077` on the `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.

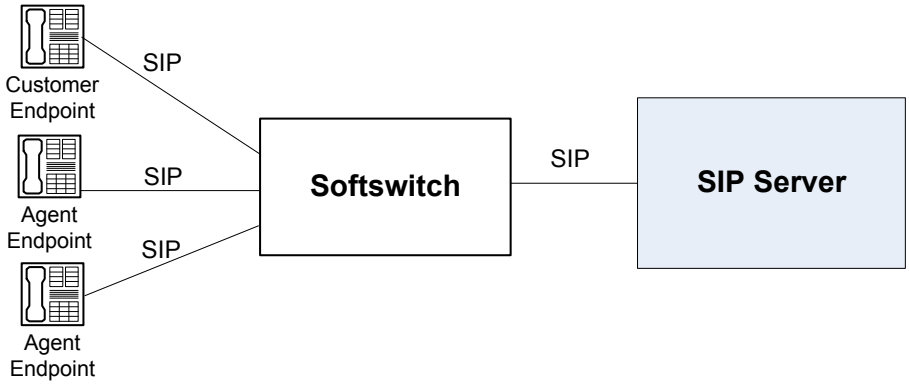

Figure 3: Application Server Mode

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.

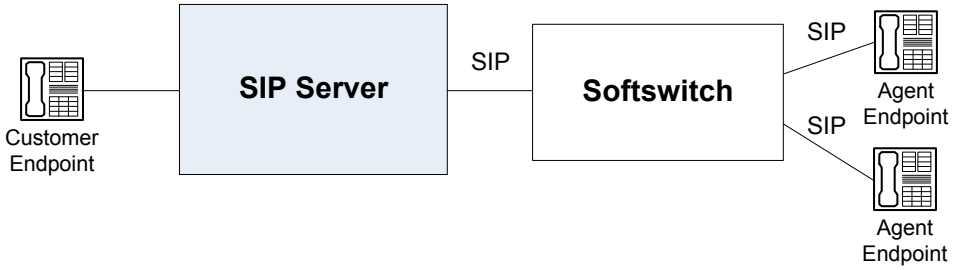

Figure 4: Customer-Side Proxy Mode

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

For outbound calls (from agents to customers), the IP addresses are determined from the `Request URI` message, or they are configured in Configuration Manager 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 Stream Manager used in conjunction with SIP Server and Genesys business applications.

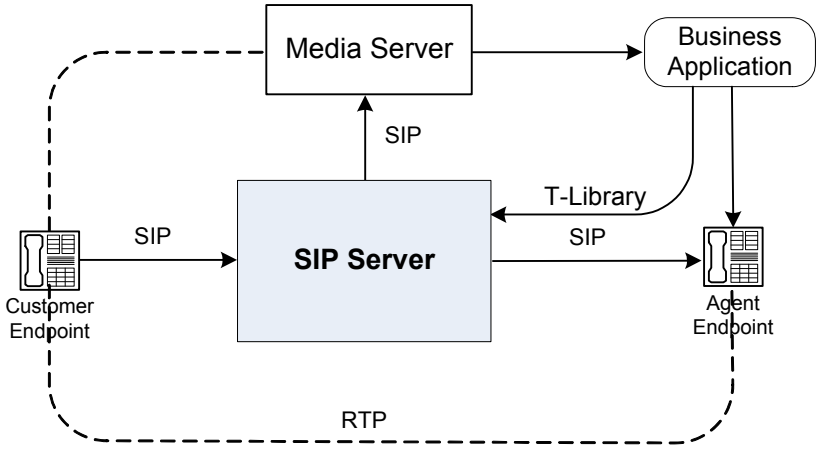

Figure 5: Media Server Deployment Architecture

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 8.0 SIP Server High-Availability Deployment Guide*.

Load Balancing

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

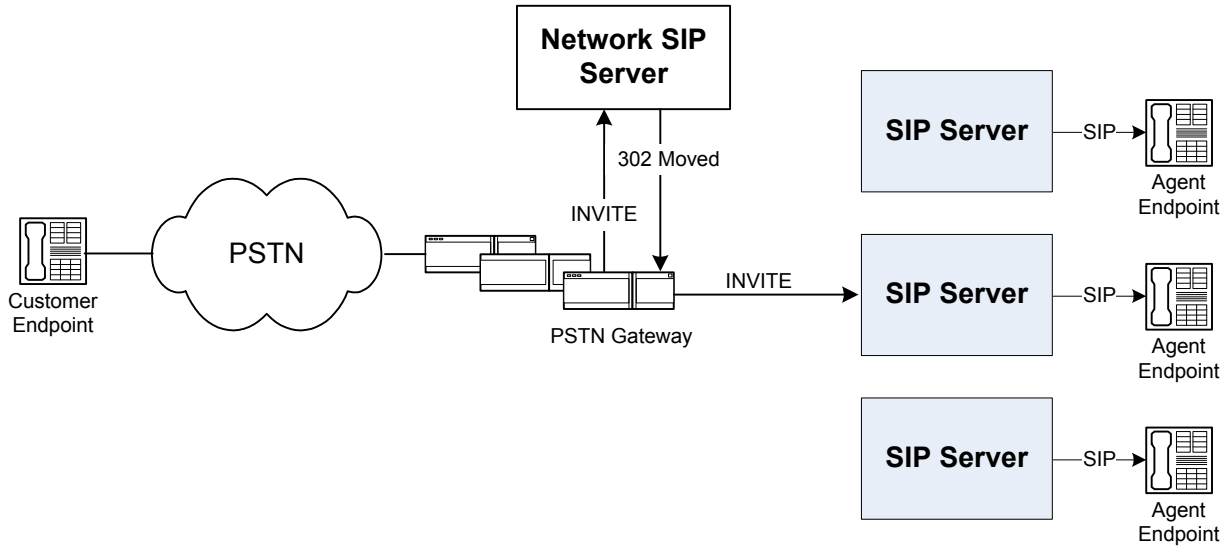

Figure 6: SIP Server Sample Load-Balancing Configuration

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. Each can serve up to 2000 agents.
- 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 30). 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

In release 8.0.3, using the `sip-link-type` (page 316) option, SIP Server can be configured to run in either multi-threaded mode, or as a single thread for backward compatibility. If running in multi-threaded mode, in addition to the main thread responsible for processing T-Library requests and distributing Events, SIP Server creates a separate thread to manage SIP calls and process SIP signaling, and another thread for the SIP transport layer.

Logging

By default, logging in the multi-threaded mode is provided in a single stream. This leads to a mix of messages from different threads in a single log file. However, using the `x-sip-log` (page 328) option, you can configure SIP Server to split message logging into several streams. In release 8.0.3, log-splitting results in two separate log files: one for T-Library specific logging, another for SIP processing.

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 391.

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 33. Otherwise, you may want to proceed to Chapter 5, “SIP Server Feature Support,” on page 85, where you will find information about feature configurations that SIP Server supports.

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:

- Prerequisites, page 33
- Network Considerations, page 38
- Deployment Sequence, page 40
- Wizard Deployment of SIP Server, page 41
- Manual Deployment of SIP Server, page 44
- Next Steps, page 50

Note: You *must* read the *Framework 8.0 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, Configuration Manager, and, at your option, Deployment Wizards. If you intend to monitor or control SIP Server through the Management Layer, you must also install and configure components of this Framework layer, such as Local Control Agent (LCA), Message Server, Solution Control Server (SCS), and Solution Control Interface (SCI), before deploying SIP Server.

Refer to the *Framework 8.0 Deployment Guide* for information about, and deployment instructions for, these Framework components.

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 Manua*l white paper for the list of operating systems and database systems supported in Genesys releases 7.x. You can find this document on the Genesys Technical Support website at http://genesyslab.com/support/dl/retrieve/default.asp?item=B6C52FB62DB42BB229B02755A3D92054&view=item.

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.

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 8.0 Deployment Guide* for recommendations on server locations.

Supported Platforms

Refer to the *Genesys Supported Media Interfaces* white paper for the list of supported switch and PABX versions. You can find this document on the Genesys Technical Support website at http://genesyslab.com/support/dl/retrieve/default.asp?item=A9CB309AF4DEB8127C5640A3C32445A7&view=item.

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 7 Licensing Guide* available on the Genesys Documentation Library DVD.

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 7 Licensing Guide* for complete licensing information.

Licensing Basic Implementations

A stand-alone 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 7 Licensing Guide* available on the Genesys Documentation Library DVD.

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 relevant Wizard pages or on the `Options` tab of your SIP Server `Application` object in Configuration Manager. 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 469. *SIP Server-specific* configuration options are described in Chapter 7, "SIP Server Configuration Options," on page 277. SIP Server also supports unified Genesys log options, as described in the Chapter 10, "Common Configuration Options," on page 449.

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 8.0 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.

Note: Remote agent deployment behind media gateways on circuit-switching networks (such as PSTN) is not fully supported in the current release of SIP Server. If inbound calls are answered by, or transferred to, such remote agents, the `Annex` tab > `TServer` section of the corresponding `DN` object in Configuration Manager must have the `refer-enabled` option set to `false`. This value limits call-transfer functionality to `Consult Transfer` only. In addition, your media gateway must be configured as a DN of type `Trunk`, with the `refer-enabled` option set to `false`.

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

Genesys recommends deploying SIP Server by using the SIP Server Configuration Wizard. However, if for some reason you must manually deploy SIP Server, you will also find instructions for doing that in this chapter.

The recommended sequence to follow before deploying SIP Server is described below.

Task Summary: SIP Server Deployment Sequence

Objective	Related Procedures and Actions
1. Deploy Configuration Layer objects and ensure Configuration Manager is running.	See the *Framework 8.0 Deployment Guide* for details.
2. Deploy Network objects (such as Host objects).	See the *Framework 8.0 Deployment Guide* for details.
3. Deploy the Management Layer.	See the *Framework 8.0 Deployment Guide* for details.

Task Summary: SIP Server Deployment Sequence (Continued)

Objective	Related Procedures and Actions
4. Deploy SIP Server using the Wizard (recommended), or manually.	See "Wizard Deployment of SIP Server" on page 41. If you are deploying SIP Server manually, see "Manual Deployment of SIP Server" on page 44.
5. Test your configuration and installation.	See Chapter 3, "Starting and Stopping SIP Server," on page 51.

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.

Wizard Deployment of SIP Server

Configuration wizards facilitate component deployment. SIP Server configuration and installation involves many steps, and Genesys strongly recommends that you set up SIP Server using the Wizard rather than manually. The SIP Server Configuration Wizard guides you through a series of steps and options to customize your deployment of SIP Server.

Wizard Configuration of SIP Server

The first step to take for a Wizard-based configuration is to install and launch Genesys Wizard Manager. (Refer to the *Framework 8.0 Deployment Guide* for instructions.) When you first launch Genesys Wizard Manager, it suggests that you set up the Management Layer and then the Framework. The Framework setup begins with configuring and creating the objects related to SIP Server, starting with the `Switch` and `Switching Office` objects, and the SIP Server's `Application` object itself.

Note: With the wizard, you create your SIP Server `Application` object in the course of creating your `Switch` object.

During creation of the `Switch` object, you also have an opportunity to run the Log Wizard to set up SIP Server logging. Then, you can specify values for the most important SIP Server options. Finally, you can create contact center objects related to SIP Server, such as `DNs`, `Agent Logins`, and some others.

Note: During configuration of a `Switch` object, the Wizard prompts you to copy a SIP Server installation package to an assigned computer. After that package is copied to the destination directory on the SIP Server host, complete the last steps of the SIP Server configuration. Then, install SIP Server on its host.

After you complete the Framework configuration, the Genesys Wizard Manager screen no longer prompts you to set up the Framework. Instead, it suggests that you set up your solutions or add various contact center objects to the Framework configuration, including the `Switch, DNs` and `Places, Agent Logins, Agent Groups, Place Groups`, and, in a multi-tenant environment, a `Tenant`. In each case, click the link for the object you wish to create. Again, you create a new SIP Server `Application` object in the course of creating a new `Switch` object.

Wizard Installation of SIP Server

After creating and configuring your SIP Server and its related components with the wizard, you can proceed to SIP Server installation. That installation process closely mimics that of previously installed components.

Note: Certain Wizard-related procedures are not described in this document. Refer to the *Framework 8.0 Deployment Guide* for general instructions.

Warning! Genesys does not recommend installation of its components via a Microsoft Remote Desktop connection. The installation should be performed locally.

Procedure: Installing SIP Server on UNIX using the wizard

Start of procedure

1. In the directory to which the SIP Server installation package was copied during Wizard configuration, 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, confirm the application name of the SIP Server that is to be installed.

5. Specify the destination directory into which SIP Server is to be installed, with the full path to it.
6. If asked which version of the product to install, the 32-bit or the 64-bit, choose the one appropriate to your environment.
7. Specify the license information that SIP Server is to use.
8. 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 test your configuration and installation, go to Chapter 3, "Starting and Stopping SIP Server," on page 51, and try it out.
- To configure and install redundant SIP Servers, see "Redundant SIP Servers (High Availability)" on page 30.
- To install T-Servers for a multi-site environment, proceed to Chapter 9, "Multi-Site Support," on page 391.

Procedure: Installing SIP Server on Windows using the wizard

Start of procedure

1. Open the directory to which the SIP Server installation package was copied during Wizard configuration.
2. Locate and double-click `Setup.exe` to start the installation. The `Welcome` screen launches.
3. When prompted, specify the connection parameters to the Configuration Server associated with this SIP Server.
4. Identify the SIP Server `Application` object in the Configuration Layer to be used by this SIP Server.
5. Specify the license information that SIP Server is to use.
6. Specify the destination directory into which SIP Server is to be installed.
7. Click `Install` to begin the installation.
8. 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 test your configuration and installation, go to Chapter 3, "Starting and Stopping SIP Server," on page 51, and try it out.
- To configure and install redundant SIP Servers, see "Redundant SIP Servers (High Availability)" on page 30.
- To install SIP Servers for a multi-site environment, proceed to Chapter 9, "Multi-Site Support," on page 391.

Manual 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.

Manual Configuration of Telephony Objects

This section describes how to manually configure SIP Server telephony objects if you are using Configuration Manager.

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 Configuration Manager, be sure to register a `Switching Office` object of type `SIP Switch` that accommodates your `Switch` object under `Environment`. Until you have done this, you cannot register a `Switch` object under `Resources` (single-tenant environment) or a `Tenant` (multi-tenant 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 391, 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 59.
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 434, for information on setting up DNs for multi-site operations.

Manual Configuration of SIP Server

Note: Use the *Framework 8.0 Deployment Guide* to prepare accurate configuration information. You may also want to consult *Configuration Manager 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 manually

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 8.0 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 `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 277. The configuration options common to all T-Servers are described in the Chapter 11, "T-Server Common Configuration Options," on page 469 chapter. SIP Server also uses common Genesys log options, described in the Chapter 10, "Common Configuration Options," on page 449.

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 391.

End of procedure

Next Steps

- See "Manual Installation of SIP Server" on page 47.

Manual Installation of SIP Server

The following directories on the Genesys 8.0 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 manually

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 manually" on page 46 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 manual installation of SIP Server" on page 49.
- To test your configuration and installation, go to Chapter 3, "Starting and Stopping SIP Server," on page 51, and try it out.
- To configure and install redundant SIP Servers, see "Redundant SIP Servers (High Availability)" on page 30.
- To install SIP Servers for a multi-site environment, proceed to Chapter 9, "Multi-Site Support," on page 391.

Procedure:
Installing SIP Server on Windows manually

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 manually" on page 46 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 manual installation of SIP Server" on page 49.
- To test your configuration and installation, go to Chapter 3, "Starting and Stopping SIP Server," on page 51, and try it out.
- To configure and install redundant T-Servers, see "Redundant SIP Servers (High Availability)" on page 30.
- To install SIP Servers for a multi-site environment, proceed to Chapter 9, "Multi-Site Support," on page 391.

Procedure:
Verifying the manual installation of SIP Server

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

Prerequisites

- Procedure: Installing SIP Server on UNIX manually, on page 47

- Procedure: Installing SIP Server on Windows manually, on page 49

Start of procedure

1. Open the Properties dialog box for a corresponding Application object in Configuration Manager.
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 on the Start Info tab.
4. Click Apply and OK to save any configuration updates.

End of procedure

Next Steps

At this point, you have either used the wizard to configure and install SIP Server, or you have done it manually, using Configuration Manager. In either case, if you want to test your configuration and installation, go to Chapter 3, "Starting and Stopping SIP Server," on page 51, and try it out. Otherwise, if you want to configure and install redundant SIP Servers, see "Redundant SIP Servers (High Availability)" on page 30. If you want to install SIP Server for a multi-site environment, proceed to Chapter 9, "Multi-Site Support," on page 391.

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, page 51
- Starting and Stopping with the Management Layer, page 53
- Starting with Startup Files, page 54
- Starting Manually, page 55
- Verifying Successful Startup, page 56
- Stopping Manually, page 57
- Starting and Stopping with Windows Services Manager, page 57
- Next Steps, page 58

Command-Line Parameters

You can start and stop Framework components using the Management Layer, 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

Procedure: Configuring SIP Server to start with the Management Layer

Start of procedure

1. Open the SIP Server Application's Properties dialog box.
2. Click the Start Info tab.
3. Specify the directory where the application is installed and/or is to run as the Working Directory.
4. Specify the name of the executable file as the command-line.
5. Specify command-line parameters as the Command–Line Arguments.

   The command-line parameters common to Framework server components are described on page 51.
6. When you are finished, click Apply.
7. Click OK to save your changes and exit the Properties dialog box.

End of procedure

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

After its command-line parameters are correctly specified in the Properties dialog box, you can start and stop SIP Server from Solution Control Interface (SCI), which is the graphical interface component of the Management Layer. (The starting procedure for SCI is described in the *Framework 8.0 Deployment Guide.*) *Framework 8.0 Solution Control Interface Help* provides complete instructions on starting and stopping applications.

You can also use the Management Layer 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 is not considered an application failure. Therefore, the Management Layer does not restart applications that it has stopped unless an appropriate alarm condition and alarm reaction are configured for these applications.

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 55 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 51.

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

```
-app "T-Server Nortel"
```

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 the Management Layer to start SIP Server, check whether Solution Control Interface displays `Started` or `Service Unavailable` status for the corresponding application. Refer to the "Troubleshooting" section of the *Framework 8.0 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, HA Proxy, T-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

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 51 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.

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, page 59
- Configuring Devices and Services, page 62
- Configuring Agent Logins, page 80
- Configuring Stream Manager, page 80
- Configuring Genesys Media Server, page 81

Overview

SIP devices that represent SIP endpoints are configured in Configuration Manager as the following types of DN:

- `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 different between `Trunk` and `Trunk Group` DNs, see "About Trunk and Trunk Group DNs" on page 61.

- `Voice over IP Service`—SIP services (Stream Manager, Music-On-Hold server, GVP Media Server, and so on)
- `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 Voice Mail Service (Asterisk only)	`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 DN 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 247.

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 D. This functionality is available only with internal DNs.

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

Usage Guidelines

The following table 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. ◆ Active out-of-service detection. ◆ Primary/backup redundancy for the external device. ◆ Dial plans.	• Limited features. The features available to `Trunk Group` DNs are limited by design. For example: ◆ No active out-of-service detection. ◆ No primary/backup redundancy.

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" on page 63
- "Configuring MCUs" on page 63
- "Configuring Endpoints" on page 65
- "Configuring Gateways" on page 68

- "Configuring Music Servers" on page 70
- "Configuring Routing Points" on page 72
- "Configuring Softswitches" on page 72
- "Configuring an Application Service" on page 74
- "Configuring a Recording Service" on page 75
- "Configuring a Treatment Service" on page 77
- "Configuring an MSML Service" on page 77

Configuring ACD Queues

To configure ACD Queues, follow the provided procedure using Configuration Manager.

Procedure: Configuring an ACD Queue

Start of procedure

1. Under a configured `Switch` object, select the `DNs` folder. From the `File` menu, select `New` > `DN` to create a new `DN` object.
2. In the `New DN Properties` dialog box, on the `General` tab, specify the following properties:
   a. `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.
   b. `Type`: Select `ACD Queue` from the drop-down box.
3. When you are finished, click `Apply`.

End of procedure

Configuring MCUs

To configure a Multipoint Conference Unit (MCU), follow the provided procedure using Configuration Manager.

Procedure: Configuring an MCU

Start of procedure

1. Under a configured Switch object, select the DNs folder. From the File menu, select New > DN to create a new DN object.
2. In the New DN Properties dialog box, on the General tab, specify the following properties:
   a. 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 MCU for this property.
   b. Type: Select Voice over IP Service from the drop-down box.
3. Click the Annex tab.
4. Create a section named TServer. In the TServer section, create options as specified in Table 3.

Table 3: Configuring an MCU

Option Name	Option Value	Description
contact	SIP URI	Specifies the contact URI (Uniform Resource Indicator) that SIP Server uses for communication with the MCU. See the URI format and option description on page 334.
oos-check	0–300	(Optional) Specifies how often (in seconds) SIP Server checks a device for out-of-service status. See the option description on page 346.
oos-force	0–30	(Optional) Specifies 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. See the option description on page 347.
prefix	String	(Optional) Specifies the starting digits of the number that are used when sending calls to MCU. See the option description on page 350.

Table 3: Configuring an MCU (Continued)

Option Name	Option Value	Description
recovery-timeout	0–86400 seconds	(Optional) Specifies whether an MCU is taken out of service when an error is encountered, and for how long it is out of service. See the option description on page 353.
service-type	mcu	Set this option to mcu.

5. When you are finished, click Apply.

End of procedure

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

Configuring Endpoints

To configure SIP endpoints, follow the provided procedure using Configuration Manager.

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 Configuration Manager.

Procedure: Configuring endpoints

Start of procedure

1. Under a configured Switch object, select the DNs folder. From the File menu, select New > DN to create a new DN object.
2. In the New DN Properties dialog box, on the General tab, specify the following properties:
   a. 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.
   b. Type: Select Extension (or ACD Position) from the drop-down box.
3. Click the Annex tab.

4. Create a section named `TServer`. In the `TServer` section, create options as specified in Table 4.

Table 4: Configuring Endpoints

Option Name	Option Values	Description
`authenticate-requests`	`register, invite`	Specifies whether incoming SIP requests are treated with an authentication procedure under the following conditions: 1. The name of the incoming SIP message exists in the list of the `authenticate-requests` parameter. 2. The option `password` is configured on the same `DN` object. See the option description on page 330.
`contact`	SIP URI	(Optional, depends on the phone registration) Specifies the contact URI that SIP Server uses for communication with the endpoint. See the URI format and option description on page 334.
`dual-dialog-enabled`	`true, false`	Set the value to `false` for endpoints that accept only one active SIP dialog, or cannot provide remote CTI control by the `NOTIFY` message to answer, hold, or retrieve call operations. Set the value to `false` for Siemens optiPoint phones that are used in re-`INVITE` mode for third-party call control (3pcc) operations. See the option description on page 339.
`geo-location`	String	(Optional, not available for ACD Positions) When selecting a particular extension, SIP Server will consider those DNs whose configured `geo-location` matches the preferred geo-location that is assigned for the call. See the option description on page 341.
`make-call-rfc3725-flow`	`1, 2`	Specifies 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. See the option description on page 344.

Table 4: Configuring Endpoints (Continued)

Option Name	Option Values	Description
password	String	Specifies 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. See the option description on page 349.
recovery-timeout	true, false	Controls whether a device is taken out of service when an error is encountered, and for how long it is out-of-service. See the option description on page 353.
refer-enabled	true, false	Specifies whether the REFER method is sent to an endpoint. The recommended setting is true. See the option description on page 354.
reinvite-requires-hold	true, false	(Optional, for Genesys SIP Endpoints only) Specifies whether the endpoint is placed on hold by re-inviting it with the hold SDP. See the option description on page 354.
request-uri	SIP URI	Specifies 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. See the option description on page 356.
sip-cti-control	talk, hold	Specifies the behavior of the DN representing the SIP endpoint that supports the BroadSoft SIP Extension Event Package. See the option description on page 359.

5. When you are finished, click Apply.

End of procedure

Configuring Gateways

To configure gateways, follow the provided procedure using Configuration Manager.

Procedure: Configuring a gateway

Start of procedure

1. Under a configured Switch object, select the DNs folder. From the File menu, select New > DN to create a new DN object.
2. In the New DN Properties dialog box, on the General tab, specify the following properties:
   a. 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.
   b. Type: Select Trunk from the drop-down box.
3. Click the Annex tab.
4. Create a section named TServer. In the TServer section, create options as specified in Table 5.

Table 5: Configuring a gateway

Option Name	Option Value	Description
contact	SIP URI	Specifies the contact URI that SIP Server uses for communication with the gateway. See the URI format and option description on page 334.
geo-location	String	(Optional) Specifies 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. See the option description on page 341.
oos-check	0–300	(Optional) Specifies how often (in seconds) SIP Server checks a device for out-of-service status. See the option description on page 346.

Table 5: Configuring a gateway (Continued)

Option Name	Option Value	Description
oos-force	0–30	(Optional) Specifies 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. See the option description on page 347.
password	String	(Optional) Specifies the password for gateway registration with the local registrar. This is used for incoming REGISTER requests, not for outgoing INVITE requests.
prefix	String	(Optional) Contains 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	Any non-negative integer	(Optional) Specifies 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	true, false	Specifies whether the REFER method is sent to an endpoint. The recommended setting is true. See the option description on page 354.

Table 5: Configuring a gateway (Continued)

Option Name	Option Value	Description
recovery-timeout	0–86400 seconds	(Optional) Specifies whether a gateway is taken out of service when an error is encountered, and for how long it is out-of-service. See the option description on page 353.
replace-prefix	Digit string	(Optional) Specifies the characters that are inserted in the DN instead of the prefix for the gateway. If this annex is empty or absent, the initial characters that match the prefix option will be removed from the DN. See the option description on page 355.

5. When you are finished, click Apply.

End of procedure

Configuring Music Servers

To configure external music servers, follow the provided procedure using Configuration Manager.

Procedure: Configuring a Music Server

Start of procedure

1. Under a configured Switch object, select the DNs folder. From the File menu, select New > DN to create a new DN object.
2. In the New DN Properties dialog box, on the General tab, specify the following properties:
   a. 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.
   b. Type: Select Voice over IP Service from the drop-down box.
3. Click the Annex tab.
4. Create a section named TServer. In the TServer section, create options as specified in Table 6.

Table 6: Configuring a Music Server

Option Name	Option Values	Description
contact	SIP URI	Specifies the contact URI that SIP Server uses for communication with the music server. See the URI format and option description on page 334.
geo-location	String	(Optional) Specifies 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. See the option description on page 341.
oos-check	0–300	(Optional) Specifies how often (in seconds) SIP Server checks a device for out-of-service status. See the option description on page 346.
oos-force	0–30	(Optional) Specifies 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. See the option description on page 347.
recovery-timeout	0–86400 seconds	(Optional) Specifies whether a music server is taken out of service when an error is encountered, and for how long it is out of service. See the option description on page 353.
request-uri	SIP URI	Specifies 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. See the option description on page 356.
service-type	music or moh	Set this option to music or moh.

5. When you are finished, click Apply.

End of procedure

Configuring Routing Points

To configure routing points, follow the provided procedure using Configuration Manager.

Procedure: Configuring Routing Points

Start of procedure

1. Under a configured Switch object, select the DNs folder. From the File menu, select New > DN to create a new DN object.
2. In the New DN Properties dialog box, on the General tab, specify the following properties:
   a. 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.
   b. Type: Select Routing Point from the drop-down box.

   Note: SIP Server does not support the use of Routing Queue DNs when configuring the SIP Routing Point device.

3. When you are finished, click Apply.

End of procedure

Configuring Softswitches

If you deploy proxies or softswitches between SIP Server and any internal DNs or agent endpoints, configure the proxies or softswitches using the provided procedure in Configuration Manager.

Procedure: Configuring softswitches

Start of procedure

1. Under a configured Switch object, select the DNs folder. From the File menu, select New > DN to create a new DN object.
2. In the New DN Properties dialog box, on the General tab, specify the following properties:
   a. Number: Enter the softswitch server name. This name is currently not used for any messaging, but it must still be unique. Enter a short description for this property.
   b. Type: Select Voice over IP Service from the drop-down box.
3. Click the Annex tab.
4. Create a section named TServer. In the TServer section, create options as specified in Table 7.

Table 7: Configuring Softswitches

Option Name	Option Value	Description
contact	SIP URI	Specifies 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. See the URI format and option description on page 334.
public-contact	IP address	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.
service-type	softswitch	Set this option to softswitch.

5. When you are finished, click Apply.

End of procedure

Additional Information

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.

Configuring an Application Service

To configure an application service, follow the provided procedure using Configuration Manager.

Procedure: Configuring an application service

Start of procedure

1. Under a configured `Switch` object, select the `DNs` folder. From the `File` menu, select `New` > `DN` to create a new `DN` object.
2. In the `New DN Properties` dialog box, on the `General` tab, specify the following properties:
   a. `Number`: Enter the treatment server name.
   b. `Type`: Select `Voice over IP Service` from the drop-down box.
3. Click the `Annex` tab.
4. Create a section named `TServer`. In the `TServer` section, create options as specified in Table 8.

Table 8: Configuring an Application Service

Option Name	Option Values	Description
`contact`	SIP URI	Specifies the contact URI that SIP Server uses for communication with the treatment server. See the URI format and option description on page 334.
`service-type`	`application`	Set this option to `application`.

5. When you are finished, click `Apply`.

End of procedure

Configuring a Recording Service

To configure a recording service, follow the provided procedure using Configuration Manager.

Procedure: Configuring a recording service

Start of procedure

1. Under a configured `Switch` object, select the `DNs` folder. From the `File` menu, select `New` > `DN` to create a new `DN` object.
2. In the `New DN Properties` dialog box, on the `General` tab, specify the following properties:
   a. `Number`: Enter the recorder server name.
   b. `Type`: Select `Voice over IP Service` from the drop-down box.
3. Click the `Annex` tab.
4. Create a section named `TServer`. In the `TServer` section, create options as specified in Table 9.

Table 9: Configuring a Recording Service

Option Name	Option Value	Description
contact	SIP URI	Specifies the contact URI that SIP Server uses for communication with the recorder server. See the URI format and option description on page 334.
request-uri	SIP URI	1. Specifies 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 in the following cases: • As backup, if the recording-filename is wrong or not configured • For emergency call recording, used in conjunction with emergency-recording-filename. 2. *(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. See the option description on page 356.
service-type	recorder	Set this option to recorder.

5. When you are finished, click Apply.

End of procedure

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 (page 288) for more information.

Configuring a Treatment Service

To configure a treatment service, follow the provided procedure using Configuration Manager.

Procedure: Configuring a treatment service

Start of procedure

1. Under a configured Switch object, select the DNs folder. From the File menu, select New > DN to create a new DN object.
2. In the New DN Properties dialog box, on the General tab, specify the following properties:
   a. Number: Enter the treatment server name.
   b. Type: Select Voice over IP Service from the drop-down box.
3. Click the Annex tab.
4. Create a section named TServer. In the TServer section, create options as specified in Table 10.

Table 10: Configuring a Treatment Service

Option Name	Option Values	Description
contact	SIP URI	Specifies the contact URI that SIP Server uses for communication with the treatment server. See the URI format and option description on page 334.
service-type	treatment	Set this option to treatment.

5. When you are finished, click Apply.

End of procedure

Configuring an MSML Service

To configure an MSML service, complete the following procedure using Configuration Manager.

Procedure:
Configuring an MSML Service

Start of procedure

1. Under a configured `Switch` object, select the `DNs` folder. From the `File` menu, select `New` > `DN` to create a new `DN` object.
2. In the `New DN Properties` dialog box, on the `General` tab, specify the following properties:
   a. `Number`: Enter the MSML server name.
   b. `Type`: Select `Voice over IP Service` from the drop-down menu.
3. Click the `Annex` tab.
4. Create a section named `TServer`. In the `TServer` section, create options as specified in Table 11.

Table 11: Configuring an MSML Service

Option Name	Option Values	Description
contact	SIP URI	Set this to the Resource Manager IP address and port. Use the following format: `sip: <RM_IP_address:RM_SIP_port>` Specifies the contact URI that SIP Server uses for communication with the treatment server. See the URI format and option description on page 334.
service-type	msml	Set this option to msml.
prefix	msml=	(Optional) Required for conference and monitoring services only. Set this option to msml=.
subscription-id	Name of SIP Server tenant.	Set this option to the name of the tenant to which SIP Server switch belongs. For single-tenant, set this option to Environment. See the option description on page 365.
Required for Outbound Solution only (to support MakePredictiveCall—Genesys Media Server functionality is not affected)		
refer-enabled	false	Set this option to false. See the option description on page 354.
make-call-rfc3725-flow	1	For MSML service, set this option to 1. Specifies 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. See the option description on page 344.
ring-tone-on-make-call	false	Set this option to false.
cpd-capability	mediaservices	To enable call progress detection using the Genesys Media Server, set this option to mediaservices.

Figure 7 shows a sample configured MSML DN.

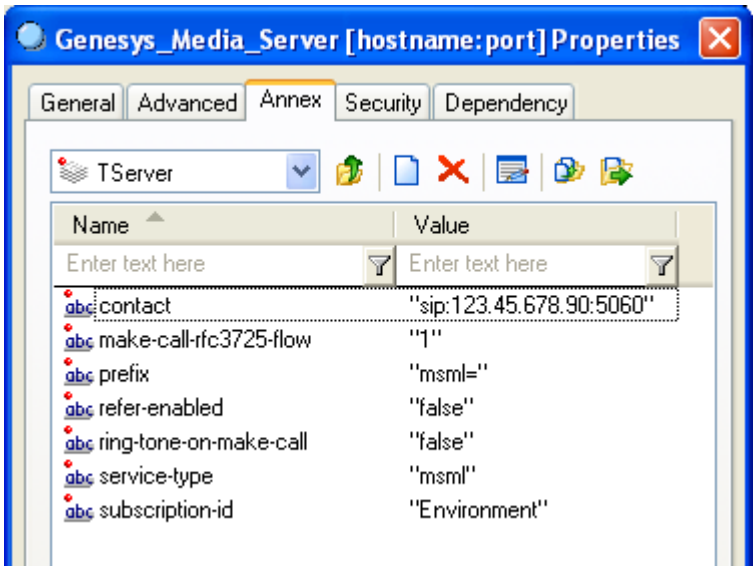

Figure 7: MSML DN

5. When you are finished, click `Apply`.

End of procedure

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 Configuration Manager, SIP Server will generate 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 Stream Manager

Stream Manager is a Genesys client application that streams media files in order to provide announcements and music to callers queued on Routing Points and ACD queues. It can also serve as a music server or as an MCU. SIP Server does not need Stream Manager for normal call handling.

You configure Stream Manager as a `DN` object of type `Voice over IP service`, with the `service-type` option set in the `TServer` section on the `Annex` tab. The values are: `music`, `mcu`, `treatment`, `conference`, and `recorder`.

To use Stream Manager as a music server, it must be configured as described in "Configuring Music Servers" on page 70.

Location of Media Files

Stream Manager must be able to access the audio files that SIP Server requests to play. These files are located in subdirectories of the installed Stream Manager root directory. The files must be in the appropriate codec format, with the filename suffix corresponding to the codec type. All Stream Managers integrated with SIP Server must contain the same announcement/music files in the same directory structure. The treatment will fail if the file does not exist in the specified directory.

Load Balancing of Media Services

When SIP Server performs load balancing of media services across multiple instances of Stream Manager, if a service fails to start at a particular instance of Stream Manager, SIP Server tries the next instance; however, if the service cannot start on any of the instances, the call fails permanently. To avoid looping or network overload, SIP Server does not retry any instance of Stream Manager where the service has previously failed.

Determining Codecs

If you are using a gateway, you must determine the codec it uses before you determine which codec is specified in Genesys.

See the *Framework 7.6 Stream Manager Deployment Guide* for more information about Stream Manager.

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, and so on—using the Media Server Markup Language (MSML). 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.

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 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.

For more information about the Media Server, see the *Genesys Media Server 8.1 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

The following table describes the basic steps required to integrate SIP Server with the Genesys Media Server for MSML service.

Task Summary: Integrating SIP Server and GVP Media Server

Objective	Key Procedures and Actions
1. Configure SIP Server for MSML.	In the `TServer` section of the SIP Server Application object, configure the following option: • `msml-support`—Set this option to `true`.

Task Summary: Integrating SIP Server and GVP Media Server (Continued)

Objective	Key Procedures and Actions
2. Configure the MSML DN.	1. Create a `Voice over IP Service` DN. 2. In the `TServer` section, configure the following options: • `contact`—Set to the RM IP address and port. • `prefix`—Set to `msml=` *(Required for conferencing and monitoring only)* • `service-types`—Set to `msml`. • `subscription-id`—Set to the name of tenant. For a detailed procedure, see Procedure: Configuring an MSML Service, on page 78.
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 8.1 Deployment Guide*.
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 to `round-robin`. • `monitor-method`—Set to `option`. • `port-usage-type`—Set to `in-and-out`. • `resource-confmaxsize`—Set to `-1`. • `service-types`—Ensure that `msml` is included in the list of service-types. For more information, see the *Genesys Media Server 8.1 Deployment Guide*.

Task Summary: Integrating SIP Server and GVP Media Server (Continued)

Objective	Key Procedures and Actions
5. Ensure MCP is connected to Resource Manager.	Ensure that the MCP instances are included on the `Connections` tab of the Resource Manager Application object. For more information, see the *Genesys Media Server 8.1 Deployment Guide*
6. Configure Resource Manager application.	• In the `rm` section, configure the following parameter: ◦ `rm.conference-sip-error-respcode`—Set to `503`. ◦ `rm.resource-available-respcode`—Set to `603`. • In the monitor section, configure the following parameter: ◦ `sip.proxy.realeaseconfonfailure`—Set to `false`.
7. 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. On the `Annex` tab of the IVR Profile, create a `gvp.general` section, adding the following options: ◦ `service-type`—Set to `voicexml`. For more information, see the *Genesys Media Server 8.1 Deployment Guide*.
8. Configure the Tenant object.	Assign the IVR Profile as the default for your tenant. 1. Right-click your `Tenant` object, then select `Properties`. 2. On the `Annex` tab of the `Properties` window, 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 8.1 Deployment Guide*.

Chapter

5 SIP Server Feature Support

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

- Alternate Routing, page 86
- Application Failure Detection, page 92
- Associating an ACD Queue with a Routing Point, page 93
- Asterisk Voice Mail Integration, page 94
- Automatic Agent Logout, page 97
- Call Recording, page 97
- Call Supervision, page 101
- Call Transfer and Conference, page 120
- Class of Service, page 126
- Contact Header Handling Options, page 137
- Control of SIP Response Code from within a Routing Strategy, page 139
- Dial Plan, page 141
- DTMF Tones Generation, page 156
- Dummy SDP, page 157
- Early Media for Inbound Calls, page 160
- Emulated Agents, page 162
- Endpoint Service Monitoring, page 167
- IMS Integration, page 169
- Instant Messaging, page 171
- Mapping SIP Headers and SDP Messages, page 177
- Media Server Reliability, page 183
- Music and Announcements, page 186
- Nailed-Up Connection, page 189
- Network Asserted Identity, page 192
- No-Answer Supervision, page 200
- Outbound IP Solution Integration, page 203

- Overload Control, page 207
- Personal Greeting, page 211
- Playing Multiple Media Files, page 214
- Presence from Switches and Endpoints, page 216
- Preview Interactions, page 226
- Private Headers, page 227
- Providing Call Participant Info, page 227
- Providing a Caller ID, page 228
- Quality of Service, page 228
- Remote Media on Genesys SIP Endpoint SDK 8.0, page 229
- Remote Server Registration, page 230
- Remote Talk, page 230
- SIP Traffic Monitoring, page 231
- Transport Layer Security for SIP Traffic, page 232
- Trunk Capacity Support, page 235
- Tromboning Control, page 235
- Trunk Optimization for Multi-Site Transfers, page 238
- Video Support, page 242
- Working with Multiple Devices, page 247
- Genesys Voice Platform Integration, page 253

Alternate Routing

SIP Server supports call delivery to a variety of alternate default-DN locations in order to handle complications that can arise during the regular processing of inbound calls. These alternate routing scenarios include the following:

- "Alternate Routing for Stranded Calls"
- "Alternate Routing for Unresponsive DNs"
- "Alternate Routing for Unresponsive URS"
- "Alternate Routing for Calls to an External Destination"

Alternate Routing for Stranded Calls

SIP Server now 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`s, 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

The following task summary provides an overview of the main steps required to configure stranded call routing.

Task Summary: Configuring Stranded Call Routing

Objective	Related Procedures and Actions
• Configure for all queues.	Go to SIP Server `Application` object > `Options` tab > `TServer` section. • `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 88. • `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.
• Configure for individual queues.	Go to SIP Server switch > DNs folder > individual `ACD Queue` DN > `Annex` tab > `TServer` section. • `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

The following table describes the valid values for the `stranded-call-overflow` and `stranded-on-arrival-calls-overflow` options, as well as their related SIP Server actions.

Table 12: 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 12: 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-call-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.

Feature Configuration

The following table shows the configuration needed to enable this feature.

Task Summary: Configuring Default Routing for Unresponsive DN

Objective	Key Procedures and Actions
• Configure the SIP Server Application.	Go to SIP Server `Application` object > `Options` tab > `TServer` section and configure the following options. • `sip-invite-timeout`—Specifies the number of seconds that SIP Server will wait 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 do not change this option without considering all the scenarios it may affect. For details, see option description on page 315.
• Configure the DN.	Go to SIP Server `switch` > `DNs` folder > individual DN > `Annex` tab > `TServer` section and configure the following option. • `no-response-dn`—Enter the DN where SIP Server will send the timed-out call. For details, see option description on page 305

Alternate Routing for Unresponsive URS

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

Feature Configuration

The following table shows the configuration needed to enable this feature.

Task Summary: Configuring Default DN for Unresponsive URS

Objective	Key Procedures and Actions
• Configure for all Routing Points.	Go to SIP Server `Application` object > `Options` tab > `TServer` section and configure the following options: 1. `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. 2. `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`. For details, see the option descriptions on pages 286 and 311.
• Configure for individual Routing Point.	Go to SIP Server switch > DNs folder > individual `Routing Point` DN > `Annex` tab > `TServer` section and configure the following option. • `default-dn`—Enter the alternate DN to be used for URS failures/timeout on this Routing Point only.

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, when 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

The following table shows the configuration needed to enable this feature.

Task Summary: Configuring Default DN for External Destinations

Objective	Key Procedures and Actions
• Configure the SIP Server Application.	Go to SIP Server `Application` object > `Options` tab > `TServer` section and 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. For more information, see the option description `default-route-point` (page 286).

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 both 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 31.

How Failure Detection Works

1. With failure detection enabled, the Management Layer monitors one or both of the following:
   - Heartbeat messages (UDP packets) sent by the SIP Server application to the LCA.
   - Heartbeat messages from individual registered SIP Server threads.
2. If the LCA discovers that the application or one of its threads 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

The following table shows the configuration needed to enable this feature.

Task Summary: Configuring SIP Server for Hang Up Detection

Objective	Key Procedures and Actions
• Configure the SIP Server Application.	Go to SIP Server `Application` object > `Options` tab > `sml` section and configure the following option. 1. `heartbeat-period`—Set this option to the length of time, in seconds, that Management Layer will wait before taking corrective action. 2. `hangup-restart`—Set this to `true` to restart SIP Server in case it becomes unresponsive, `false` to send a notification only. For details, see the option descriptions on pages 466 and 467.

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 Configuration Manager.

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.

Asterisk Voice Mail Integration

SIP Server exchanges SIP messaging with an Asterisk-based Voice Mail system using a SIP trunk. Voice mail boxes with Asterisk have the same names as their corresponding DNs, but you do not need to configure voice mail boxes as DNs in Configuration Manager. You can access these voice mail boxes by dialing a user-defined prefix and a DN number. It is also possible to dial a prefix and a special extension so that the call will be delivered to the Voice Mail Main Menu.

SIP Server configuration options are used to organize the delivery of calls to the Voice Mail system when certain conditions are met. For example, when an incoming call is not answered within a specified timeout. Universal Routing Server strategies are used to organize more flexible control over delivery of the call to a voice mail box.

Note: Voice mail for DNs is handled by an Asterisk softswitch when using Asterisk as a softswitch with SIP Server. You do not need to configure voice mail and Message Waiting Indicator (MWI) in SIP Server for any DNs. However, voice mail is configured for agents and Agent Groups in this scenario.

Genesys Voice Mail Configuration Adapter for Asterisk

The Genesys Voice Mail Configuration Adapter for Asterisk (GVMA) is a utility that allows you to synchronize the Asterisk Voice Mail configuration file (`voicemail.conf`) with data from the Configuration Server. By default, GVMA creates a voice mail box in Asterisk for each extension DN configured in Configuration Manager. GVMA can also be configured to use DNs of other types for the same purpose.

GVMA obtains the necessary Asterisk Voice Mail configuration properties from the following objects in Configuration Manager:

- DNs are used to configure voice mail boxes for extensions.
- Agent Logins are used to configure voice mail boxes for agents.
- Agent Groups are used to configure of voice mail boxes for agent groups.

When it has retrieved this information, GVMA performs the following tasks:

1. Connects to Configuration Server.
2. Makes a backup copy of the current Asterisk configuration.
3. Loads DNs from Configuration Server.
4. Updates the Asterisk Voice Mail configuration file based on the retrieved data.
5. Instructs Asterisk to reload configuration files.

GVMA is run manually or scheduled to run periodically using your operating system's scheduling tools.

Message Waiting Indicator Functionality

SIP Server supports Message Waiting Indicator (MWI) notifications by registering with Asterisk and processing MWI notifications sent by Asterisk within `NOTIFY` messages. SIP Server sends to a DN an `EventUserEvent` message with detailed information about the message waiting status attached in the `UserData` fields.

MWI Processing for Extension Voice Mail Boxes

SIP Server registers (using the `REGISTER` request) all extensions as voice mail boxes with Asterisk when the option `mwi-extension-enable` (see page 303) has a value of `true`. Asterisk sends `NOTIFY` SIP messages to SIP Server that contain MWI information. SIP Server stores MWI information received from Asterisk, and distributes an `EventUserEvent` message with the latest MWI information to each registered extension voice mail box.

MWI Processing for Agent Voice Mail Boxes

SIP Server registers each agent voice mail box with Asterisk immediately after the `EventAgentLogin` message is received and when the option `mwi-agent-enable` (see page 303) has a value of `true`. SIP Server stores MWI information received from Asterisk for each agent voice mail box and distributes an `EventUserEvent` message with the latest MWI information to the DN that the agent has logged into. SIP Server will terminate this registration immediately after an `EventAgentLogout` message is received.

MWI Processing for Agent Group Voice Mail Boxes

SIP Server registers the agent group voice mail box with Asterisk immediately after the first member's `EventAgentLogin` message is received and when the option `mwi-group-enable` (see page 304) has a value of `true`. SIP Server stores MWI information received from Asterisk for each agent group voice mail box and distributes an `EventUserEvent` message with the latest MWI information to all DNs associated with agents that are logged in and configured to use this voice mail box. SIP Server will terminate this registration immediately after an `EventAgentLogout` message is received from the last agent associated with this agent group's voice mail box.

Note: SIP Server associates an agent with an agent group voice mail box by using the option `gvm_group_mailbox` located in the `TServer` section on the `Annex` tab of the `Agent Login` configuration object. Any agent configuration object that is configured with the same voice mail box specified in this option is associated with this agent group's voice mail box.

Feature Configuration

MWI functionality is configured on the Asterisk side and on the SIP Server side. The following SIP Server configuration options support MWI functionality:

- `mwi-host` (page 304)
- `mwi-port` (page 304)
- `mwi-domain` (page 303)
- `mwi-extension-enable` (page 303)
- `mwi-agent-enable` (page 303)
- `mwi-group-enable` (page 304)
- `mwi-mode` (page 304) (for backward compatibility)

For Asterisk configuration and integration with SIP Server, see the *Framework 7.6 SIP Server Integration Reference Manual.*

Automatic Agent Logout

SIP Server can now automatically log out an agent after a specified period of inactivity, ensuring 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 agent state.
- Calls that are made or received at the DN from which the agent is logged on (SIP Server will not log out an agent who is currently on a call).

Feature Configuration

The following task summary describes the configuration steps that are required to enable the automatic agent-logout feature. For highest priority, set the following options in the `Agent Login` DN.

Task Summary: 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. For details, see the option description on page 282.
2. (Optional) Enable a stricter logout policy.	In the `TServer` section of the applicable configuration object, set the `auto-logout-ready` option to `true`. For details, see the option description on page 282.

Call Recording

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

Regular Call Recording

Call recording is performed by passing an RTP stream through Stream Manager. Stream Manager acts as a media stream proxy, recording all media packets into a file. Depending on the configuration, Stream Manager may perform media mixing, or it may save the RTP packets as is, thus improving call recording performance. (See the *Framework 7.6 Stream Manager*

Deployment Guide for details.) Call recording is always enabled on a single call leg, such as a leg with a gateway or a leg with 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 TEvents.

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

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 that first signals `EventEstablished`.

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 into the same file. The actual Real-Time Transport Protocol (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

The following task summary provides an overview of the main steps that are required to configure the call-recording functionality.

Task Summary: Configuring Call Recording

Objective	Related Procedures and Actions
1. Configure a DN.	To enable call recording on a particular DN, in the `TServer` section on the `Annex` tab of the `DN` object in Configuration Manager, 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` object.	In the `TServer` section on the `Options` tab of the SIP Server `Application` object in Configuration Manager, 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`.

Task Summary: Configuring Call Recording (Continued)

Objective	Related Procedures and Actions
3. Configure an `Extension` attribute.	Specify an `Extension` attribute with a `record` key in the `TRouteCall` request. See the key values in Table 28, "Use of the Extensions Attribute," on page 264. The routing strategy will determine whether call recording is needed.
4. Configure a Stream Manager `Application` object.	Set configuration options in the Stream Manager `Application` object for precise control of how recording is performed. See the *Framework 7.6 Stream Manager Deployment Guide* for more information. It is recommended to use `pcap` recording mode on Stream Manager for best performance.
5. Configure a recording service.	Configure a DN of type `Voice over IP Service` with the following configuration options: • `contact`: Set to the device's IP address used for recording. • `request-uri`: Set to the SIP URI. • `service-type`: Set to `recorder`. See "Configuring a recording service" on page 75 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 the `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 Configuration Manager. See "Configuring a Recording Service" on page 75 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 288.

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

Note: Refer to the *SIP Server 7.5.0 Call Recording White Paper* for more information about call recording. This document is available from Genesys Technical Support or Genesys Professional Services.

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 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 109.
- 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 112.

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.

  Note: SIP Server does not monitor consultation calls that are made either from or to a DN that is under call supervision.

- *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 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 him or her.

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 scopes available in SIP Server:

- *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 want 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 "Feature Configuration" on page 104 for more information.

A monitoring session begins when a supervisor joins a call, and it 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.

Feature Configuration

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

- Subscription, page 104
- Assistance Request, page 106
- Supervisor Auto-release, page 106
- Hiding Supervisor Presence, page 107
- Configuration Options, page 108

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 in order 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 the 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 Configuration Manager.

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 264 for more information.

Note: SIP Server identifies the agent for the call supervision 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 102).

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 the 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 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. The T-Library desktop applications used by call center employees must be able

to process the T-Library events and indicate the 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 make 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 7 Events and Models Reference Manual*.

Call Supervision For Transferred Calls

SIP Server supports supervisor monitoring for DNs that are moved from a consultation call to the main call as a result of a two-step transfer. After receiving the `TCompleteTransfer` request from the party that initiated the transfer, SIP Server can then invite the supervisor to a new monitoring session for the targeted DN, provided that the supervisor has an active subscription to monitor the DN.

Notes:
- This feature applies to two-step *transfers* only. Call supervision after 2-step conferences is not supported.
- This feature is restricted. If you need to enable call supervision for transferred calls, please contact your Technical Support representative for advice.

Configuration Options

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

- `cancel-monitor-on-disconnect` (page 284)
- `default-monitor-mode` (page 286)
- `default-monitor-scope` (page 287)
- `intrusion-enabled` (page 300)
- `monitor-internal-calls` (page 301)

Feature Limitations

The following known limitations currently apply to call supervision:

- SIP Server does not monitor consultation calls that are made either from or to a DN that is under call supervision. No monitoring is activated in this scenario, and the supervisor will not be invited to monitor the agent.
- Call supervision functionality is disabled for video calls.
- A supervisor participating in a monitoring session cannot initiate a 1pcc or 3pcc call transfers or conference calls because it can change the supervisor's status in the conference call.
- 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 for the next call on the targeted DN.
- Call supervision functionality is supported only when Stream Manager is used as an MCU. Stream Manager is required because SIP Server sends proprietary information in the SIP messages to set up a specific conference mode on the Stream Manager.
- 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.

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.

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 391). 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 110.)

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.

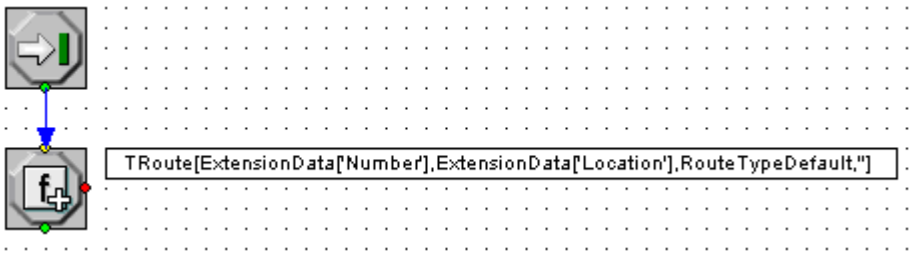

Figure 8: A Routing Strategy Design Sample

The sample strategy uses a single the `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.

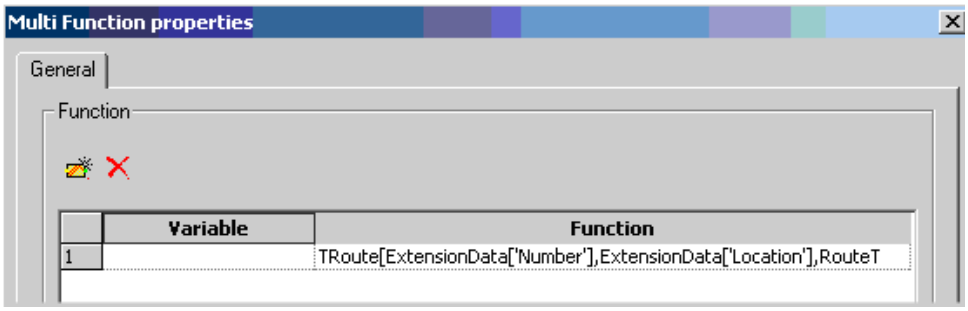

Figure 9: The Multi Function Object

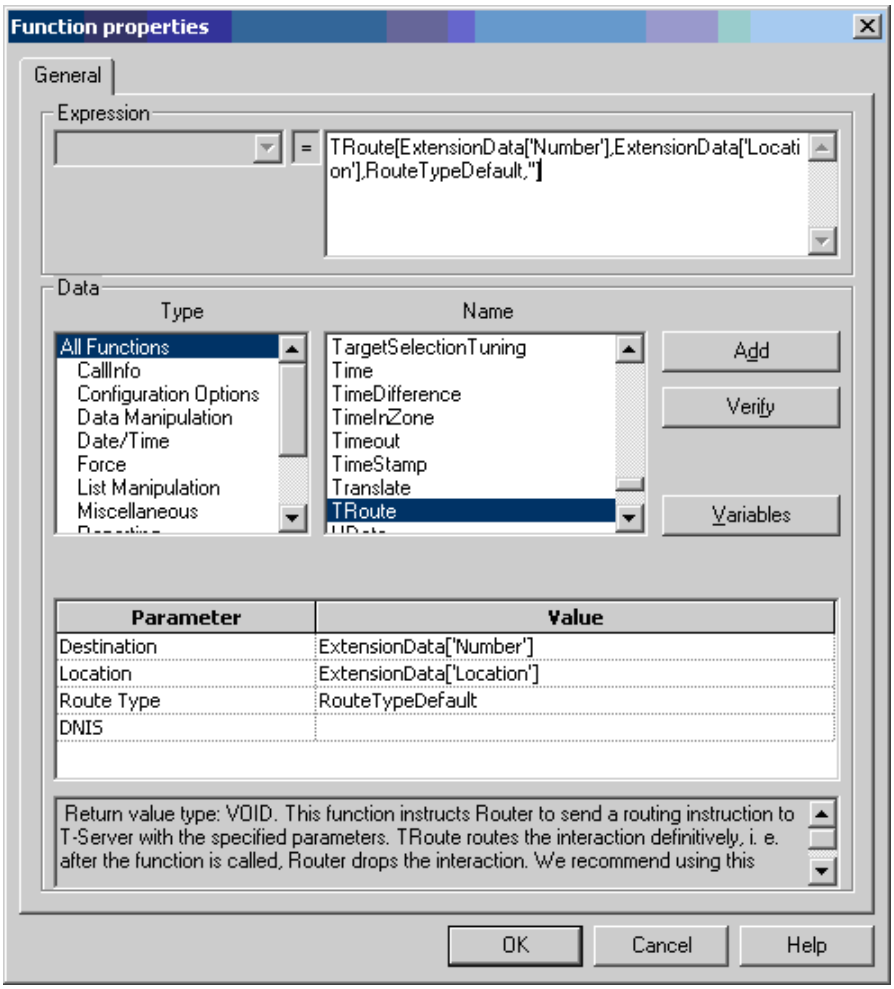

Figure 10: The Function Properties

Feature Limitations

The following known limitation currently applies 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.

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 that 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 101.

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 Configuration Manager, select the SIP Server `Application` object and add the `parking-music` option in the `TServer` section on the `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 the 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. In this sample configuration, Stream Manager is used for announcement service and caller input collection.
   a. Prepare the voice prompts in 8 KHz mono audio format, encoded with one of the codecs, supported by Stream Manager.
   b. Assign a numeric ID, unique within the Stream Manager's `announcement` directory, to each announcement.
   c. Name each recording file using the following format:

      `<ID>_<codec_suffix>.wav`

      Refer to the *Stream Manager 7.6 Deployment Guide* for the list of supported codecs and corresponding filename suffixes.

      For efficiency, to avoid real-time trans-coding, you may also use the SMzip utility, supplied with Stream Manager to trans-code your prompts off-line into zip files, containing the prompts encoded with all supported audio codecs.

      To trans-code each prompt, use a command similar to the following:

      ```
      >smzip -ac all 9000.zip 9000_pcmu.wav
      ```

d. Place the resulting audio files (either `.wav` or `.zip`) into the Stream Manager's `announcement` directory. If you use multiple Stream Managers, make sure that the files are replicated through all Stream Managers' `announcement` directories.

The following sample strategy collects the caller's Agent Login and password, requests the monitoring target number, and then initiates multiple-call monitoring, using the collected caller's input.

The strategy uses the following prompts.

Table 13: Strategy Prompts

Prompt text	ID	File Name
Welcome to the Sample Remote Supervision.	9000	9000_pcmu.wav
Please enter your Agent login.	9010	9010_pcmu.wav
Please enter your password.	9020	9020_pcmu.wav
Please enter the monitoring target number.	9030	9030_pcmu.wav
An error occurred while processing your request.	9090	9090_pcmu.wav

4. In Interaction Routing Designer (IRD), design your routing strategy. See "Routing Strategy Design Sample" on page 114.
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.*

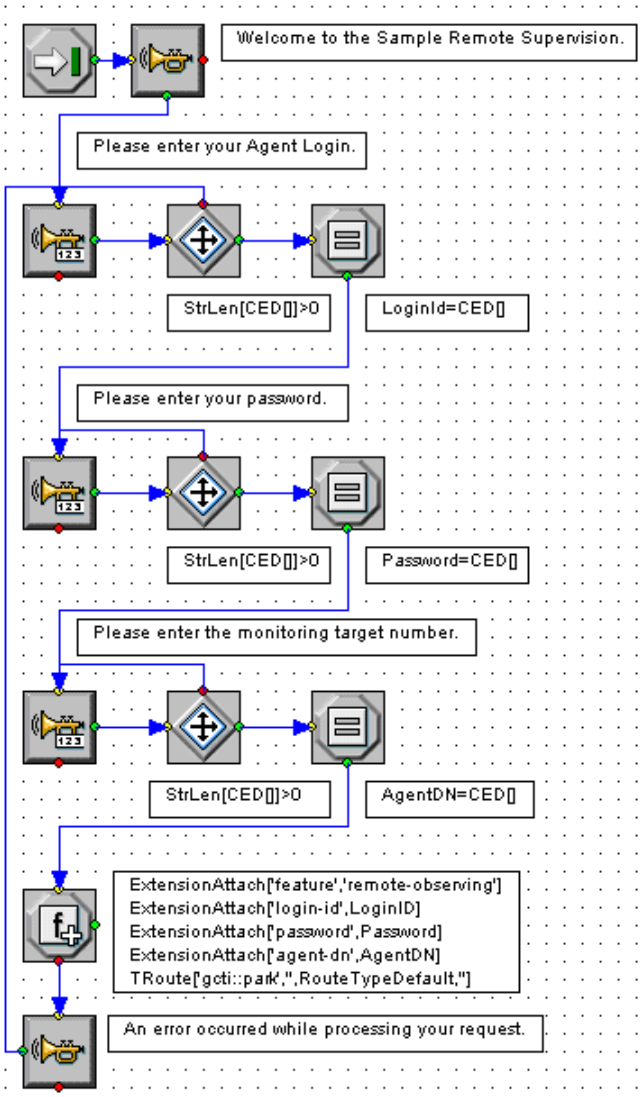

Figure 11: A Routing Strategy Design Sample

Initial Greeting

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

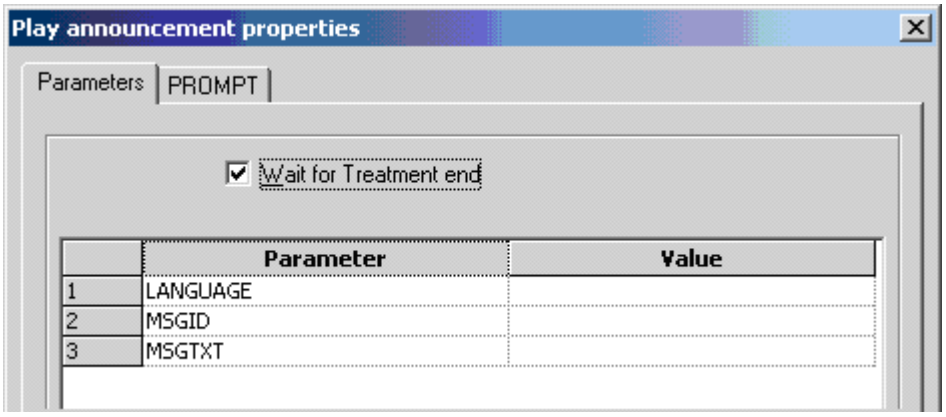

Figure 12: Initial Greeting: Parameters Tab

The `LANGUAGE`, `MSGID` and `MSGTXT` parameters are not used by the 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.

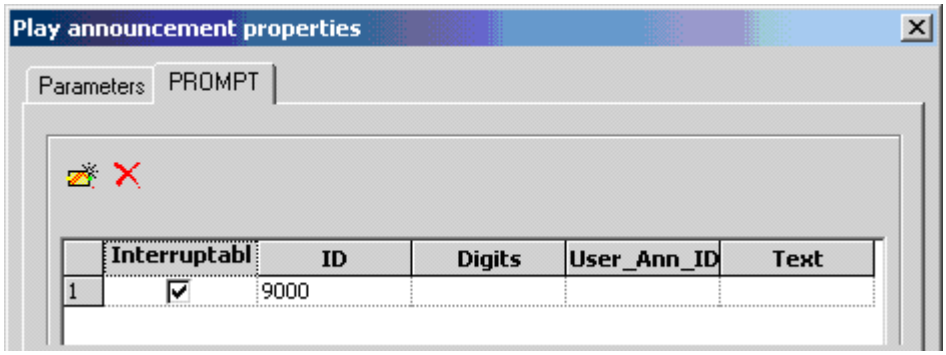

Figure 13: Initial Greeting: PROMPT Tab

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 that they could 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.

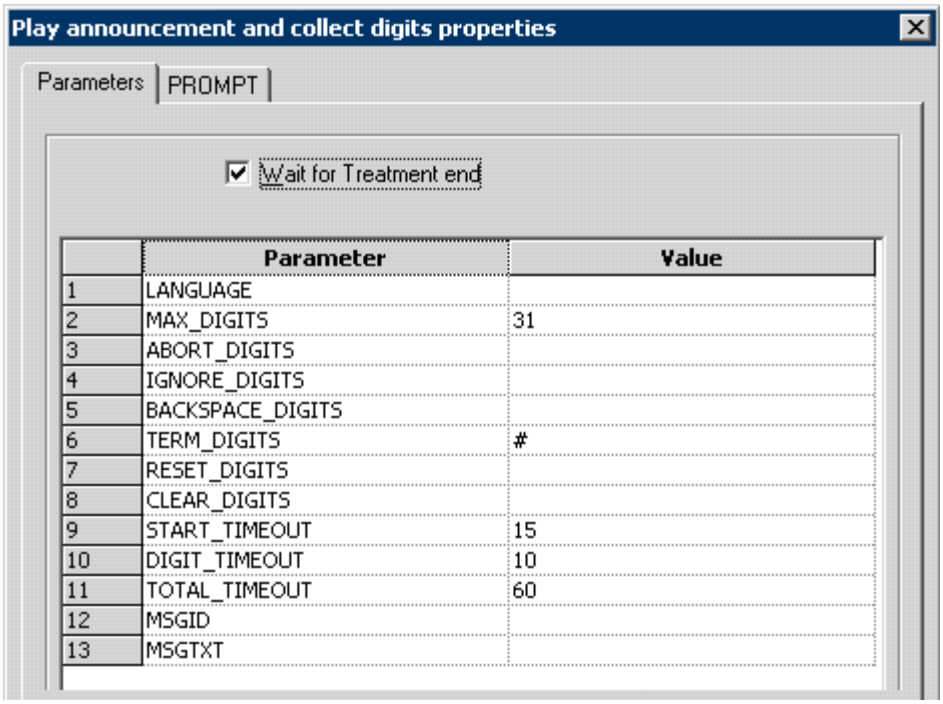

Figure 14: Collecting Digits: Parameters Tab

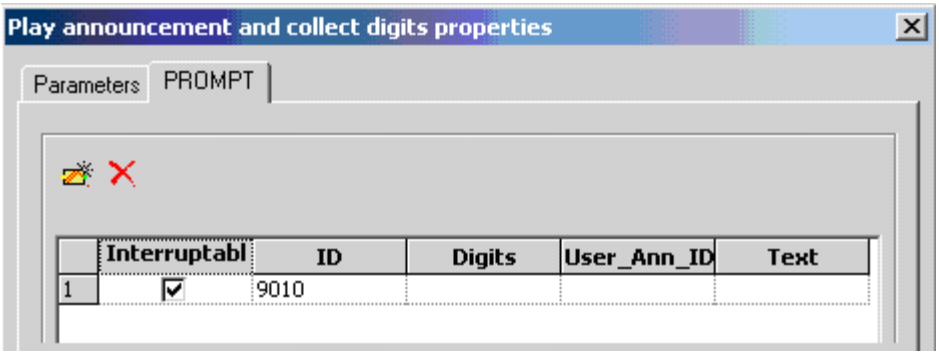

Figure 15: Collecting Digits: PROMPT Tab

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).

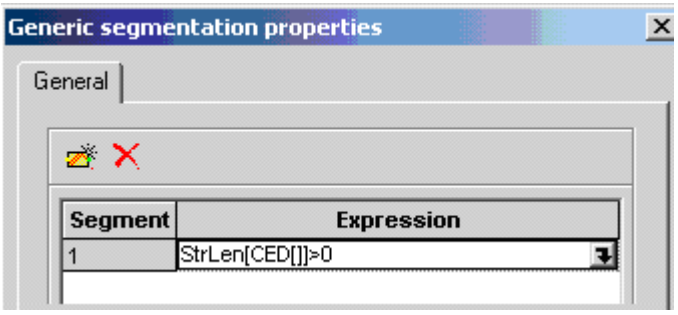

Figure 16: The Generic Segmentation Object

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).

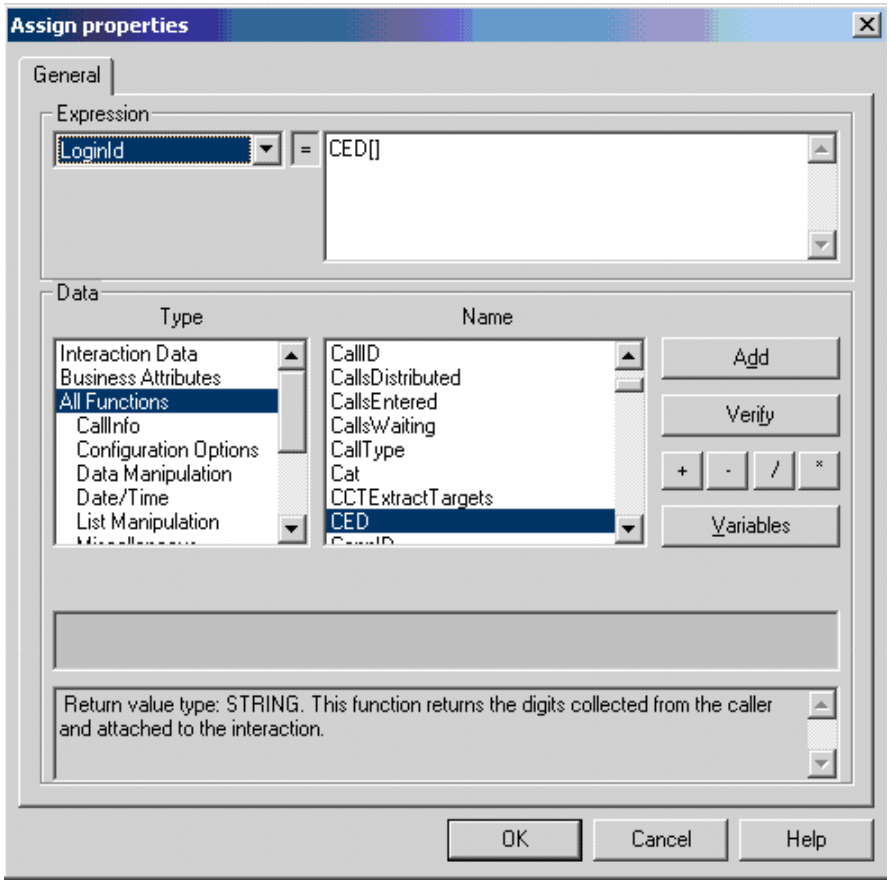

Figure 17: The Assign Object

Collecting Password and Monitoring Target

Subsequent routing strategy blocks prompt the caller to enter a password and the desired monitoring target DN number, an 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).

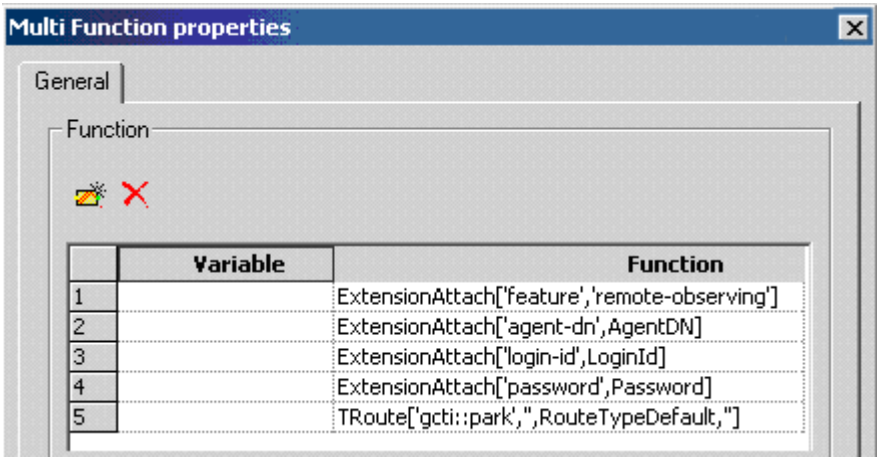

Figure 18: The Multi Function Object

The `Multi Function` object attaches the extensions parameters to the `TRouteCall` request, as the request extensions, and 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 return 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, ACD Queue, 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.

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.

Note: First-party call control calls are not supported when SIP Server is integrated with the Siemens HiPath 8000 switch.

Starting with release 7.5, 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 (see page 347).

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 (page 348), 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, and then the call is routed to an external destination.

- An agent transfers an inbound call by using the blind transfer to a Routing Point, and 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.

Feature Configuration

The following task summary describes the configuration steps required to enable routing of inbound calls to an external destination using the REFER.

Task Summary: 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 on the Annex tab: • oosp-transfer-enabled—Set this to true. • refer-enabled—Set this to false. This ensures that 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.

Feature Configuration

The following task summary describes the configuration steps required to enable routing of outbound calls to an external destination using the REFER.

Task Summary: 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 on the Annex tab: • oosp-transfer-enabled—Set this to true. • refer-enabled—Set this to true.

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 (see page 354). 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 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.

Table 14 describes how these two options control the SIP methods used.

Table 14: Trunk DN Configuration for External Transfers

refer-enabled	oosp-transfer-enabled	SingleStepTransfer /Route	2StepTransfer
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 conferences with central mixing, using MCUs that support more than three participants.

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 (see page 302) must be set to the valid name of the silent audio file to be played in applicable conferences (more than two active participants).

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 now 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 method or the SIP REFER method 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

The following task summary describes the configuration that is required to enable TCompleteTransfer operations using REFER.

Task Summary: Enabling TCompleteTransfer by REFER

Objective	Related Procedures and Actions
Configure the Trunk DN.	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. For more information about these options, see the descriptions on page 367 and page 362.

Feature Limitations

The following known limitations currently apply to call transfer and conference scenarios:

- 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-inter-trunk` option is set to true.

Class of Service

Class of Service (COS) is the functionality that defines telephony capabilities for a device or an agent. In SIP Server, COS telephony capabilities are defined by configuring the following:

- Outbound dialing rules (deprecated in release 8.0.2; Genesys recommends using the "Dial Plan" feature instead)
- 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.

Outbound Dialing Rules

Outbound dialing rules define whether an agent or device is able to make outbound calls, and they also translate dialed numbers according to the specified patterns.

Note: Starting in release 8.0.2, the out-rule functionality provided by the Class of Service feature is deprecated, replaced by the Dial Plan feature. For SIP Server versions 8.0.2 and above, Genesys recommends using the Dial Plan feature for configuring all outbound dial plan rules.

For more information, see "Dial Plan" on page 141.

An outbound dialing rule is specified as a pair of patterns. The first pattern, `in-pattern`, is used to match the dialed number. The second pattern, `out-pattern`, defines how to transform the dialed number, when matched, to the `in-pattern`.

It is possible to specify several outbound dialing rules within a single COS definition.

When an outbound call is initiated either from a device (1pcc) or from a softphone (3pcc), SIP Server attempts to match the dialed number with one of the outbound dialing rules. If the matching rule is not found, SIP Server will proceed with the outbound call as it is dialed. If the matching rule is found (that is, `in-pattern` in one of the outbound dialing rules matched the dialed number), SIP Server will apply this rule to the call according to the following procedure:

1. `out-pattern` from the outbound dialing rule is applied to the dialed number to produce the resulting dialed number.
2. If the resulting dialed number is empty, SIP Server rejects an attempt to make the outbound call.
3. If the resulting dialed number is not empty, SIP Server attempts to place the outbound call to the resulting dialed number.

If there are multiple outbound dialing rules configured within a single COS, and `in-pattern` in more than one rule matches the dialed number, SIP Server will choose the outbound dialing rule with the most matched digits in it.

Example

Here is an example of the outbound dialing rule:

```
in-pattern=411;out-pattern=9411
```

Dialing Rule Format

Dialing rules must conform to the following syntax, represented using Augmented Backus-Naur Form (ABNF) notation.

```
dialing-plan-rule      = [name] in-pattern [out-pattern]
name                   = *( ALPHA / DIGIT / "-")
in-pattern             = 1*(digit-part / abstract-group)
out-pattern            = 1*(symbol-part / group-identifier) *param-part
digit-part             = digits / range / sequence
symbol-part            = digits / symbols
range                  = "[" digits "-"  digits "]" group-identifier
sequence               = "[" 1*(digits [","] ) "]" group-identifier
abstract-group         = fixed-length-group / flexible-length-group
fixed-length-group     = 1*group-identifier
flexible-length-group  = "*"  group-identifier
param-part             = ";" param-name "=" param-value
param-name             = "ext" / "phone-context" / "dn"
param-value            = 1*ANYSYMBOL
group-identifier       = ALPHA
digits                 = 1*DIGIT
symbols                = 1*("-" / "+" / ")" / "(" / ".")
```

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-rule = [name] in-pattern [out-pattern]`

  Where:

  - `[name]` is the name of 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 SIP Server, a name is required.
  - `in-pattern` is the part of the rule to which SIP Server looks when attempting to match the input number.
  - `[out-pattern]` is the part of the rule that instructs SIP Server 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. SIP Server uses this when selecting the most appropriate rule from the entire dial plan.
  - `abstract-group` represents one or more letters with each letter representing one or more numbers. SIP 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`. Remember that 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 SIP 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. SIP 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 SIP 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

  Where:
  - fixed-length-group specifies a group composed of a specific number of digits, and determined by the number of 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 SIP Server how many digits belong in that group. For example, rule-04 (see page 132) 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.

  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.

- `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"` represents the 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`
- `digits = 1*DIGIT`
- `symbols = 1*("-" / "+" / ")" / "(" / ".")`

Transformation

The transformation algorithm consists of two phases:

1. Input number parsing based on `in-pattern`.
2. Output number construction based on `out-pattern` and the parsed input string.

The first phase consists of selecting all group elements from `in-pattern` (the `abstract-group`, `range`, and `choice` parameters), and assigning them a corresponding group of digits from the input number. There are three kinds of such assignment:

- Explicit—For ranges, sets, and fixed-length-groups where the length and position of the group is set explicitly in `in-pattern`.
- Positional—For flexible-length-groups; the digits are selected exclusively based on their position with respect to other groups.
- Analytical—For entities; the digits are selected based on some kind of analytical processing.

Examples

These are examples of how SIP Server applies the rules configured above to various input numbers.

Example 1

Table 15: Class of Service

Input Number	in-pattern	out-pattern	Output Number
914159131472	9*A	+A	+14159131472
011441581234567	011#CABBB*D	+A-(B)-D	+44-(158)-1234567
1472	[1-8]ABBB	AB	1472
911	AAA	9A	9911
14159131472	*A9131472	80409131472	80409131472
16503570622	1[415,650]A*B	B	3570622

Example 2

```
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
```

rule-01 SIP Server receives input number `2326`.

As a result of the rule selection process, SIP 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, SIP Server detects two groups: Group `A = 2` and Group `B = 326`.

SIP Server formats the output string as `2326`.

rule-02 SIP Server receives input number `9122`.

As a result of the rule selection process, SIP 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, SIP Server detects one group: Group `A = 9122`.

SIP Server formats the output string as `9122`.

rule-03 SIP Server receives input number `16503222332`.

As a result of the rule selection process, SIP 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, SIP Server detects two groups: Group `A = 650` and Group `B = 3222332`.

SIP Server formats the output string as `3222332`.

rule-04 SIP Server receives input number `19253227676`.

As a result of the rule selection process, SIP 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, SIP Server detects three groups: Group `A = 925`, Group `B = 322`, and Group `C = 7676`.

SIP Server formats the output string as `919253227676`.

rule-05 SIP Server receives input number `4089137676`.

As a result of the rule selection process, SIP 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, SIP Server detects two groups: Group `A = 408` and Group `B = 7676`.

SIP Server formats the output string as `804079137676`.

rule-06 SIP Server receives input number `011441112223333`.

As a result of the rule selection process, SIP 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, SIP Server detects two groups: Group `A = 44` and Group `B = 1112223333`.

SIP Server formats the output string as `9011441112223333`.

Call Rejection by Out-Dialing Rules

A call attempt can be rejected by the out-dialing rules. To indicate this condition, SIP Server generates an `EventError` response to the corresponding request, with the reason code `Invalid Destination DN (415)`.

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` (page 355) 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` (page 355) 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` response to the corresponding request, with the reason code `Invalid Destination DN (93)`.

Feature Configuration

The following task summary provides an overview of the main steps required to configure Class of Service.

Task Summary: Configuring Class of Service

Objective	Related Procedures and Actions
1. Configure a COS DN.	Configure a DN of type `Voice over IP Service` to represent the COS entity itself. This DN should have outbound dialing rules and ring-through rules specified in the options. See "Configuring a COS DN" on page 135.
2. Assign COS to a device.	Assign the COS DN to one or multiple DNs of type `Extension` or `ACD Position` within the same `Switch` configuration object. See "Assigning COS to a Device" on page 136
3. Assign COS to an agent.	Assign the COS DN to one or multiple Agent Logins. See "Assigning COS to an Agent" on page 136.

Procedure: Configuring a COS DN

Start of procedure

1. Create a COS DN under the SIP Server `Switch` object with a type of `Voice over IP Service`.
2. On the `Annex` tab of the COS `DN` object, in the `TServer` section, set the configuration option `service-type` to `cos`.
3. On the `Annex` tab of the COS `DN` object, in the `TServer` section, specify the outbound dialing rules. Each rule is defined using the configuration option named `out-rule-<n>` (page 348)—for example:

   ```
   out-rule-1 = in-pattern=411;out-pattern=9411
   ```

   Rules must be numbered sequentially—for example, `out-rule-1`, `out-rule-2`, and so on.
4. On the `Annex` tab of the COS `DN` object, in the `TServer` section, specify the ring-through rules—for example:

   ```
   reject-call-incall = true
   reject-call-notready = true
   ```

   Figure 19 illustrates a sample configuration for the COS DN with outbound dialing rules and ring-through rules specified.

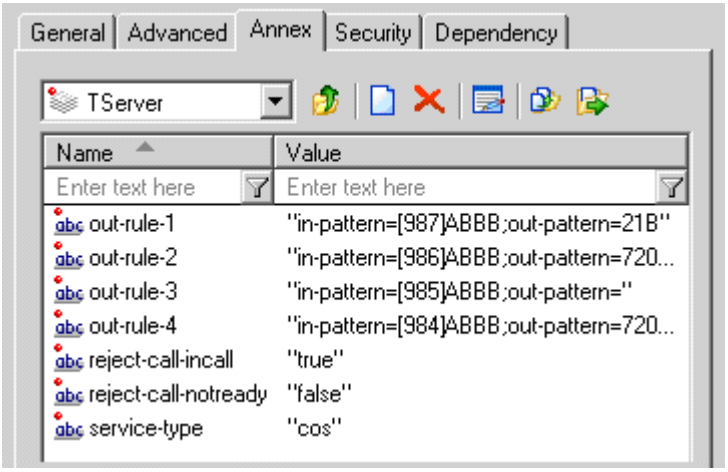

Figure 19: Configuring COS DN: Sample Configuration

5. When finished, click `Apply` to save your configuration.

End of procedure

Assigning COS to a Device

COS is assigned to a device by associating the COS DN with the device DN of type `Extension` or `ACD Position`. In the `TServer` section on the `Annex` tab, add the `cos` configuration option, with the value set to the name of the COS DN.

Figure 20 illustrates a sample configuration of the device DN with COS assigned.

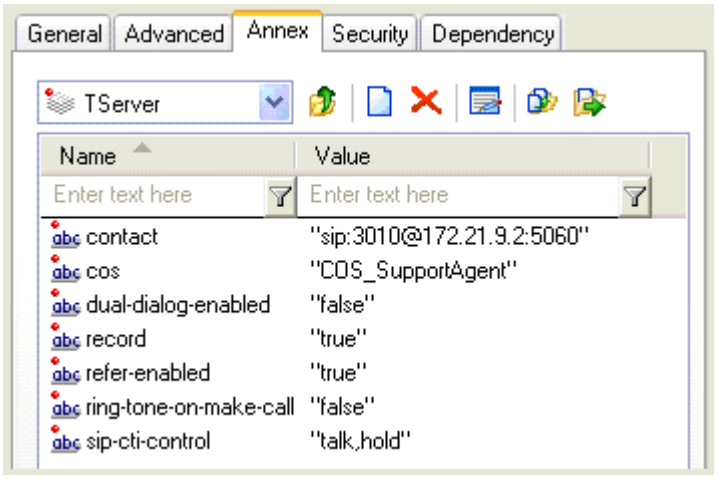

Figure 20: Assigning COS to a Device: Sample Configuration

COS may be assigned to multiple DNs by using the `Manage Options` command in Configuration Manager.

Assigning COS to an Agent

COS is assigned to an agent by associating the COS DN with the `Agent Login` object. In the `TServer` section on the `Annex` tab of the `Agent Login` object, add the `cos` configuration option, with the value set to the name of the COS DN.

Figure 21 illustrates a sample configuration of the `Agent Login` object with COS assigned.

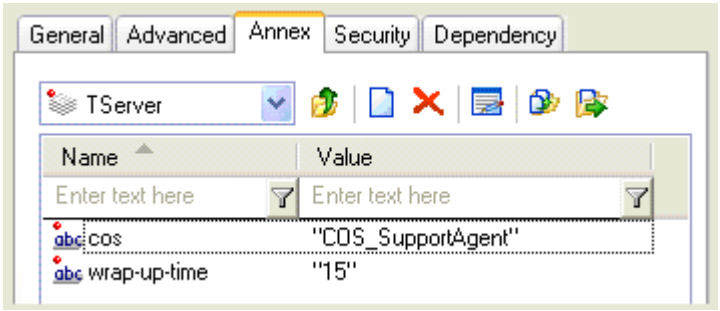

Figure 21: Assigning COS to an Agent Login: Sample Configuration

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, SIP Server checks if COS is configured for the device.
4. If COS is configured for the device, SIP Server applies this device's COS.
5. 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).
6. If the `reject-call-incall` and `reject-call-notready` options are specified, SIP Server uses these options.
7. If none of the preceding apply, SIP Server considers the destination available.

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 22 shows a multi-site scenario that involves an SBC, and in which agents and callers at different locations have the same DN number.

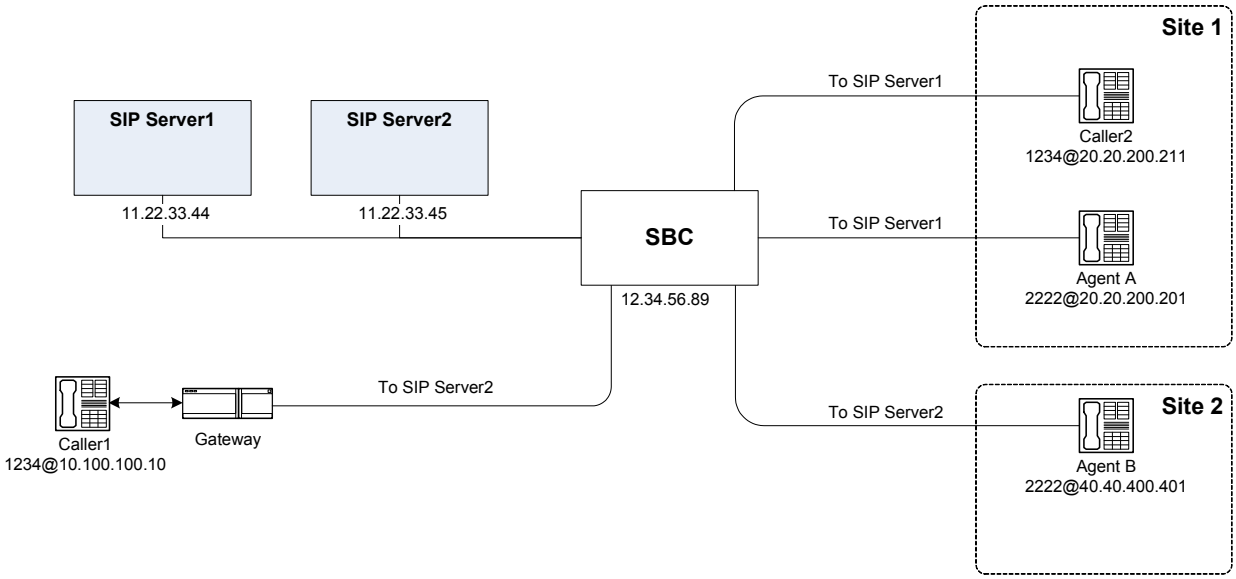

Figure 22: SIP Server with Cookie Persistence Enabled

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

The following task summary provides an overview of the steps that are required to configure the contact header handling methods.

Task Summary: Configuring Contact Header Handling

Objective	Related Procedures and Actions
• Configure for all DNs.	Go to SIP Server `Application` object > `Options` tab > `TServer` section. • To strip contact header—Set `sip-preserve-contact` to the default `false`. • To preserve contact header—Set `sip-preserve-contact` to `true`. For details, see the option description on page 318.
• Configure for individual DNs.	Go to SIP Server switch > DNs folder > individual DN > `Annex` tab > `TServer` section. • To strip contact header—Set `sip-preserve-contact` to the default `false`. • To preserve contact header—Set `sip-preserve-contact` to `true`. The DN-level setting takes precedence over the application-level setting.

Control of SIP Response Code from within a 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

The following table shows the configuration needed to enable this feature.

Task Summary: Configuring SIP Error Response Codes

Objective	Key Procedures and Actions
• Send SIP Response code to routing strategy.	Go to SIP Server `Application` object > `Options` tab > `TServer` section and 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:
   ```
   ExtensionAttached['{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 this case, in the `TServer` section on the `Options` tab 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

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.

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 highest to lowest priority:

- Agent Login—Applies to calls made by a caller logged in under this Agent Login ID.
- 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.
- 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: 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 126.

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).

If the initiating DN is assigned a dial-plan, SIP Server can apply dial-plan functionality for any of the following 3pcc requests:

- TMakeCall
- TInitiateTransfer
- TInitiateConference
- TSingleStepTransfer
- TSingleStepConference
- TRedirectCall
- TCompleteTransfer
- TCompleteConference

SIP Server will also apply dial-plan logic to 1pcc INVTE, REFER, and 302 (moved temporarily) operations.

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 following syntax.

Table 16: 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.

Table 16: Asterisk Dial Plan Syntax for Pattern Matching (Continued)

Special Character	Pattern Matching
N	Matches any single digit from 2-9.
[ ]	Matches any of the digits found in 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 semi-colon. 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 17 describes additional parameters you can use to define the behavior of the dial-plan rule.

Table 17: 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 17: 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. You can use the `${DIGITS}` variable to configure `ondnd`.
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.

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 `MakeCall`, `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 highest to lowest priority:

- Agent Login—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 cos is undefined).
- 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

The following task summary provides an overview of the steps required to configure a dial plan.

Task Summary: Configuring a 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 `TServer` section, configure the following options: • `service-type` • `dial-plan-rule-<n>` For details, see Procedure: Configuring a Dial Plan DN, on page 149.
2. Configure a class of service DN.	1. Create a `Voice over IP Service` DN. 2. In the `TServer` section, configure the following options: • `service-type` • `privilege-level` • `fwd-privilege-level` For details, see Procedure: Configuring Class of Service for a Dial Plan, on page 151. **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 126.

Task Summary: Configuring a Dial Plan (Continued)

Objective	Related Procedure and Key Actions
3. Assign the service DNs.	**Global Configuration** 1. Open the SIP Server `Application` object. 2. In the `TServer` section, configure the following options: ◆ `cos` ◆ `dial-plan` **Individual Configuration** 1. Open the DN or `Agent Login` object. 2. In the `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 153 • Procedure: Assigning COS to a device, on page 155 • Procedure: Assigning the dial plan and COS globally, on page 155
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. 1. Open a dial-plan `Voice over IP Service` DN. 2. In the `TServer` section, configure the following option: ◆ `include-dial-plan-<n>` See Procedure: Including additional dial-plans, on page 152.

Procedure: Configuring a Dial Plan DN

Start of procedure

1. Under the SIP Server `Switch` object, create a `Voice over IP Service` DN.
2. On the `Annex` tab of the dial-plan `DN` object, in 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
```

Notes: 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 142.
- `digits`—See "Digit Translation" on page 143
- `param1=value1`, and so on—See "Dial Plan Parameters" on page 144.

Figure 23 illustrates a sample configuration for a dial-plan DN with dial-plan rules specified.

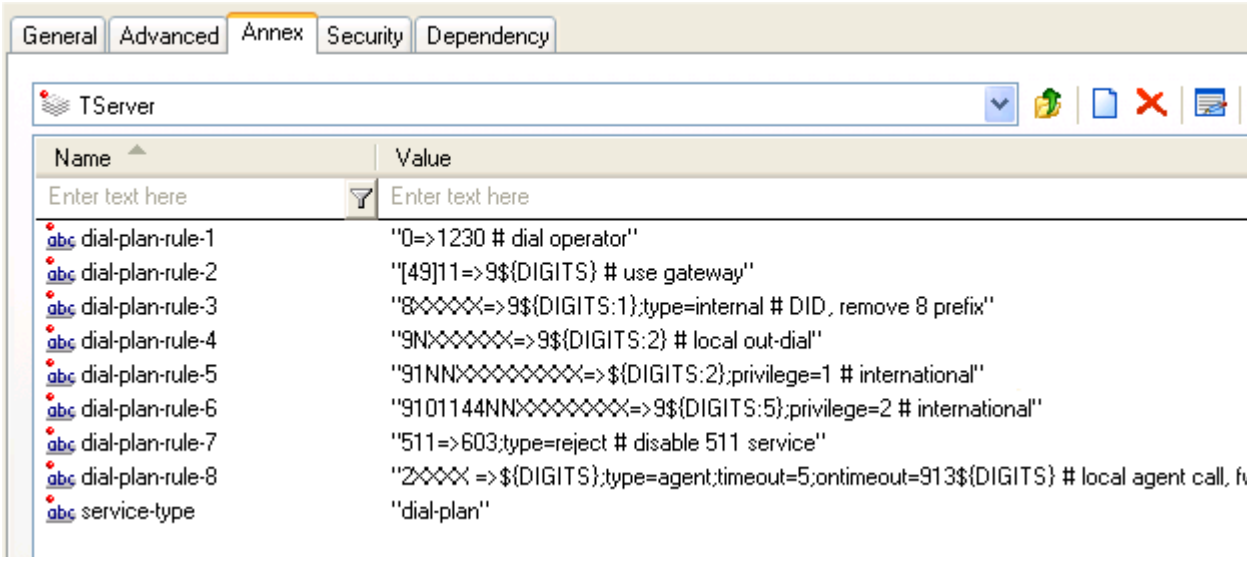

Figure 23: Configuring the Dial-Plan DN: Sample Configuration

4. When finished, click `Apply` to save your 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.
- To assign the dial plan to a particular device, see Procedure: Assigning the dial plan to a device.
- 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.

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 `Annex` tab of 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.

   Figure 24 illustrates a sample configuration for the COS DN with the ring-through rules and privilege-level specified.

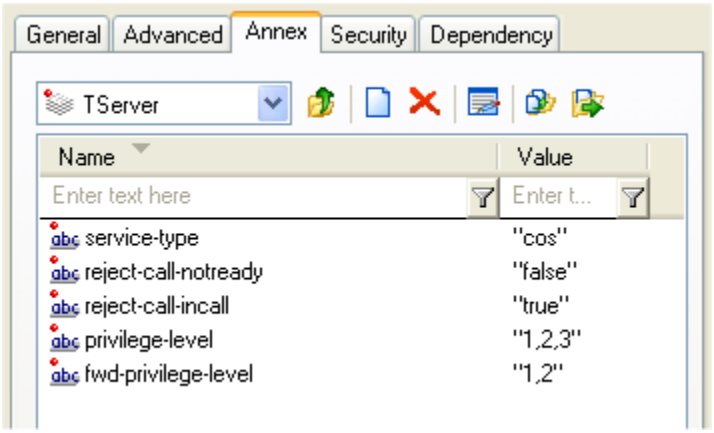

Figure 24: Configuring COS DN for a Dial Plan: Sample Configuration

4. When finished, click `Apply` to save your configuration.

End of procedure

Next Steps

- To assign the Class of Service to a particular device, see Procedure: Assigning COS to a device.
- 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.

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.

> **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. On the `Options` tab, in the `T-Server` section, configure the following options:
   - `include-dial-plan-<n>`—Set this 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 25 shows a sample dial-plan DN that includes other dial plans.

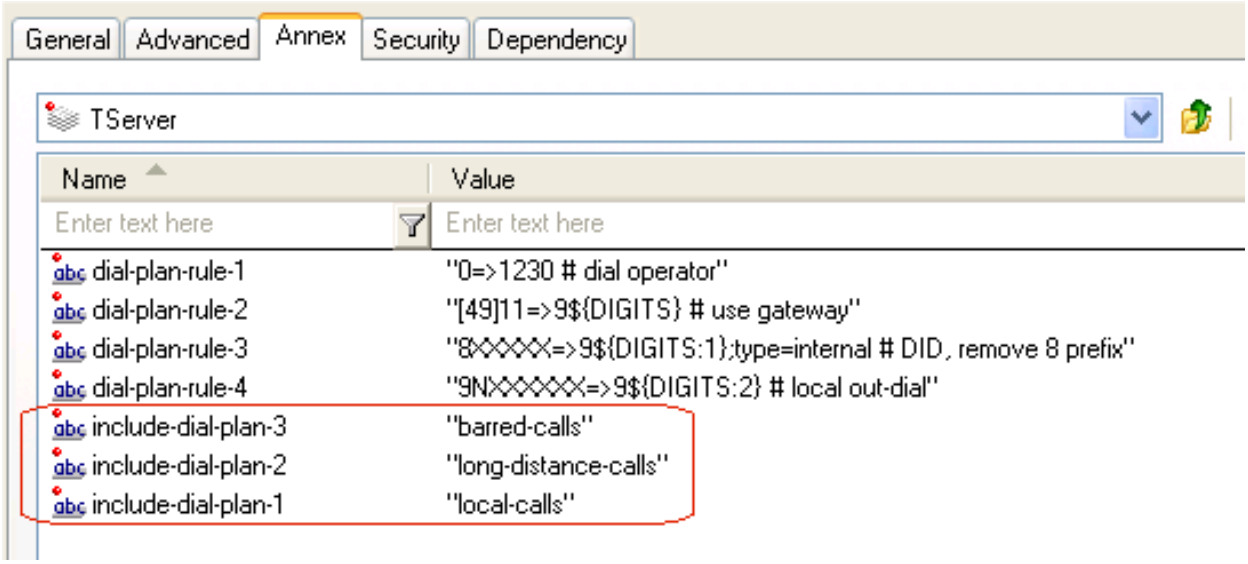

Figure 25: Sample Dial-Plan DN That Includes Other Dial Plans

4. When finished, click `Apply` to save your configuration.

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.

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.

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` object
   - `Extension` DN
   - `ACD Position` DN
   - `Trunk` DN
2. Add the `dial-plan` configuration option, with the value set to the name of the dial-plan DN.

Figure 26 illustrates a sample configuration of the device DN with an assigned dial-plan and COS.

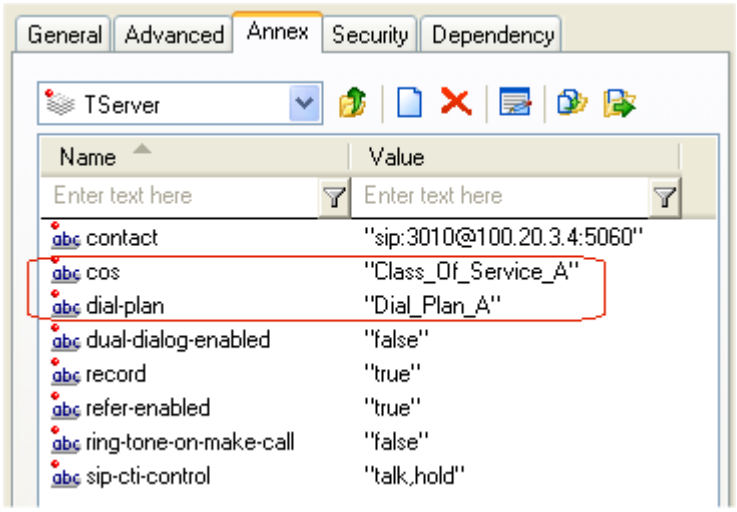

Figure 26: Assigning the Dial-Plan to a Device: Sample Configuration

Tip: Both COS for the dial-plan and the dial-plan itself can be assigned to multiple DNs by using the `Manage Options` command in Configuration Manager.

3. When finished, click `Apply` to save your configuration.

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.

- 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.

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 Procedure: Configuring a COS DN.

Start of procedure

1. Under the SIP Server `Switch`, open the configuration object to which you want to apply the COS DN. Supported objects include:
   - `Agent Login` object
   - `Extension` DN
   - `ACD Position` DN
2. In the `TServer` section on the `Annex` tab, add the configuration option `cos`, with the value set to the name of the COS DN that you configured in Procedure: Configuring a COS DN, on page 135.
3. When finished, click `Apply` to save your configuration.

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.
- A configured COS DN with defined privilege-levels (`Voice over IP Service` DN with `service-type` set to `cos`). See Procedure: Configuring a COS DN.

Start of procedure

1. Go to the `Applications` folder and open the SIP Server `Application` object.
2. On the `Options` tab, in the `T-Server` section, configure the following options:
   - dial-plan—Set this to the name of the dial-plan DN.
   - cos—Set this to the name of the COS DN.
3. When finished, click `Apply` to save your configuration.

End of procedure

DTMF Tones Generation

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 Genesys Agent Desktop.
- A routing strategy can instruct SIP Server to send a request to the media server using the `TApplyTreatment` request with the `PlayApplication` treatment.

DTMF Parameters for Play Application Treatments

The following key-value pairs (see Table 18) are used in the attribute `parameters` for the `PlayApplication` treatment:

Table 18: Key-Value Pairs for TreatmentPlayApplication

Key	Type	Value
GSIP_APP_ID	Integer	Number of the specific application to be executed. Set this parameter to the value of `502` to trigger the play application.
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.

Table 18: Key-Value Pairs for TreatmentPlayApplication (Continued)

Key	Type	Value
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

The following task summary provides an overview of the main steps required to configure DTMF tones generation support.

Task Summary: Configuring DTMF Tones Generation Support

Objective	Related Procedures and Actions
1. Configure a SIP Server `Application` object.	*Required for both methods: TSendDTMF and TreatmentPlayApplication.* In the `TServer` section on the `Options` tab of the SIP Server `Application` object in Configuration Manager, set the configuration option `sip-dtmf-send-rtp` to `true`. **Note**: This option setting is required for both methods: `TSendDTMF` and `TreatmentPlayApplication`.
2. Configure an application service.	Configure a DN of type `Voice over IP Service` with the following configuration options: • `contact`: Set to the device's IP address used for sending DTMF tones. • `service-type`: Set to `application`. See "Configuring an application service" on page 74 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 `RouteRequest`. These parameters are configured in the routing strategy as key-value pairs in the `AttributeExtensions` of the `TRouteCall` request.

The following table describes the key-value pairs used to add dummy SDP-related `Extensions` to the `TRouteCall`, in order to form the dummy SDP in the corresponding `INVITE`.

Table 19: Key-value Pairs for the Dummy SDP

Key	Value
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.

Table 19: Key-value Pairs for the Dummy SDP (Continued)

Key	Value
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**: This parameter does not apply to SDP formation only, and can be used for other reasons. For more information, see the description for the option `after-routing-timeout` on page 278.

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

The following task summary provides an overview of the steps required to configure a dummy SDP.

Task Summary: Configuring Dummy SDP

Objective	Related Procedure and Key Actions
1. Configure the Application.	In the `TServer` section of the SIP Server `Application` object, configure the following options: • `sip-treatments-continuous`—Set this to `true`. • `sip-enable-100rel`—Set this to `true`. • `divert-on-ringing`—Set this to `false`.
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 8.0 Reference Manual*.

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, in order 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.

Sample Call Flow

The following 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
   - SIP 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.

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`. The following task summary provides an overview of the main steps that are required to configure this feature.

Task Summary: 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 for the incoming gateway as follows: • `sip-early-dialog-mode`—On the `Annex` tab, in the `TServer` section, set this option to `1`. OR `sip-server-inter-trunk`—Set this option to `true`. For details, see the option description on page 359.
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`. For details, see the option description on page 333.
3. Configure the SIP Server Application.	Genesys recommends that you also configure the SIP Server Application as follows: • On the `Options` tab, in the `TServer` section, set `ringing-on-route-point` to `false`. For details, see the option description on page 311.

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 as a *business call* any call distributed to an agent either from a Queue or from a Routing Point. Use the following configuration options to define what additional calls to or from an agent are classified as business calls:

- `inbound-bsns-calls` (page 298)
- `outbound-bsns-calls` (page 305)
- `inherit-bsns-type` (page 299)
- `internal-bsns-calls` (page 299)
- `unknown-bsns-calls` (page 325)
- `agent-only-private-calls` (page 280)

Work-Related Calls

SIP Server categorizes as a *work-related call* any call that an agent makes while in the busy (with a business call), After Call Work (ACW), or aux-work state. 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 as a *private call* any call that does not fall into the business or work-related categories. 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:

- `emulate-login` (page 369)
- `emulated-login-state` (page 289)
- `agent-strict-id` (page 280)

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 7 Events and Models Reference Manual and the Voice Platform SDK 7.6 .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 163).

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 Configuration Manager. 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 (page 324)
- acw-in-idle-force-ready (page 277)

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` (see page 282).

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` (page 300).

Endpoint Service Monitoring

When SIP Server starts up, it considers that all DNs that are configured in Configuration Manager 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.
- When the device entry in Configuration Manager 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.

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` (MCU, treatment, softswitches, and so on)
- `Trunk`

Enabling Active Out-of-Service Detection

To enable out-of-service detection, configure the following options in the `TServer` section of the `Annex` tab of the corresponding DN:

- `oos-check` (see page 346)
- `oos-force` (see page 347)

Note: Genesys recommends that you do 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 `OPTION` 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 now 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 new DN-level option `oos-options-max-forwards` (page 347) 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
```

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) in order 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 Stream Manager 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.

Feature Configuration

The following table provides an overview of the basic mandatory steps required to integrate SIP Server into an IMS network.

Task Summary: Integrating SIP Server with IMS

Objective	Related Procedures and Actions
1. Configure SIP Server.	Go to SIP Server `Application` object > `Options` tab > `TServer` section • `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.
2. Configure IMS endpoints.	Go to SIP Server `switch` > `DNs` folder . 1. Create a DN of type `Extension`. 2. In the `Annex` tab, `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 `Annex` tab, `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 any valid SIP URI for the S-CSCF. ◦ `enable-ims`—Set to `true`.

Note: This task table 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.

  The IM SDP message must be in a form that is generated and accepted by Microsoft Live Communication Server and Office Communicator. 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.

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 the chat transcript, and to send an instant message to other parties of 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.

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 the new participant who was added to the call.

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

SIP Server supports the following 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 `'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

- 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.
- 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.

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 transfer 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 Stream Manager 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 Stream Manager 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 101. 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 8.0 Reference Manual*.

The agent can also handle one voice call and several IM sessions at once.

Feature Configuration

Processing UserData

The following options are used to configure how `UserData` is processed:

- `user-data-im-enabled` (see page 325)
- `sip-signaling-chat` (see page 364)
- `sip-chat-format` (see page 358)

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.

Notes: 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 220.

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.

Mapping SIP Headers and SDP Messages

SIP Server can now 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.

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`

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 headers and parameters of the `INVITE` message and then populated into the `Extensions` or `UserData` attributes in the `EventRouteRequest` message. You can configure which headers and 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 values of the headers and the header parameters that are extracted from the SIP message. The names of the options within this section contain a prefix, a dash (-), and a suffix, in the following format:

`<prefix><-><suffix>`

The prefix is the name of the TEvent attribute, and begins with either `extensions` or `userdata`. The prefix `extensions` instructs SIP Server to put the SIP header or parameter into the `Extensions` attribute. The prefix `userdata` instructs SIP Server to put the SIP header or parameter into the `UserData` attribute.

The suffix value is a number, and must be unique for all option names containing the same prefix.

The option value determines which header in the header parameter of the SIP message is processed.

The value of the header or header parameter is added as a key-value pair into the attribute, using the form `<header_name>=<header_value>`. Use the colon character (`:`) to address the parameter name of a header. 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.

Mapping occurs only if both the following conditions are true:

- The header or header parameter is contained within the incoming `INVITE` message.
- The 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 within the SIP `INVITE` message.

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 in SIP headers
- Mapping data in the body of the 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 in order 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.

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. For integration with GVP, no special configuration on GVP or on the SIP Server switch is required.

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 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 177).

To map data from the `REFER` message into the `Extensions` and/or `UserData` attributes, configure `userdata-<n>` options and `extensions-<n>` option in the `REFER` section.

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.

Known Limitations

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.

From T-Library Messages to SIP Messages

SIP Server can map headers or header parameters passed in the RequestRouteCall 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 RequestRouteCall 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.

Using the extensions-<n> Option

To use extensions-<n> to configure T-Library request attributes to the SIP mapping, you configure extensions-<n> options in a SIP Server Application object using Configuration Manager. Those options are specified in the section named after the SIP request used to route the call, which is INVITE.

Configuration Example

This example demonstrates how to map a TRouteCall request extension called InfoToSendInInvite to the outgoing INVITE message.

The following steps show how 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.

Example

This example demonstrates how mapping works in SIP Server when it receives the following `TRouteCall` request:

```
message RequestRouteCall
              AttributeThisDN'5000'
              AttributeConnID006e01886c3d7001
              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 that 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
```

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.

Feature Configuration

The following table describes the configuration required to enable dynamic DN replacement.

Task Summary: Configuring DN Replacement in SIP Headers

Objective	Related Procedures and Actions
Configure the Application.	In the T-Server section of the SIP Server Application object, configure the following option: • tlib-map-replace-dn—Set this to true.

Media Server Reliability

SIP Server supports reliability for music-on-hold; treatments that are set to continuous playback, conference, or supervisor; and 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

In order 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. It can also detect service failure in the midst of a call, after successful service engagement. In this case, if SIP Server detects an out-of-service device, it is able to continue the service on an alternate available media server device.

  Note: For more information, see "Endpoint Service Monitoring" on page 167

- 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.

- 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.

- 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 will detect the failure and restart 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 will restart the Supervisor features on a second media server, with minimal impact on the customer experience.

For details about these features, see the following references:

- "Call Supervision" on page 101.
- The "Supported Functionality in Stream Manager" chapter of the *Framework 7.6 Stream Manager Deployment Guide*.

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" on page 97.

Feature Configuration

No special configuration is required to enable reliability for the supported media service features. 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 midst of a call.

Configuring Active Out-of-Service Detection

To configure the length of time between OPTION requests, use the options oos-check and oos-force options (configured on the Annex tab of the Voice over IP Service DN). For more information about configuring this feature, see "Active Out-of-Service Detection" on page 168.

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.

Music and Announcements

SIP Server is able to control the playing of announcements by using Stream Manager. Stream Manager plays files using a codec negotiated with the SIP Server switch. The list of possible codecs is configured in SIP Server with the audio-codecs option (page 281). For information about how to configure Stream Manager refer to "Configuring Stream Manager" on page 80.

Announcement Treatments on Routing Points

The two announcement treatments, PlayAnnouncement and PlayAnnouncementAndDigits, include the parameters listed in Table 20.

Table 20: 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 189.
ID	Contains an integer, that refers to the Stream Manager announcement/<integer> file. For example, the value 1 refers to the file announcement/1_alaw.wav, if the G.711 A-law codec is used.

Table 20: Announcement Treatment Parameters (Continued)

Parameter	Description
DIGITS	Ignored.
USER_ID	Supported by Stream Manager using the users/<customer id>_<ann.id> file.
USER_ANN_ID	Supported by Stream Manager using the users/<customer id>_<ann.id> file.
TEXT	Ignored.

Music-on-hold is not available for these treatment types. Stream Manager is the only source for the announcements.

If the treatment is terminated early because of a problem with Stream Manager 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 7.6 Reference Manual* for more information on the use and configuration of strategies.

Note: Leave the Wait For Treatment End check box selected, to allow the treatment to play until completion.

Music Treatments on Routing Points

Table 21 describes the parameters for music treatments:

Table 21: Music Treatment Parameters

Parameter	Description
MUSIC_DN	Specifies the music source that Stream Manager plays. The format is: <directory>/<music file name> Where <directory> is a sub-directory of the Stream Manager root directory, and <music file name> refers to the name of the file—without the codec extension. For example, music/in_queue refers to the file music/in_queue_alaw.wav if the G.711 A-law codec is used. To specify the number of repetitions, the parameter repeat=<N> must be used, where <N> is any positive integer. If no repetition is specified, the music file loops forever. The valid formats are: • <directory>/<music file name>—The specified file loops endlessly. • <directory>/<music file name>;repeat=<N>—The specified file is repeated <N> times. 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. This treatment ends before music is played. To continue playing music after the treatment terminates, consider creating one of the following strategies: • 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 8.0 Reference Manual* for more information on use and configuration of strategies.

Other Treatments on Routing Points

These treatments in Table 22 continuously loop a pre-defined audio file to a call. You must configure Stream Manager to use these treatments. `Music-on-hold` cannot be the source of the audio file. The treatment types are as follows:

Table 22: 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 (page 284).
Fast Busy	Plays a fast busy tone. To define the fast busy tone audio file, configure the SIP Server `fast-busy-tone` option (page 293).
Silence	Plays no sound. To define the `silence` audio file, configure the SIP Server `silence-tone` option (page 312).
Ringback	Plays ringback tone. To define the `ringback` audio file, configure the SIP Server `ring-tone` option (page 310).
CollectDigits	Collects customer-entered digits.
RecordUser-Announcement	Records a user's announcement and saves into a `users` folder.

Refer to the *Universal Routing 7.6 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 Connection

SIP Server can now provide a persistent SIP session for agents who require a dedicated connection to the contact center—typically, for TDM agents who are dialing in to the contact center from the PSTN. The agent can make calls, wait for calls, as well as send or accept transfers while maintaining the current SIP connection to the contact center. If a call to the agent DN is released by either the caller or by an agent 3pcc request (`TReleaseCall`, `TSingleStepTransferCall`, or `TCompleteTransfer`), SIP Server does not end the

SIP session but, instead, it parks the agent DN on the predefined `gcti::park` device. The session can be released only after the agent hangs up the phone (1pcc release call).

How SIP Server Establishes the Nailed-Up Connection

SIP Server can set up the nailed-up connection between the remote TDM agent and the contact center using the following methods:

- SIP Server calls the agent to start a session—SIP Server sends the call to a remote TDM agent 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 Stream Manager, keeping the connection for the call leg to the nailed-up connection.
- The agent calls the contact center to start a session—The remote TDM agent (configured for the nailed-up feature) initiates a call (1pcc) to the contact center.

SIP Server Calls the Agent to Start a Session

The basic call flow when SIP Server first calls an agent who is 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. Stream Manager plays a silent treatment while the nailed-up connection is maintained.

The Agent Calls the Contact Center to Start a Session

The basic call flow when a remote TDM agent (who is configured for the nailed-up feature) first calls the contact center 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. Stream Manager plays a silent treatment while the nailed-up connection is maintained.

Processing Contact Center Calls for the Nailed-Up Connection

Agents on a nailed-up DN can make new calls from the parked state, using the `TMakeCall` request. In this case, the existing SIP session is used as the calling party.

When a new call is initiated toward the parked device, the existing SIP session is used to connect the call to the device. SIP Server considers the device to be in auto-answer mode when the line is parked.

Processing a Customer Call to an Agent with a Nailed-Up Connection

The basic call flow when the contact center directs a customer call to a nailed-up connection 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 that will handle the call.
3. The selected agent DN already has a nailed-up connection (parked on Stream Manager). SIP Server disconnects the nailed-up connection from Stream Manager, and then connects the agent to the customer leg.
4. At the end of the call, when the customer releases the call, does not disconnect the call leg to the nailed-up connection, but instead parks the agent once again on Stream Manager. Stream Manager plays a silent treatment while the nailed-up connection is maintained.

Reporting on Nailed-Up Connection Calls

Calls that involve the nailed-up connection use T-Library events that are identical to those that used for a regular agent. Each customer interaction established with the nailed-up agent is reported as a distinct Genesys call, allowing monitoring and reporting on each individual call.

Feature Configuration

The following task summary provides an overview of the basic steps that are required to configure the nailed-up feature for a remote TDM agent DN.

Task Summary: Configuring a Nailed-Up Connection

Objective	Related Procedures and Actions
1. Configure the gateway.	• In the `Trunk` DN that represents the gateway that is used to contact the remote agent, set `refer-enabled` to `false`. For general information about creating a gateway DN, go to: • Procedure: Configuring a gateway, on page 68
2. Configure the agent DN.	In the `ACD Position` or `Extension` DN for the agent, configure the following options: • `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 info about creating agent DNs, go to: • Procedure: Configuring endpoints, on page 65.

Feature Limitations

Consultation calls for nailed-up DNs are supported in single dialog mode only.

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, then 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, then 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 the INVITE is sent to an untrusted entity, then the INVITE will not include the P-Asserted-Identity header (no identifying user information is passed).

Trusted

For example, an INVITE between trusted 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
...
```

Non-trusted

An INVITE that either arrives from an untrusted entity, or is being sent to an untrusted entity 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
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

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 header.
- `Extension` DN is configured with `privacy` option set to `true`.
- The `privacy` option is set to `id` on the originating DN for the call, or on the incoming `Trunk` or `Voice over IP Service` DN for further call processing.

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 highest to lowest priority:

1. Value of the `p-asserted-identity` option configured on the `Trunk`, `Voice over IP Service`, or `Extension` DN.
2. Content in the `P-Asserted-Identity` header included in the incoming `INVITE` from a trusted origination.
3. Content in the `P-Preferred-Identity` header included in the incoming `INVITE` from a trusted origination.
4. 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 27 shows a sample call flow for an inbound call between a trusted external domain and a trusted destination DN on SIP Server.

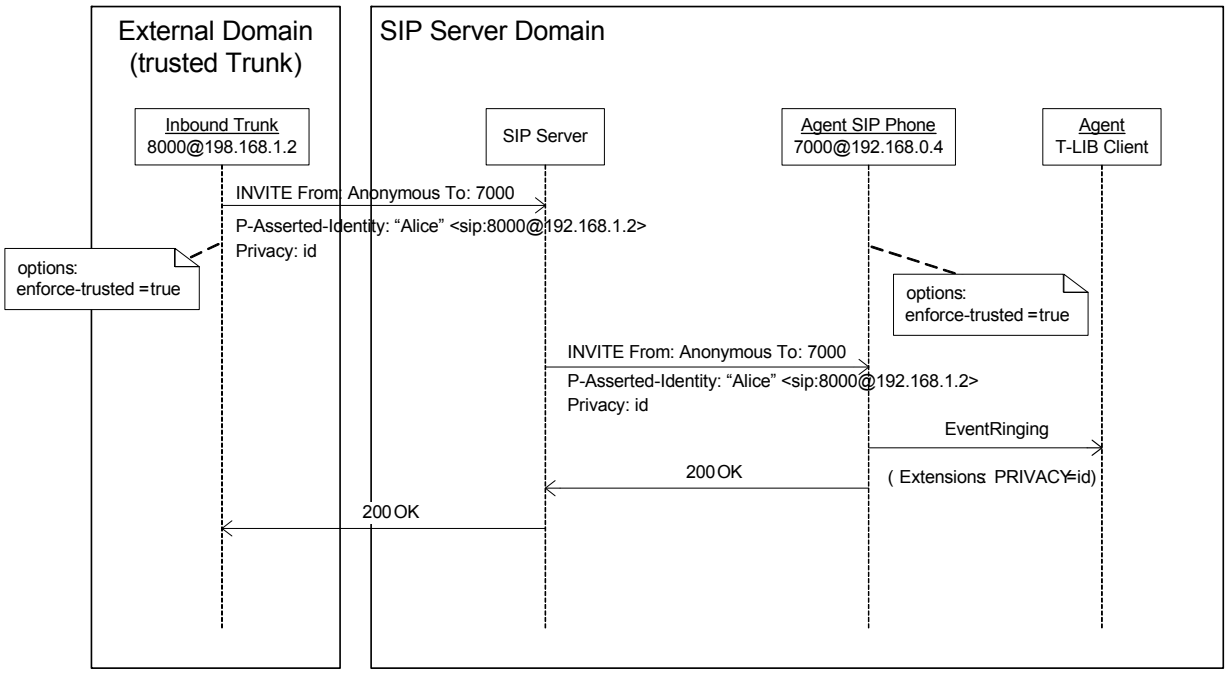

Figure 27: Trusted External Call To Trusted Destination DN (with privacy restriction)

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 28 shows a sample call flow for an outbound call from a trusted internal DN to an untrusted external destination.

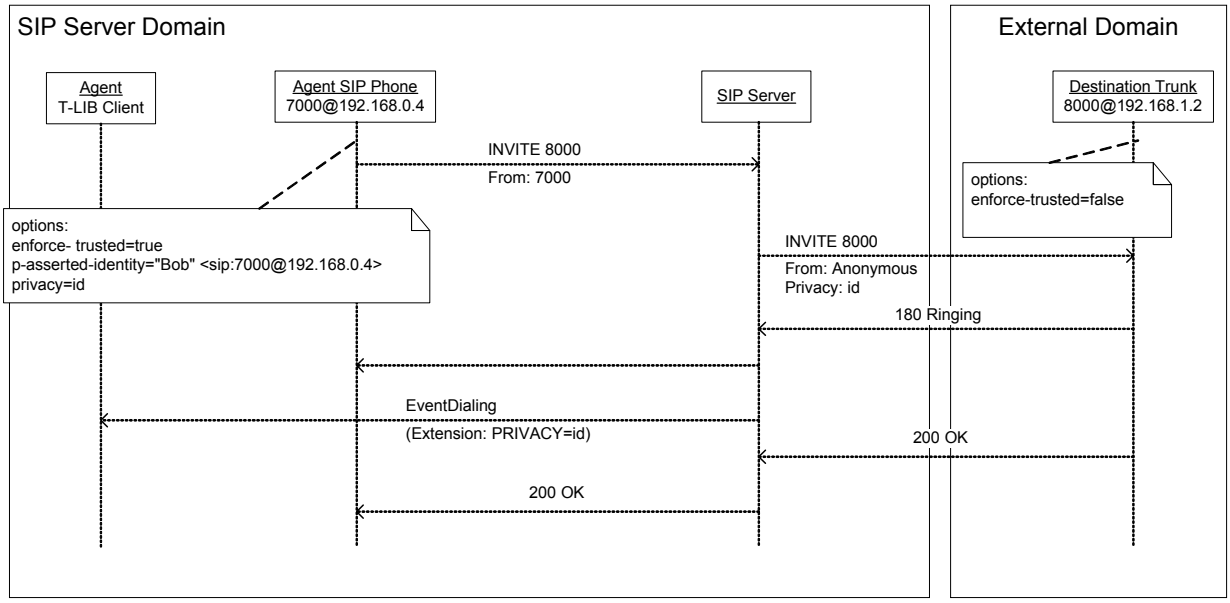

Figure 28: Outbound Call to Non-Trusted Destination (With Privacy Restriction)

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 (`trusted` option set to `true`), 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 29 shows a sample call flow for an internal call from a trusted `Extension` DN to an untrusted `Extension` DN within the SIP Server domain.

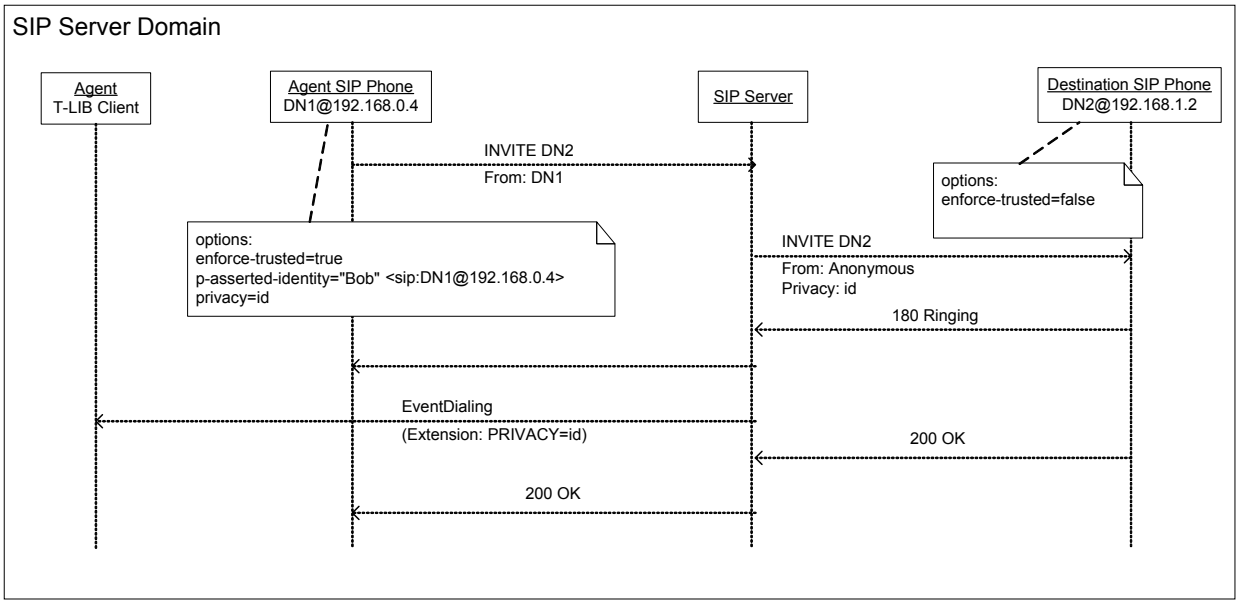

Figure 29: Internal Call From Trusted Entity to Non-Trusted Entity

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

The following table shows the basic configuration steps required to control how privacy is provided for SIP messaging within the network.

Task Summary: Configuring Network Asserted Identity

Objective	Key Procedures and Actions
Request privacy for outbound calls.	In `Extension` DN that places the call, 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`, configure the following option: • `enforce-trusted`—Set to `true`. **If the INVITE does not include privacy...** In the `Trunk` DN for the external device, 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, 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, configure the following option: • `enforce-trusted`—Set to `true`.

Task Summary: Configuring Network Asserted Identity

Objective	Key Procedures and Actions
Request presentation for outbound calls.	In `Extension` DN that places the call, configure the following option: • `p-asserted-identity`—Set to correct URI.
Enable COLR for inbound calls.	To restrict presentation of the called line identity to the caller (COLR), do the following: 1. Configure the inbound `Trunk` DN with `enforce-trusted` set to `true`. Configure the destination DN with the following options: • `privacy`—Set to `id`. • `p-asserted-identity`—Set this to the identity you want to include in the `P-Asserted-Identity` header.

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 as a *business call* any call distributed to an agent either from a Queue or from a Routing Point. Use the following configuration options to define what additional calls to or from an agent are classified as business calls:

- `inbound-bsns-calls` (page 298)
- `outbound-bsns-calls` (page 305)
- `inherit-bsns-type` (page 299)
- `internal-bsns-calls` (page 299)
- `unknown-bsns-calls` (page 325)

Private Calls

SIP Server categorizes as a *private call* any call that does not fall into the business or work-related categories. 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` (page 305).

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` (page 279)
- `agent-no-answer-overflow` (page 279)
- `agent-no-answer-timeout` (page 280)
- `nas-private` (page 305)
- `set-notready-on-busy` (page 312)

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` (page 292)
- `extn-no-answer-timeout` (page 293)

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` (page 307)
- `posn-no-answer-timeout` (page 308)

Device-Specific Overrides

SIP Server provides three configuration options with which you can configure device-specific overrides for individual devices. You set the values for these options in the `TServer` section on the `Annex` tab of the individual device. The options are:

- `no-answer-action` (page 344)
- `no-answer-overflow` (page 344)
- `no-answer-timeout` (page 345)

Extension Attributes for Overrides for Individual Calls

For all of the No-Answer Supervision options, you can specify the corresponding `Extension` 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_TIMEOUT`
- `NO_ANSWER_OVERFLOW`
- `NO_ANSWER_ACTION`

See “Using the Extensions Attribute” on page 264 for more information.

Feature Limitations

No-Answer Supervision functionality will not be activated if the `divert-on-ringing` configuration option is set to `false`.

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 Server 8.0 Deployment Guide*.

SIP Server Features for Outbound IP Solution

Table 23 describes SIP Server features required for the Outbound IP Solution.

Table 23: 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. For more information, see "Working with Multiple Devices" on page 247.	• `partition-id` (page 307) • `cpd-capability` (page 336)

Table 23: SIP Server Feature Support for Outbound IP Solution (Continued)

Feature	Related Options
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 now 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` (page 365) • `msml-support` (page 302)
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` (page 332)
Updated timeout for CPD INFO messages SIP Server supports a modified `cpd-info-timeout` option.	• `cpd-info-timeout` (page 285)
Additional Extensions for TMakePredictiveCall requests SIP Server now supports two new key-value pairs, `AnsMachine` and `FaxDest`, that you can include in the `Extensions` attribute of `TMakePredictiveCall` requests. Use these extensions to override the corresponding SIP Server Application-level options `am-detected` and `fax-detected`. Both extensions can be configured with only one possible value: `drop`. If either extension appears in the `TMakePredictiveCall`, SIP Server drops the call if the requested CPD result (AM or FAX) is detected.	• `am-detected` (page 281) • `fax-detected` (page 293)

Table 23: SIP Server Feature Support for Outbound IP Solution (Continued)

Feature	Related Options

New media server alarms
SIP Server supports two new alarms used to report the state of the SIP Server subscription to the GVP Resource Manager, 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 GVP 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` (page 346)

Predictive calls on a Routing Point DN SIP Server supports a modified `predictive-call-router-timeout` option.	• `predictive-call-router-timeout` (page 308)

Table 23: SIP Server Feature Support for Outbound IP Solution (Continued)

Feature	Related Options
Mapping SIP error codes to Genesys call states SIP Server now 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>` (page 329)

Feature Configuration

Basic configuration for the Outbound IP solution includes the following options, which you configure in the `TServer` section of the GVP Resource Manager `Trunk Group` DN:

Task Summary: Configuring the GVP DN for Outbound IP

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.
`ring-tone-on-make-call`	Set this option to `false` (no ring tone is required for scenarios that may include CPD).

Task Summary: Configuring the GVP DN for Outbound IP (Continued)

Option	Setting
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>;gvp-tenant-id=<Tenant_Name> SIP Server sends an INVITE to Resource Manager with the Request-uri modified to tell GVP to act as media server for the call.
subscription-id	Set this to the name of this Trunk Group DN. **Note:** Starting in 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 should apply.
cpd-capability	Set this option to mediaserver. This enables CPD analysis to be performed by the MCP.

Feature Limitation

- 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.

Overload Control

SIP Server includes an overflow control mechanism in order 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.

If SIP Server discovers that 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 30 shows the different stages of overload control, and how the capacity rates determine how the stages are implemented.

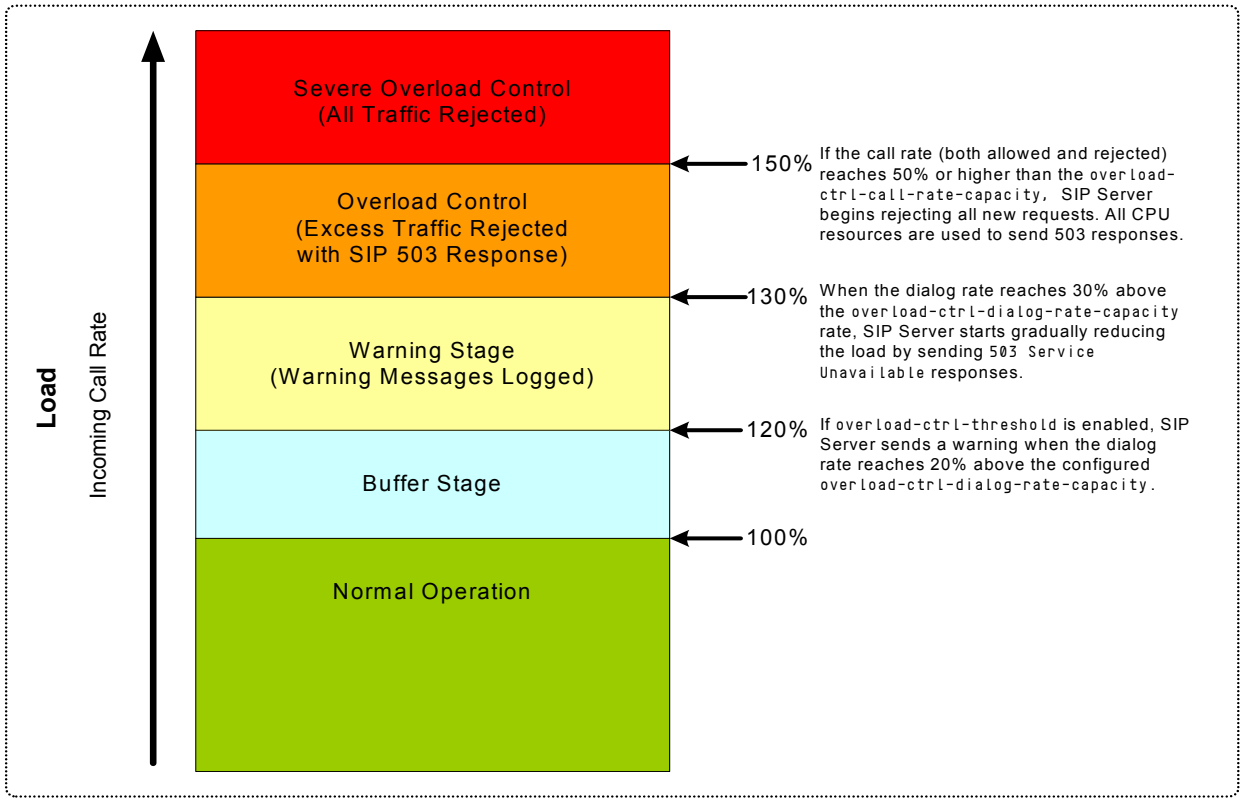

Figure 30: Stages of Overload Control Reaction

Recommended Settings

Figure 31 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.

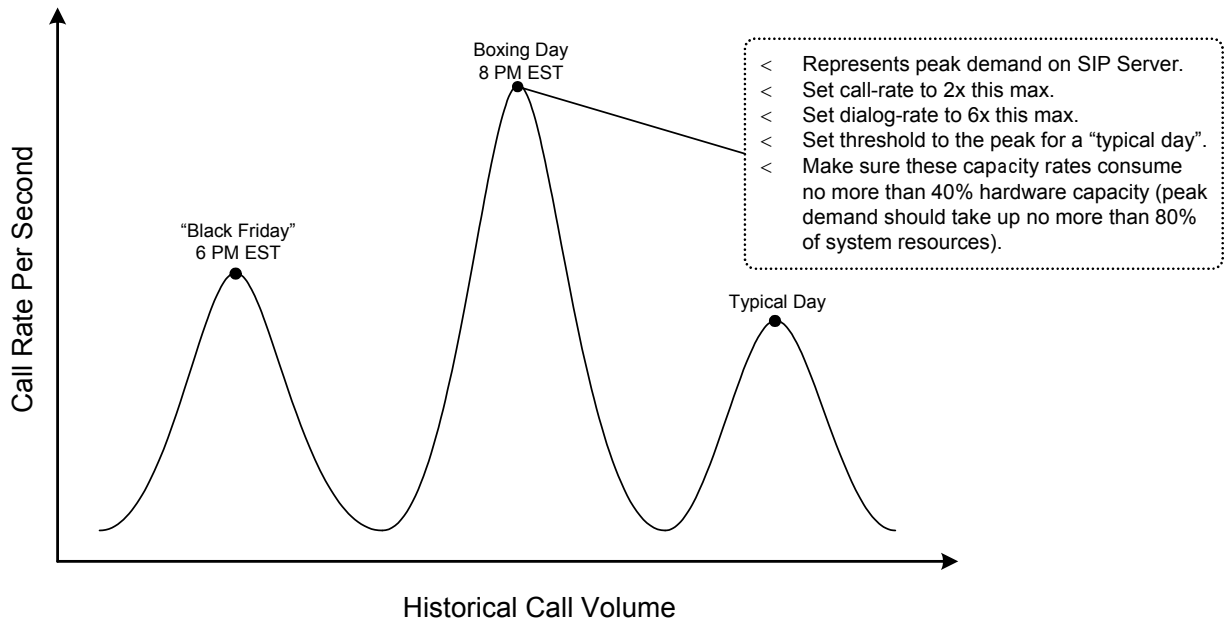

Figure 31: Sample Historical Call Volume and Recommended Overload Control Settings

Feature Configuration

The following task summary provides an overview of the steps required to configure overload control for any of the different overload factors.

Task Summary: Configuring Overload Control

Objective	Related Procedures and Actions
• Enable the overload control mechanism.	In the SIP Server `Application` object, in the `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`.

Task Summary: Configuring Overload Control (Continued)

Objective	Related Procedures and Actions
• Set the capacities for individual load factors.	In the `TServer` section, set the thresholds for the individual load factors as follows: • `overload-ctrl-dialog-rate-capacity`—Set this factor to the SIPDialog 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.
• Disable the overload control mechanism.	In the SIP Server `Application` object, in the `TServer` section, configure the following: • `overload-ctrl-threshold`—Set this threshold to `0`. This will completely disable the overload control feature.

Personal Greeting

Personal greeting functionality enables Stream Manager or 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. Starting in 8.0.3, you can also control how SIP Server handles the personal greeting feature in different scenarios.

Feature Configuration

The following table describes the basic steps required to enable and control this feature.

Task Summary: Enabling Personal Greeting

Objective	Key Procedures and Actions
Configure the Agent Login (mandatory).	Specify the file to be played in the Agent Login object. • In the `TServer` section of the Agent Login, configure the following: ◆ `agent-greeting` ◆ `customer-greeting` For a detailed procedure, see Procedure: Configuring a personal greeting, on page 213.
Configure the SIP Server Application (optional).	Control how SIP Server provides greetings for different scenarios. • In the `TServer` section of the SIP Server Application, configure the following: ◆ `greeting-after-merge`—Set to `true` to enable greetings after transfers or conferences. ◆ `greeting-call-type-filter`—Set to `internal` and/or `consult` 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, `true` to send events after. ◆ `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.

Procedure:
Configuring a personal greeting

Start of procedure

1. Under a configured `Switch` object, select the `Agent Logins` folder. From the `File` menu, select `New` > `Agent Login` to create a new `Agent Login` object.
2. In the `New Agent Login Properties` dialog box, on the `General` tab, specify the Agent Login name in the `Code` text box.
3. Click the `Annex` tab.
4. Click or create a section named `TServer`. In the `TServer` section, create options as specified in Table 24.

Table 24: Configuring a Personal Greeting

Option Name	Option Value	Description
`agent-greeting`	A file name	Specifies the name of the media file that will be used as a greeting for the agent. See the option description on page 330.
`customer-greeting`	A file name	Specifies 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. See the option description on page 336.

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.

5. When you are finished, click `Apply`.

End of procedure

Additional Information

You can also attach these values from a routing strategy. The `agent-greeting` and `customer-greeting` keys enable personal greetings. These keys are processed by SIP Server using the `AttributeExtension` parameters of the `TRouteCall` request.

Notes: The keys that are contained in attribute `Extension` have a higher priority than the options specified in the `Agent Login` object.

The customer greeting plays continuously until the agent greeting finishes playing.

Playing Multiple Media Files

Stream Manager and Genesys Media Server can play different media files—for various contact center music on hold and video on hold requirements—from a single instance of SIP Server. For example, Stream Manager or Genesys Media Server can provide each queue on a single SIP Server with its own particular media file.

Supported Scenarios

This feature supports the following scenarios:

- Stream Manager or Genesys Media Server plays a file that is associated with an agent DN if the agent places the call on hold.
- Stream Manager or Genesys Media Server plays a file that is associated with an `ACD Queue` DN while the call is in the queue.

Where to Configure the Media Files

You can enable media on hold—as well as define the file to be played—using any of several, prioritized methods:

- From an agent or `ACD Queue` DN—When the agent places the call on hold, the defined file will be played for 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`
- From the SIP Server Application—The default file that is configured on the application is played if no other configured filename is found.

Media File Order of Priority

Listed in order of priority from highest to lowest, the following settings determine which file is played:

1. Extensions in the client request—Typically, the agent, logged in to Genesys Agent Desktop, manually selects the music or video file that the agent wants to play to the caller when the caller is placed on hold.
2. DN-level options—When the agent places the call on hold, the default file configured on the agent DN or `ACD Queue` DN is played to the caller.
3. Application-level options—When the agent places the call on hold, the default file that is specified on the SIP Server application is played to the caller.

Configuring the Media Files

Table 25 describes the different configuration methods, in order of priority.

Table 25: Configuring Multiple Media Files for a Single SIP Server

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`—Specifies the file name for music on hold for the call.
Configure an Agent DN.	On the `Annex` tab of the agent 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`—Specifies the music file that is to be played when this DN places the call on hold.
Configure an ACD Queue.	On the `Annex` tab of the `ACDQueue` 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 calls that are queued on this DN. • `default-music`—Specifies 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 `ACDQueue` DN. This option has now been deprecated at the DN-level.

Table 25: Configuring Multiple Media Files for a Single SIP Server (Continued)

Objective	Related Procedures or Actions
Configure the SIP Server Application.	On the `Options` tab of the SIP Server Application, in 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.

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 217
- "SIP Server Subscription to Endpoints Behind a Switch" on page 217
- "Endpoint Sends PUBLISH Requests to SIP Server" on page 218
- "Mapping Polycom Agent-Status Updates to T-Library Events" on page 219
- "Presence Integration with Microsoft Office Communications Server 2007" on page 220

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.

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 62.

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 230.

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 Configuration Manager. 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 Annex tab of the Trunk DN:

- contact (page 334)
- subscribe-presence-domain (page 366)
- subscribe-presence-from (page 366)
- subscribe-presence-expire (page 366)

3. Create a DN of type Extension in Configuration Manager.
4. Configure these options in the TServer section on the Annex tab of the Extension DN:
   - contact (page 334)
   - request-uri (page 356)
   - subscribe-presence (page 367)
   - enable-agentlogin-presence (page 339)
5. Create an Agent Login object in Configuration Manager 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 Configuration Manager by specifying the subscribe-presence option (page 367). 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.

Mapping Polycom Agent-Status Updates to T-Library Events

For Polycom IP phones that support agent-status updates that are 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 Polycom IP phone. In this case, when the agent presses `Login` and specifies their agent ID and password on the IP phone, the agent endpoint sends a `SUBSCRIBE` request to SIP Server. SIP Server generates an `EventAgentLogin` message after the authentication exchange with the IP phone. 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.

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`.

The following table describes the basic steps required 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 in order to convey presence. In these cases, the `ocs-dn` configuration is required.

Task Summary: 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).

Task Summary: Configuring Presence for Microsoft OCS

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 `register`.
	This enables SIP Server to view and map MOC user agent statuses.
	For a screenshot of a configured OCS Trunk DN, see Figure 32 on page 224.

Task Summary: Configuring Presence for Microsoft OCS

Objective	Key Actions and Procedures
4. Configure the Routing Point.	Create the `Routing Point` DN to be exposed in the Microsoft OCS environment. • The name should match the account/name that you configured in Step 1. For example, `ocs-rp`. • Set `subscribe-presence` to the name of the OCS Trunk used for subscription. For a screenshot of a configured OCS Routing Point DN, see Figure 33 on page 225.
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 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 to `session`. • `sip-chat-format`—Set to `text`. • `ocs-dn`—Set this to the expert account/user name that you created in Step 1. For a screenshot of a configure Expert User DN, see Figure 34 on page 225
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 Limitation” on page 225.

Sample OCS Trunk DN

Figure 32 shows a sample screenshot of a configured OCS `Trunk` DN.

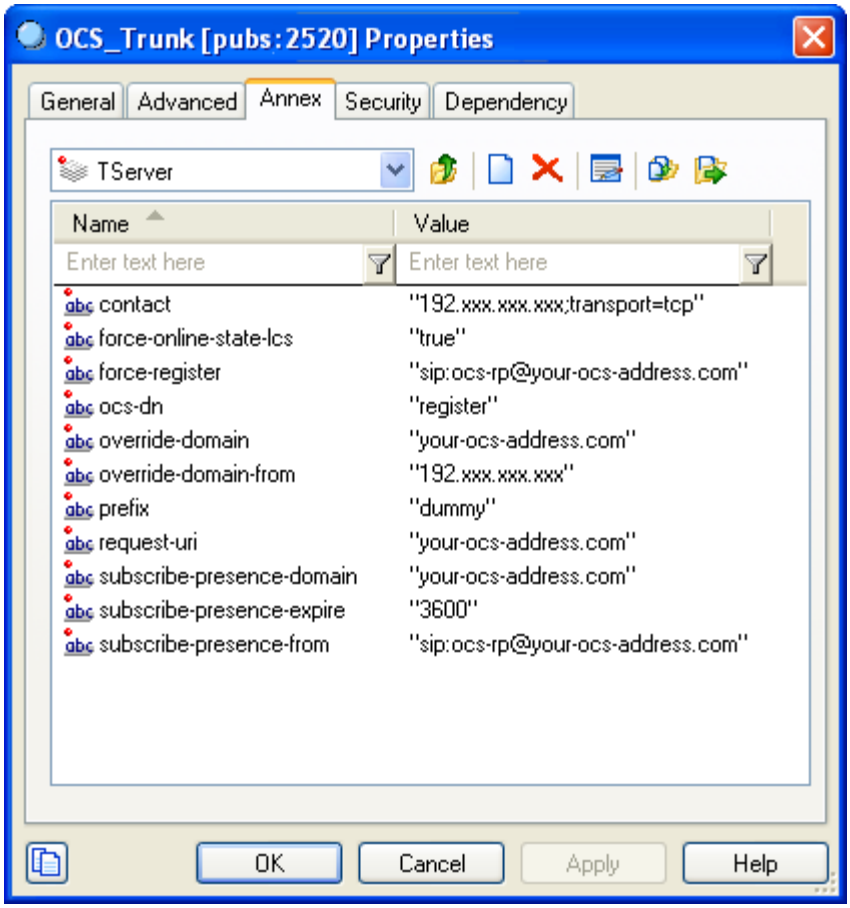

Figure 32: Sample OCS Trunk DN

Sample OCS Routing Point DN

Figure 33 shows a sample screenshot of a configured OCS `Routing Point` DN.

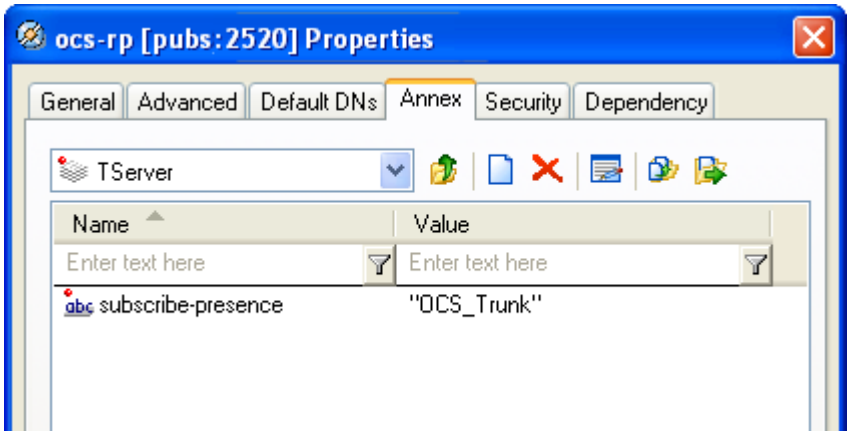

Figure 33: Sample OCS Routing Point DN

Sample OCS Expert User DN

Figure 34 shows a sample screenshot of a configured OCS expert user `Extension` DN.

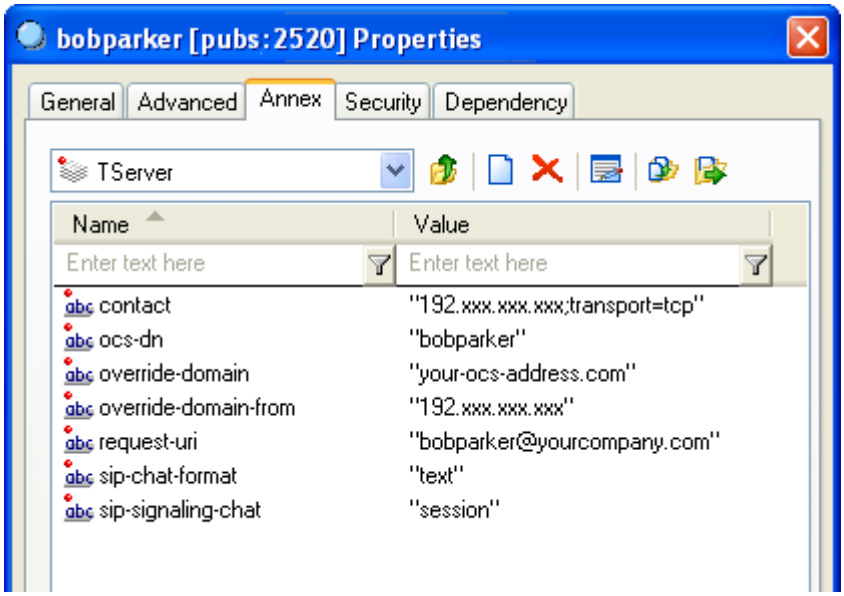

Figure 34: Sample OCS Expert User DN

Feature Limitation

- 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 Route Point user account as configured in the OCS 2007 Active Directory.

- If additional visibility and access controls are necessary, add all agents that will be authorized to view the status of the Routing Point as contacts.

These steps are required in order to make changes to visibility. For a simpler solution, consult Microsoft documentation.

- Due to a limitation on Microsoft, 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 `RouteCall` request to a DN that is configured with the `preview-interaction` option (page 350) set to `true`.

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 (page 309).
- 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`

(page 350) option to chat. For more information, see "Preview Interaction" on page 172.

Private Headers

SIP Server may include private, Genesys-proprietary 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.

Providing Call Participant Info

SIP Server can distribute information about all call participants—except the trunks allocated for communication between SIP Servers, distribution devices (such as Routing Points or ACD), and supervisors' DNs—to logged-in agents by using the SIP NOTIFY method and EventUserEvent messages.

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.

Feature Configuration

The following task summary provides an overview of the steps required to configure call info for agents.

Task Summary: Configuring Call Info for Agents

Objective	Related Procedures and Actions
1. Configure the Trunk DNs.	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 238.
2. Configure the SIP Server Application object.	Set the sip-enable-call-info option to true.

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, and TRouteCall.

  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. This Extension is not applicable when performing a TMakeCall request using the REFER method.
- The cpn option (page 336) 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.

Quality of Service

SIP Server can set quality-of-service (QoS) bits to a user-defined value in order to prioritize SIP signaling traffic. Use the option sip-ip-tos (page 315) 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, 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.

Remote Media on Genesys SIP Endpoint SDK 8.0

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 8.0:

- 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 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

The following table shows the configuration needed to enable this feature.

Task Summary: Enabling Remote Media Control on SIP Server

Objective	Key Procedures and Actions
• Configure the DN.	Go to SIP Server switch > DNs folder > SIP endpoint DN > Annex tab > TServer section and 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.
• 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 8.0, consult the *SIP Endpoint SDK 8.0 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 (page 340) 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 "Configuring endpoints" on page 65 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 (see page 359).

The SIP endpoint must support the BroadSoft Application Server interface in order to use the Remote Talk feature for remote call control.

SIP Traffic Monitoring

SIP Server actively monitors the level of SIP traffic that it receives, in order 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 167).
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).
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

The following table describes the options required to configure SIP traffic monitoring.

Task Summary: 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. In the `Voice over IP Service` DN, 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. SIP Server adds the sum of these two options. The maximum sum out of all configured DNs is used as the timeout for SIP traffic monitoring. For details, see "Endpoint Service Monitoring" on page 167.

Task Summary: Configuring SIP Traffic Monitoring (Continued)

Objective	Key Actions and Procedures
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 causing 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 do not enable the `sip-pass-check` option, as any oos-monitored DNs would be unavailable or non-operational during this period.

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 8.0 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

The following table describes the options required to configure TLS for the SIP messaging that SIP Server engages in.

Task Summary: Configuring TLS for SIP communication

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 8.0 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 8.0 Security Deployment Guide*.

Task Summary: Configuring TLS for SIP communication (Continued)

Objective	Key Actions and Procedures
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` (for Solaris, Linux, and AIX) 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 to true and SIP Server requests client certificate to initiate mutual TLS connection. • `sip-tls-crl`—Set this to the file name containing one or more certificates in PEM format, in order to define the **Certificate Revocation List**. • `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.

Task Summary: Configuring TLS for SIP communication (Continued)

Objective	Key Actions and Procedures
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-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.

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;tls=transport`

Trunk Capacity Support

SIP Server enables control of the number of calls to be handled by a specific Voice over IP device represented in the SIP Server configuration as `Trunk`. The following DN-level configuration options support this feature: `capacity` and `capacity-group`. See the option descriptions and configuration examples on pages 332–pages 333.

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, in order to prevent SIP Server from incorrectly treating an external call as internal.

Duplicated DN Names

If two SIP Server instances at different sites share the same DN number, , 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` (page 291) 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 be excluded from resource matching also.

To control this behavior, configure override-domain-from (page 348) 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

The following task summary provides an overview of the main steps required to configure tromboning control.

Task Summary: Configuring Tromboning Control

Objective	Related Procedures and Actions
1. Configure basic anti-tromboning.	In the SIP Server Application object, in the TServer section of the Options tab, configure the following option: • enforce-external-domains—Enter a comma-separated list of hostnames or IP addresses for each external SIP or T-Server from which you expect to receive ISCC transfers.

Task Summary: Configuring Tromboning Control (Continued)

Objective	Related Procedures and Actions
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 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 35 and 36 show the state of the call before and after the multi-site transfer.

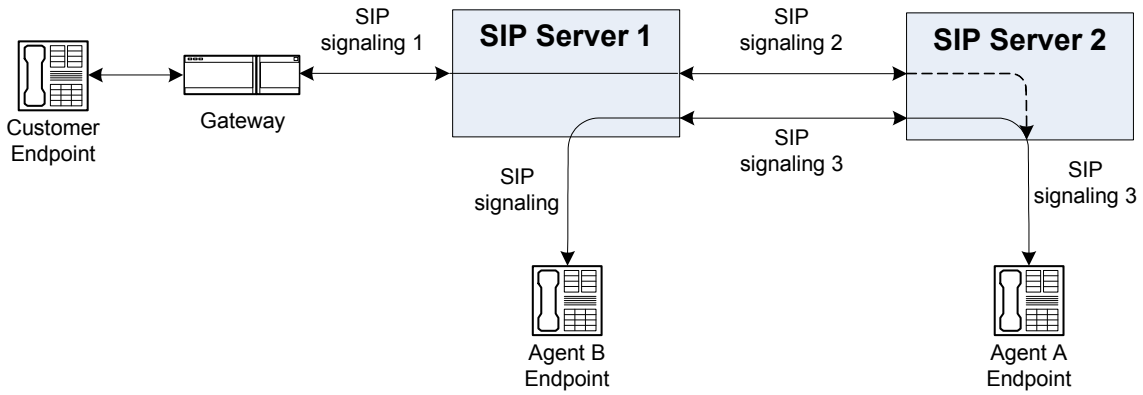

Figure 35: Call Before REFER with Replaces Transfer

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.

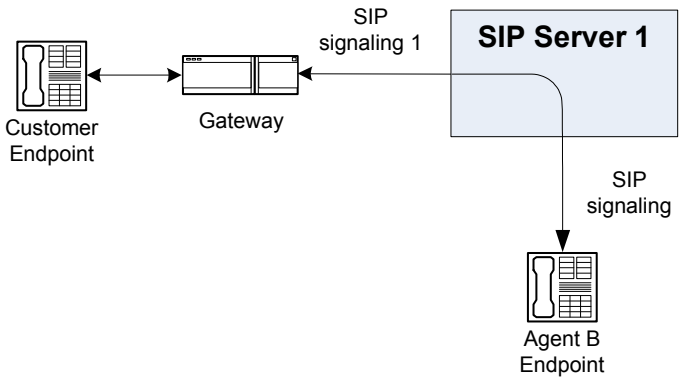

Figure 36: Call After REFER with Replaces Transfer

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 37 and 38 show the state of the call before and after the multi-site transfer.

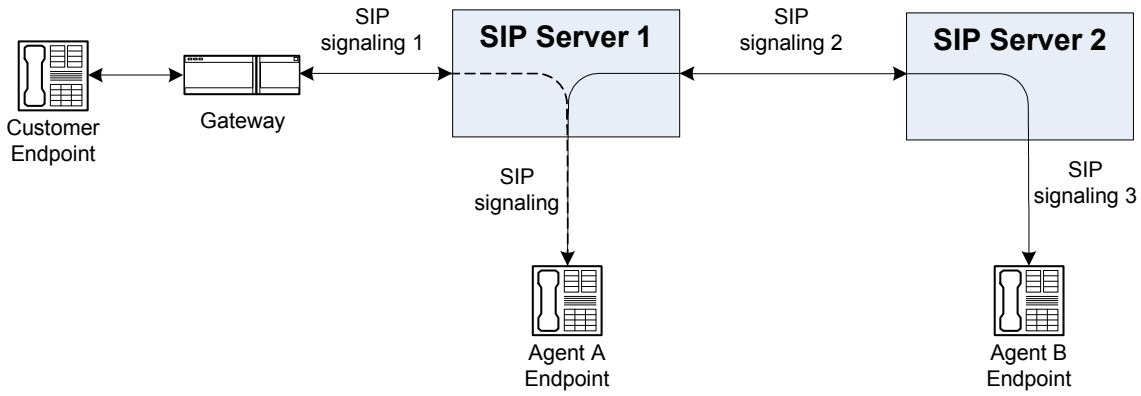

Figure 37: Call Before INVITE with Replaces Transfer

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.

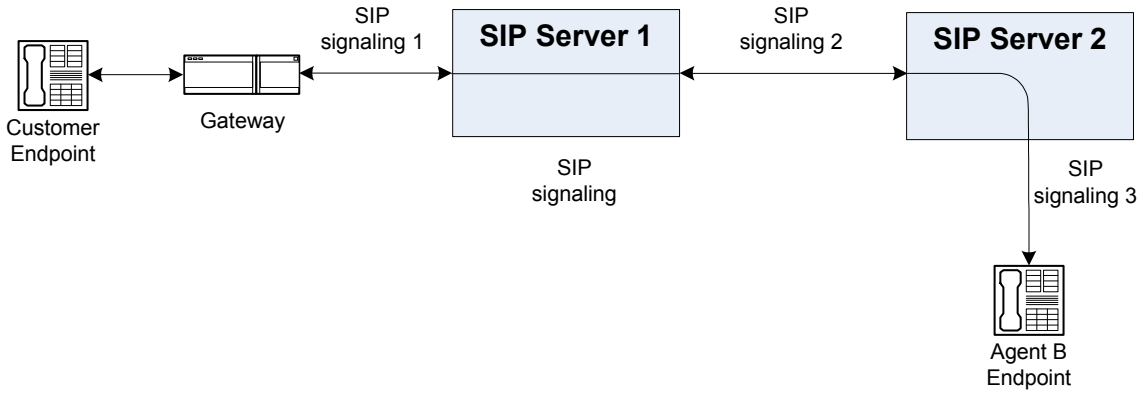

Figure 38: Call After INVITE with Replaces Transfer

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 39 and 40 show the state of the call before and after the multi-site transfer.

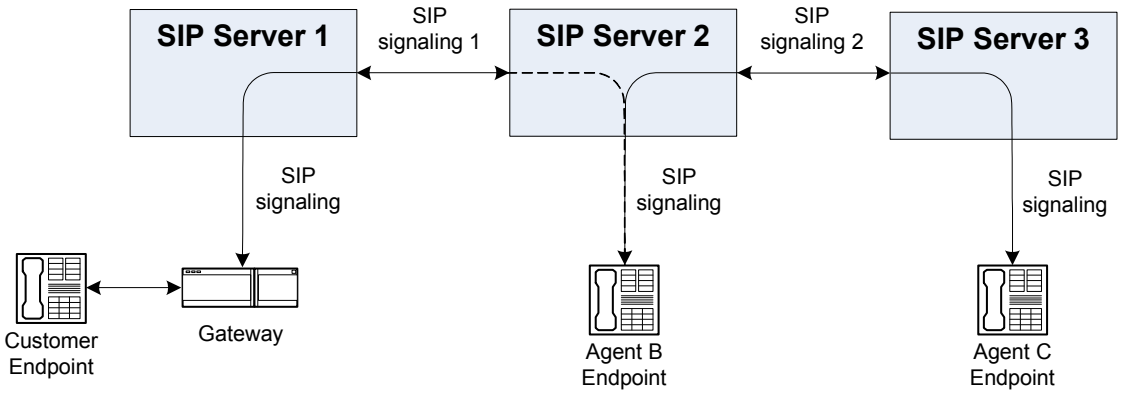

Figure 39: Call Before INVITE with Replaces Transfer

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.

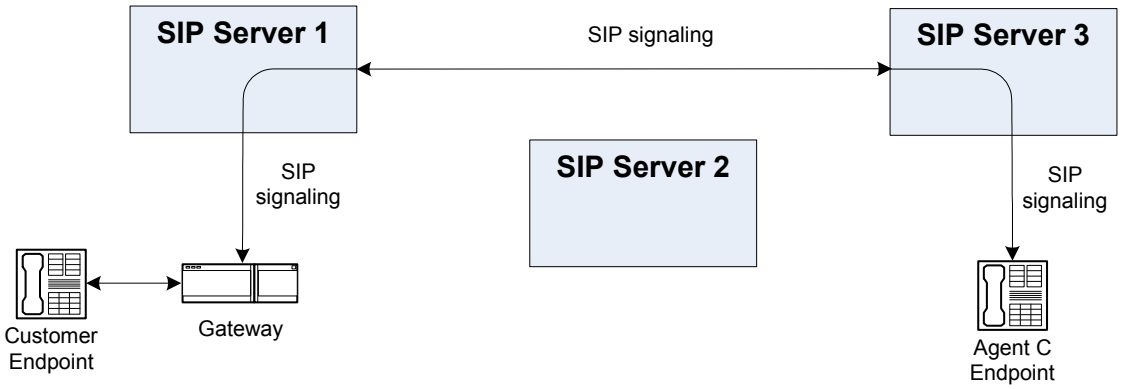

Figure 40: Call After INVITE with Replaces Transfer

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.

Feature Configuration

The following task summary provides an overview of the main steps required to configure trunk optimization.

Task Summary: 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 `Annex` tab, configure the following options: • `refer-enabled`—Set this option to `false` (see page 354). 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` (see page 347). • `sip-server-inter-trunk`—Set this option to `true` (see page 363).

Task Summary: Configuring Trunk Optimization (Continued)

Objective	Related Procedures and Actions
2. Configure the SIP Server `Application` object.	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.

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 Stream Manager or 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 (page 287).
- `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

Stream Manager or 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. Stream Manager or 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, Stream Manager or 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

The following task summary provides an overview of the main steps required to configure video support.

Task Summary: 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. If using the Genesys SIP Endpoint, launch the Audio and Video Tuning Wizard. Complete the wizard steps, and select the installed video camera on the corresponding wizard page.
3. Configure a SIP Server `Application` object.	In the SIP Server `Application` object, in the `TServer` section on the `Options` tab, 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, complete the following procedure: • Procedure: Configuring a gcti::video device, on page 245
5. Configure a video service.	For Push Video, complete the following procedure: • Procedure: Configuring a video service, on page 246

Complete the following procedures using Configuration Manager.

Procedure: Configuring a gcti::video device

Start of procedure

1. Under a configured `Switch` object, select the `DNs` folder. From the main menu, select `File` > `New` > `DN` to create a new `DN` object.
2. In the `New DN Properties` dialog box, click the `General` tab, and then specify the following properties:
   a. `Number`: Enter `gcti::video`.

   b. Type: Select Trunk from the drop-down box.

3. When you are finished, click Apply.

End of procedure

Next Steps

- Procedure: Configuring a video service

Procedure:
Configuring a video service

Start of procedure

1. Under a configured Switch object, select the DNs folder. From the main menu, select File > New > DN to create a new DN object.
2. In the New DN Properties dialog box, click the General tab, and then specify the following properties:
   a. Number: Enter the DN name. This name is currently not used for any messaging, but it must still be unique.
   b. Type: Select Voice over IP Service from the drop-down box.
3. Click the Annex tab, and create the TServer section. Under the TServer section, set the options as specified in Table 26.

Table 26: Configuring a Video Service

Option Name	Option Values	Description
contact	SIP URI	Specifies the value using the Stream Manager or Genesys Media Server application settings in the following format: IP address:SIP port
request-uri	SIP URI	Specifies 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	video	Set this option to video.

4. When you are finished, click `Apply`.

End of procedure

Feature Limitations

The Video Conference functionality is not supported with Stream Manager. However, Video Conference with active speaker detection is available for integrations with Genesys Media Server.

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 in order 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" below.

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.

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 Configuration Manager, 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.

Feature Configuration

The following task summary provides an overview of the main steps required to define the device-selection process—which priorities in the algorithm will be considered when selecting a device.

Task Summary: 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.
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 `Voice over IP Service` and `Trunk` DNs at a particular premise with the same `geo-location` value. 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 `Extension` attribute with the same value as the DNs at the same premise. **Note**: This method takes precedence over `geo-location` configured on the DN-level. For more information, see Procedure: Setting the geo-location for a call, on page 251

Task Summary: Defining the Device-Selection Process

Objective	Related Procedures and Actions
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`
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`—On the `Annex` tab, in the `TServer` section, set this option to the desired priority level. For details, see the option description on page 351.
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. For details, see the option descriptions on pages 332 and 333, respectively.

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):

- `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.

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.

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 `Annex` 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 `Annex` 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 `Annex` 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

c. 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

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 Stand-Alone—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 8.0 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.

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 81, as well as the *Genesys Media Server 8.1 Deployment Guide*.

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, page 255
- Using the Extensions Attribute, page 264
- Error Messages, page 272
- Known Limitations, page 274

Using T-Library Functions

Table 27 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 27, 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 7 Events and Models Reference Manual* and the *Voice Platform SDK 7.6 .NET (or Java) API Reference* for complete information on the T-Server events, call models, and requests.

Table 27 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 27 are reserved for Genesys Engineering and are listed here merely for completeness of information.

Table 27: Supported Functionality

Feature Request	Request Subtype	Corresponding Event(s)	Supported
General Requests			
TOpen Server		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 27: 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]		EventHeld, EventDialing*, EventReleased, EventPartyChanged	N
TSingleStepTransfer[b]		EventReleased*, EventPartyChanged	Y
TSingleStepConference		EventRinging*, EventEstablished	Y

Table 27: 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 27: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
	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 MultiFrequency) 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 27: 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[f]		EventMuteOff	Y
TSetMuteOn[f]		EventMuteOn	Y
TListenDisconnect		EventListenDisconnected	N
TListenReconnect		EventListenReconnected	N
TSetDNDOn		EventDNDOn	Y
TSetDNDOff		EventDNDOff	Y
TSetMessageWaitingOn		EventMessageWaitingOn	N
TSetMessageWaitingOff		EventMessageWaitingOff	N
Query Requests			
TQuerySwitch[a]	DateTime	EventSwitchInfo	N
	ClassifierStat		N

Table 27: 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 27: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TQueryLocation[a]	AllLocations	EventLocationInfo[g]	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[h]			
TNetworkConsult		EventNetworkCallStatus	N
TNetworkAlternate		EventNetworkCallStatus	N

Table 27: Supported Functionality (Continued)

Feature Request	Request Subtype	Corresponding Event(s)	Supported
TNetworkTransfer		EventNetworkCallStatus	N
TNetworkMerge		EventNetworkCallStatus	N
TNetworkReconnect		EventNetworkCallStatus	N
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 `TSetMuteOn` and `TSetMuteOff` only for established conferences, to allow for service observing

g. Two subtypes are supported by `EventLocationInfo`: `LocationInfoLocationMonitorCanceled` and `LocationInfoAllLocationsMonitorCanceled`.

h. 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 7 Events and Models Reference Manual* and the *Voice Platform SDK 7.6 .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 28 are also supported.

Table 28: Use of the Extensions Attribute

Extension		Used In	Description
Key	**Type**		
Call Recording			
record	String	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" on page 97 for details.
Call Supervision			
MonitorMode	String	TMonitorNextCall	Specifies the monitor mode: • `mute, normal`—A mute connection. • `connect`—A three-party conference call. • `coach`: Only the agent can hear the supervisor (whisper coaching). See "Call Supervision" on page 101 for details.

Table 28: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	**Type**		
MonitorScope	String	TMonitorNextCall	Specifies the required intrusion/observation scope: • `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.
AssistMode	String	TSingleStep-Conference	Specifies the required assistance mode: • `connect`—This is the default value - a three-party conference call. • `coach`—Only the agent can hear the supervisor (whisper coaching).
Dummy SDP			
sdp-c-host	String	TRouteCall	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.

Table 28: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	**Type**		
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. • The actual port used can equal this low value. • Only even-numbered ports are used.
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. • 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.
Music			
music	String	THoldCall TAlternateCall TInitiateTransfer TInitiateConference	Specifies the file name for music on hold for the call. See "Playing Multiple Media Files" on page 214.

Table 28: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	**Type**		
No Answer Supervision			
NO_ANSWER_TIMEOUT	String	TRouteCall	If set, the value of this Extension 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 200 for details.
NO_ANSWER_ACTION	String	TRouteCall	If set, the value of this Extension overrides any value set in any of the following configuration options for the current call: • `no-answer-action` • `agent-no-answer-action`
NO_ANSWER_OVERFLOW	Comma-separated list	TRouteCall	If set, the value of this Extension 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`
Call Progress Analysis			
AnsMachine	String	TMakePredictiveCall	If set, the value of this Extension (can only be set to `drop`) overrides the application-level parameter `am-detect`. If the CPD result shows that the predictive call reached an answering machine, SIP Server drops the call.

Table 28: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	**Type**		
FaxDest	String	TMakePredictiveCall	If set, the value of this Extension (can only be set to `drop`) overrides the application-level parameter `fax-detect`. If the CPD result shows that the predictive call reached a fax machine, SIP Server drops the call.
Providing Call Participant Info			
LCTPartiesLength	Integer	EventUserEvent	If set, the value of this Extension specifies how many parties are involved in a single call. See "Providing Call Participant Info" on page 227 for details.
LCTParty<n>	Integer	EventUserEvent	If set, the value of this Extension represents a party of the call, where *n* is an integer value starting from `0`. See "Providing Call Participant Info" on page 227 for details.
Providing Caller ID			
CPNDigits	String	TMakeCall, TMakePredictiveCall, TInitiateConference, TInitiateTransfer	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 228 for details.

Table 28: Use of the Extensions Attribute (Continued)

Extension		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 112 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 28: Use of the Extensions Attribute (Continued)

Extension		Used In	Description
Key	**Type**		
Routing			
after-routing-timeout	Integer	TRouteCall	An integer that overrides the value of the `after-routing-timeout` configuration option (page 278), 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.
T-Server Common Part Extensions			
EventServerInfo	sdn-licenses-in-use	integer	Specifies how many SDN licenses are currently in use.
	sdn-licenses-available	integer	Specifies how many SDN licenses are currently available.

Table 28: Use of the Extensions Attribute (Continued)

Extension		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 242 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 242 for details.

Error Messages

Table 29 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 29: 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.
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.
93	TERR_DEST_INV_STATE	Invalid destination state.
96	TERR_CANT_COMPLETE_CONF	Call cannot add new conference party.
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.

Table 29: Error Messages for SIP Server (Continued)

Code	Symbolic Name	Description
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.
177	TERR_TARG_DN_INV	DN target (in route call) was invalid.
195	TERR_CFW_DN_INV	Call forwarding address is invalid.
243	TERR_CLNT_NOT_MON	Internal error—client corrupted in T-Server.
259	TERR_INV_PASSWD	Invalid credentials (login_id or password).
302	TERR_INV_DTMF_STRING	DTMF string invalid.
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
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.
1605	TERR_INVALIDPARTY	Party in request was invalid on switch.
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.
3002	TERR_PRIVVIOLATION	User doesn't have security privilege on the switch.
3005	TERR_UNSUCC_ROUTECALL	`Routecall` request was unsuccessful.

Known Limitations

Several known limitations result from the current SIP Server and softswitches/gateways interface:

1. 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.
2. SIP Server supports `TSetMuteOn` and `TSetMuteOff` only for established conferences, to allow for service observing.
3. 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 in integrations with the BroadWorks softswitch, 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.
4. You must configure the Outbound Contact Solution to use a single-step transfer when you use `Transfer` Mode. SIP Server does not support consultation calls when it is working with the Outbound Contact Solution.
5. A single-step conference from an existing call or an existing conference to an external destination is not supported.
6. An `EventReleased (switch::)` message is issued when the last internal party leaves a call.
7. A Network Attended Transfer and Conference is not supported when SIP Server is the originating server.
8. SIP Server allows an agent to complete a consultation transfer of a call only if the call is located on one of the following:
   - A `Routing Point` DN.
   - A device in a ringing state, if the device supports an offer/answer exchange using the `UPDATE` method.

   Third-party call control (3pcc) blind conference calls are not supported.

9. The `TDeleteFromConference` request is not supported for first-party call control (1pcc) conference calls.
10. SIP Server does not report a first-party call control (1pcc) conference with mixing on an endpoint.

Third-Party Equipment—Known Limitations

The known limitations when SIP Server is operating with a third-party equipment are as follows:

1. The Siemens HiPath 8000 switch is supported with the following limitations:
   - 1pcc (first-party call control) calls are not supported.
   - The remote answer feature (`TAnswerCall`) is not available with version 2.0. However, the remote answer feature is supported with version 2.2.
   - Call scenarios containing the `REFER` message are not supported with this switch. 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 `Annex` tab of the DN.
2. The following media gateways support `re-INVITE`-based call transfers only:
   - Alcatel 7515
   - Cisco A5350
   - Cisco A5400
   - Asterisk
   - Sonus
3. 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.
4. When integrated with BroadSoft BroadWorks Release 13, SIP Server does not distribute an `EventOutOfService` message if a SIP endpoint is unplugged, because of the lack of any BroadSoft notification. The event error results in subsequent attempts to make a 3pcc call, until the device is plugged back in.

Chapter

7 SIP Server Configuration Options

This chapter describes the configuration options that are unique to SIP Server and contains the following sections:

- Application-Level Options, page 277
- Agent Login–Level and DN-Level Options, page 330
- GVP Integration Options, page 368
- Reserved Options, page 369
- Changes from Release 7.6 to Release 8.0, page 370

SIP Server also supports common log options described in Chapter 10 on page 449 and options common to all T-Servers described in Chapter 11 on page 469. Please refer to the *Framework 7.6 Stream Manager Deployment Guide* if you need to configure Stream Manager for use with SIP Server.

Application-Level Options

Configuration options specific to SIP Server functionality are set in Configuration Manager, in the corresponding sections on the `Options` tab of the SIP Server `Application` object.

T-Server 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 162

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 he or she was in 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.

after-routing-timeout

Default Value: 10
Valid Values: Any integer
Changes Take Effect: Immediately

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 specified value, the EventError message is issued. It will contain the Reference ID of the TRouteCall request.

Notes: 1. Set the value of the after-routing-timeout option less than the value of the rq-expire-tmout option.

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.

2. You can override this option using the Extensions attribute as applied to the TRouteCall. Configure the routing strategy to include the after-routing-timeout key-value pair to the Extensions attribute of the TRouteCall.

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 Annex tab of the DN object.

2. In the `TServer` section of the `Options` tab of the SIP Server `Application` object.

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 200

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 264 for more information about how this option is used.

SIP Server obtains the value for this option in the following order of precedence:

1. In the `TServer` section of the `Annex` tab of the `Agent Login` object.
2. In the `TServer` section of the `Options` tab of the SIP Server `Application` object.

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 `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 `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 value set when the `agent-no-answer-timeout` option expires.

Changes Take Effect: Immediately
Related Feature: "No-Answer Supervision" on page 200

Specifies a sequence of overflow destinations (separated by 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 264 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. In the TServer section of the Annex tab of the Agent Login object.
2. In the TServer section of the Options tab of the SIP Server Application object.

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 200

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.

When 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 264 for more information about how this option is used.

SIP Server obtains the value for this option in the following order of precedence:

1. In the TServer section of the Annex tab of the Agent Login object.
2. In the TServer section of the Options tab of the SIP Server Application object.

agent-only-private-calls

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately
Related Feature: "Emulated Agents" on page 162

Specifies whether SIP Server blocks the classification of a call's business type as private when there is no agent on the call. When set to false, calls with no agents present are classified with business type private, enabling No-Answer Supervision (NAS) to be applied for private calls.

When set to true, calls remain classified with business type unknown and NAS is not applied to such calls.

agent-strict-id

Default Value: false

Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Emulated Agents" on page 162

Specifies whether, for agents, SIP Server enables any `AgentID` to be used during login (value `false`), or only those configured in the Configuration Layer (value `true`).

am-detected

Default Value: `drop`
Valid Values:

`drop`	The call is released.
`connect`	The connected call stays connected.

Changes Take Effect: Immediately

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, G711, 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: Immediately
Related Option: `sip-enable-sdp-codec-filter` on page 314

Specifies a list of codecs that SIP Server uses to modify the 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 in the initial `INVITE` to the caller for a 3pcc `MakeCall` operation.

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: 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 Agent Logout" on page 97

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 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 (`Annex` tab)
- DN object (`Annex` tab of the `ACD Position` or `Extension` DN) that represents the device to which the agent is logged on.
- DN object (`Annex` tab of the `Routing Point` or `ACD Queue` DN) that represents the queue to which here the agent is logged on.
- SIP Server `Application` object (`Options` tab), 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 Agent Logout" on page 97

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 (`Annex` tab)
- DN object (`Annex` tab of the `ACD Position` or `Extension` DN) that represents the device to which the agent is logged on.
- DN object (`Annex` tab of the `Routing Point` or `ACD Queue` DN) that represents the queue to which the agent is logged on.
- SIP Server `Application` object (`Options` tab), which specifies the server-wide default.

backwds-compat-acw-behavior

Default Value: `false`

Valid Value: `true, false`
Changes Take Effect: Immediately
Related Feature: "Emulated Agents" on page 162

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 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 makes or 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.

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 his or her previous `NotReady` state. Note that no legal-guard time is applied, because the agent does not go into the `Ready` state.

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 that 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 that is played to a party because of a busy condition or an error.

call-rq-gap

Default Value: `0`
Valid Value: Any integer from `0`–`1000`
Changes Take Effect: Immediately

Specifies (in milliseconds) the length of delay applied to a request issued against a busy call (a call that has another request working on it already). This prevents race conditions on the different call legs.

Note: Genesys recommends that you set this option to `0`.

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 101

Specifies whether call supervision subscription is canceled when the client that requested it disconnects from SIP Server.

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 the noncompletion tone played during DTMF digit collection.

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 126, and “Dial Plan” on page 141

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 126
- “Dial Plan” on page 141

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

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 on 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 as an `Extension` in the `TMakePredictiveCall` request, the value of this key will override the value of the `cpd-info-timeout` option.

default-dn

Default Value: NULL
Valid Value: Any valid DN
Changes Take Effect: Immediately

Specifies the DN to which calls are sent when URS is nonoperational, or when the timeout specified in the router-timeout option expires.

Note: You can also use this option for emergency ACD routing.

This option can be set as 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.

default-route-point

Default Value: No default value
Valid Value: Any valid DN
Changes Take Effect: For the next call

Specifies the DN to which an inbound call is routed 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 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, TInitiateTransfer, and so on.

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 101

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 101

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: “Playing Multiple Media Files” on page 214

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 both the Application and the DN level. The DN-level option takes precedence.

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 242

Contains the name of the video file that will be 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: For the next call
Related Feature: “Dial Plan” on page 141

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:

- Agent Login—Applies to calls made by a caller logged in under this `Agent Login` ID.

- 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.
- Application-level—Applies to all calls (where no `Agent Login` or DN dial-plan is defined).

divert-on-ringing

Default Value: `true`
Valid Values:

`true`	SIP Server generates `EventRouteUsed` and `EventDiverted` messages when the SIP `180 Ringing` response 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 `rq-expire-tmout` option, the destination SIP dialog is canceled and an `EventError` message is generated.

Changes Take Effect: Immediately

Determines SIP Server behavior when routing calls.

dtmf-payload

Default Value: `101`
Valid Values: Any integer from `0`–`128`
Changes Take Effect: Immediately

Specifies the value to be used in negotiations for the DTMF payload type, when Stream Manager release 7.0.2 (or earlier) treatments are used.

Note: This option is obsolete. It can be used for backward compatibility with the previous SIP Server releases.

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 162

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. In the `Agent Login` object, on the `Annex` tab in the `TServer` section.
2. In the `DN` object which represents the device, on the `Annex` tab in the `TServer` section.
3. In the `DN` object which represents the Agent Group, such as an ACD Queue, on the `Annex` tab in the `TServer` section.
4. In the SIP Server `Application` object, on the `Options` tab in the `TServer` section.

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 the 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 `RequestAgentReady` or `RequestAgentNotReady` 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 ID taken from the `Request-URI` in the `SUBSCRIBE` request.

Note: This functionality has been verified using the Polycom SoundPoint IP 501/500 Phone.

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 Configuration Manager. 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 177

This option activates Unicode support for the SIP-to-TLib and TLib-to-SIP mapping functionality. By default, Unicode support is disabled. To activate this

functionality, 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 call attached data or `AttributeExtensions` of a `TRouteCall` request encoded in a local character to the remote destination using SIP messages.

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 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: Next call
Related Feature: “Network Asserted Identity” on page 192

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.

Note: To establish the trust-level for individual DNs, see the DN-level option `enforce-trusted` on page 340.

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 `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 in order 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 only 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 `extn-no-answer-timeout` expires.
recall	SIP Server returns the call to the last distribution device (the device reported in the `ThisQueue` attribute of the call) when `extn-no-answer-timeout` expires.
`release`	SIP Server releases the call.
Any valid overflow destination	SIP Server returns the call to the specified destination when the value set for the `extn-no-answer-timeout` option expires.

Changes Take Effect: Immediately

Related Feature: "No-Answer Supervision" on page 200

Specifies a sequence of overflow destinations (separated by 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 264 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, T-Server skips to the next destination in the list.

SIP Server obtains the value for this option in the following order of precedence:

1. In the `TServer` section of the `Annex` tab of the `DN` object of type `Extension`.
2. In the `TServer` section of the `Options` tab of the SIP Server `Application` object.

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 200

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`.

When set to `0`, the No Answer Supervision feature for `DN`s of type `Extension` is disabled. See the `NO_ANSWER_TIMEOUT` extension in section "Using the Extensions Attribute" on page 264 for more information about how this option is used.

SIP Server obtains the value for this option in the following order of precedence:

1. In the `TServer` section of the `Annex` tab of the `DN` object of type `Extension`.
2. In the `TServer` section of the `Options` tab of the SIP Server `Application` object.

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 stays connected.

Changes Take Effect: Immediately

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.

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 its 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 were 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: Immediately

Determines SIP Server behavior for choosing a gateway or trunk for the outbound call.

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.

greeting-after-merge

Default Value: `false`
Valid Values: `false, true`
Changes Take Effect: Immediately
Related Feature: "Personal Greeting" on page 211

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: An empty string
Valid Values: `internal, consult` (one or both, in any order)
Changes Take Effect: Immediately
Related Feature: "Personal Greeting" on page 211

This option specifies—using a space-, comma- or semicolon-separated list—the types of calls to which a greeting will not be played. The default value (an empty string) means that a greeting will be played to all calls (8.0.2 backward compatibility support). If the option is set to `internal` and/or `consult`, a greeting will not be played to internal and/or consultation calls.

greeting-delay-events

Default Value: `false`
Valid Values: `false, true`
Changes Take Effect: Immediately
Related Feature: "Personal Greeting" on page 211

This option 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.

greeting-notification

Default Value: An empty string

Valid Values: `started`, `complete` (one or both, in any order)
Changes Take Effect: Immediately
Related Feature: "Personal Greeting" on page 211

This option 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 Greeting" on page 211

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.

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 `genesyslab.com`, and `ims-3pcc-prefix` is set to `3pcc`, then 1pcc calls will use the URI:
`sip+1234@genesyslab.com`

while 3pcc calls will use the URI
`sip:+1234@3pcc.genesyslab.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-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-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.

inbound-bsns-calls

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Business-Call Handling" on page 162

Specifies whether SIP Server considers all established inbound calls on an agent as business calls.

info-pass-through

Default Value: No default value
Valid Values:

Value	Description
`*`	Allows all `INFO` messages to be sent to the remote party.
`-`	Disables all `INFO` messages from being sent to the remote party.
`<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 remote party.
`*,<list>`	Allows all `INFO` messages to be sent to the remote party, 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 through to the remote device, to disable all `INFO` messages from being sent to the remote device, 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 remote party, except for the following:

- `application/vnd.radisys.msml+xml`
- `application/x-www-form-urlencoded`

- `application/x-detect` (not sent if predictive call is in progress, or if no AudioCodes CPD result header is found)

Note: You can define this option at both the Application and the DN level. 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 162

Determines whether a consultation call that is made from a business primary call contains the `business call` attribute.

internal-bsns-calls

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Business-Call Handling" on page 162

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 semi-colon (;)
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. For example, if DN `4813` is configured in Configuration Manager, DN `4814` is not configured, and the internal registrar is enabled, 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 `Annex` tab of the corresponding DN.

In `hot standby` configuration, set the `internal-registrar-persistent` option to `true` to enable Configuration Server to propagate changes of the `contact` information to the backup SIP Server.

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 101

Determines SIP Server behavior when a `TMonitorNextCall` request is submitted while the monitored agent is on a call.

legal-guard-time

Default Value: `0`
Valid Value: Any integer from `0`–`30`
Changes Take Effect: Immediately
Related Feature: "Emulated Agents" on page 162

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.

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 a `TMakeCall` request. 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

This option 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. When 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-legs-per-sm

Default Value: `0`
Valid Values: Any integer from `0`–`65000`
Changes Take Effect: Immediately for new legs. Previously created legs are not dropped.

Specifies the maximum number of legs that are created on each connected Stream Manager. If all connected Stream Managers already have the specified number of legs, no new leg is created.

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 101

Specifies SIP Server behavior to start monitoring sessions.

msml-support

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Outbound IP Solution Integration" on page 203

Enables SIP Server to engage 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 a 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 like (treatments, music, greetings and conferences) except recording 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`. If using NETANN service through Stream Manager, set this option to the default `false`.

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 123

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 in on 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 that 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 Multiple Media Files" on page 214

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 Stream Manager DN `request-uri` option is used instead. For more information about this parameter, see the *Framework 7.6 Stream Manager Deployment Guide*.

---

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

Related Feature: "Message Waiting Indicator Functionality" on page 95

Enables or disables MWI for the agent's voice mail box.

mwi-domain

Default Value: No default value
Valid Value: Any computer name
Changes Take Effect: During the next attempt to register for MWI
Related Feature: "Message Waiting Indicator Functionality" on page 95

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.

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
Related Feature: "Message Waiting Indicator Functionality" on page 95

Enables or disables MWI for the extension's voice mail box.

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
Related Feature: "Message Waiting Indicator Functionality" on page 95

Enables or disables MWI for the agent group's voice mail box.

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
Related Feature: "Message Waiting Indicator Functionality" on page 95

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.

mwi-mode

Default Value: `REGISTER`
Valid Values: `REGISTER, SUBSCRIBE`
Changes Take Effect: After SIP Server restart
Related Feature: "Message Waiting Indicator Functionality" on page 95

When this option is set to `SUBSCRIBE`, SIP Server activates SIP subscriptions for all voice mail box owners as configured by other `mwi-<>` options. When 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`.

mwi-port

Default Value: No default value
Valid Value: Any available port
Changes Take Effect: During the next attempt to register for MWI
Related Feature: "Message Waiting Indicator Functionality" on page 95

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.

nas-private

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "No-Answer Supervision" on page 200

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.

no-response-dn

Default Value: No default value
Valid Values: Any valid DN
Changes Take Effect: Immediately
Related Option: "sip-invite-timeout" on page 315

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`.

observing-routing-point

Default Value: No default value
Valid Values: A `Routing Point` DN
Changes Take Effect: For the next call
Related Feature: "Remote Supervision" on page 112

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. No routing strategy is required to be loaded on the observing Routing Point.

outbound-bsns-calls

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Business-Call Handling" on page 162

Specifies whether SIP Server considers all established outbound calls on an agent as business calls.

overload-ctrl-dialog-rate-capacity

Default Value: `400`
Valid Values: Valid integer between `0` and `10000`
Changes Take Effect: Immediately
Related Feature: "Overload Control" on page 207

Specifies the SIP Dialog Rate (dialogs/sec) that this instance of 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-call-rate-capacity

Default Value: `200`
Valid Values: Valid integer between `0` and `10000`
Changes Take Effect: Immediately
Related Feature: "Overload Control" on page 207

Specifies the Call Rate (calls/sec) that this instance of 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-threshold

Default Value: `0`
Valid Values: Valid integer between `0` and `65535`
Changes Take Effect: Immediately
Related Feature: "Overload Control" on page 207

Specifies the call rate (calls/second) after which the overload control feature becomes active. To disable the feature, set this option to the default `0`.

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.

parking-music

Default value: `music/silence`
Valid Values:
Changes Takes Effect: For the next parked call
Related Feature: "Remote Supervision" on page 112

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 `posn-no-answer-timeout` expires.
recall	SIP Server returns the call to the last distribution device (the device reported in the `ThisQueue` attribute of the call) when `posn-no-answer-timeout` expires.
`release`	SIP Server releases the call.
Any valid overflow destination	SIP Server returns the call to the specified destination when the value set for the `posn-no-answer-timeout` option expires.

Changes Take Effect: Immediately
Related Feature: "No-Answer Supervision" on page 200

Specifies a sequence of overflow destinations (separated by 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 264 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. In the TServer section of the Annex tab of the DN object of type ACD Position.
2. In the TServer section of the Options tab of the SIP Server Application object.

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 200

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.

When set to 0, the No Answer Supervision feature for DNs of type ACD Position is disabled. See the NO_ANSWER_TIMEOUT extension in section "Using the Extensions Attribute" on page 264 for more information about how this option is used.

SIP Server obtains the value for this option in the following order of precedence:

1. In the TServer section of the Annex tab of the DN object of type Position.
2. In the TServer section of the Options tab of the SIP Server Application object.

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 ApplyTreament, or if the router-timeout is activated for any new request coming in for the call. For example, a RouteCall 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 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 226

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.

Note: 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.

record-consult-calls

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately (established connections remain unaffected)
Related Feature: “Call Recording” on page 97
Related Options: record, recording-filename

Specifies whether consultation calls are eligible for regular call recording.

When 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).

When set to true, SIP Server allows recording of consultation calls that include at least one appropriately set DN.

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" on page 97

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.

registrar-default-timeout

Default Value: No default value
Valid Values: `1 - 3600`
Changes Take Effect: Immediately

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.

If `registrar-default-timeout` is not configured, SIP Server responds to the `REGISTER` request using a `200 Ok` with the same `Expires:` value as the one that came in with the `REGISTER`. If the option is configured, SIP Server responds to the `200 OK` with an `Expires:` value that equals the value of this option.

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.

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`.

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-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).

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-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-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-retain-timeout

Default Value: `1`
Valid Values: `0–3600`
Changes Take Effect: Immediately

Defines how long (in seconds) a T-Library call is kept in SIP Server's memory after all call parties (members of the call) are released. This option does not set any delay in generating an `EventReleased` message for a particular DN when the `BYE` message is received (or sent).

sip-dtmf-send-rtp

Default Value: `false`
Valid Values:

`true`	SIP Server instructs Stream Manager 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 page 156

Specifies whether SIP Server instructs Stream Manager to send DTMF tones when a T-Library client issues a `TSendDTMF` request.

sip-enable-call-info

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: "Providing Call Participant Info" on page 227

When set to `true`, SIP Server distributes the information about call participants to logged-in agents by using the SIP `NOTIFY` method and `EventUserEvent` messages.

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

Enables processing of `100rel` parameters and messages.

Note: SIP Server does not process reliable responses in the following scenarios:

- In third-party call control (`TMakeCall`) operations
- When re-`INVITE` methods are used
- When call routing is performed to any destination

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: “Playing Multiple Media Files” on page 214

Enables or disables music-on-hold..

Note: You can define this option at both the Application and the DN level. The DN-level option takes precedence.

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: Immediately
Related Option: `audio-codecs` on page 281.

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.

sip-hold-rfc3264

Default Value: `false`

Valid Value:

`true`	RFC3264-compliant implementation.
`false`	RFC2543-compliant implementation.

Changes Take Effect: Immediately

Specifies which implementation of hold media SDP is used by SIP Server for third-party call control (3pcc) hold operations.

Note: When this option is set at the `DN` level in the `Annex` tab > `TServer` section, it will override the `Application` level value.

sip-invite-timeout

Default Value: `0` (in effect, 32 seconds)
Valid Values: `0`–`34`
Changes Take Effect: Immediately

Specifies the number of seconds that SIP Server waits for a response to the `INVITE` message. The call times out if no response is received. When set to `0`, or if a value is not specified, then the default SIP call timeout of `32` seconds is used.

sip-invite-treatment-timeout

Default Value: `0` (in effect, 32 seconds)
Valid Values: `0`–`34`
Changes Take Effect: Immediately

Specifies the number of seconds that 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. When 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-ip-tos

Default Value: No default value
Valid Values: Any integer from `0-255`, either in decimal format or in hexadecimal format with `0x` prefix
Changes Take Effect: Immediately
Related Feature: “Quality of Service” on page 228

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). 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-link-type

Default Value: `0`
Valid Values: `0,3`
Changes Take Effect: After SIP Server restart
Related Feature: “Multi-Threaded Architecture” on page 31

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 as a single-threaded server, as in pre-8.0.3 releases.
- 1,2—Reserved for debugging.
- 3—Enables multi-threaded mode.

Note: This option is temporary to allow for backward compatibility for single-threaded mode. In future releases, this option will be removed.

sip-pass-check

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Features: “Endpoint Service Monitoring” on page 167, “SIP Traffic Monitoring” on page 231

This option enables tracking of SIP messages that reach the primary SIP Server, including responses from SIP devices (DNs) monitored by SIP Server using the `oos-check` and `oos-force` options.

The primary SIP Server summarizes results of the checks on DNs for out-of-service status and monitors the time passed since the last received SIP message. If the primary SIP Server does not receive SIP messages for a certain period of time, SIP Server reports the `SERVICE_UNAVAILABLE` status to LCA/SCS. The period of time is chosen as the maximum of sums (among the sums of the oos-check and oos-force option values, configured for service DNs). When SIP Server reports the `SERVICE_UNAVAILABLE` status to LCA/SCS, SCS switches the primary SIP Server over to the backup server, and it 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 in this case, 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-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 232

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

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.

Note: 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 setting at the Application-level.

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. When 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`

When 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 should be configured as a Trunk for supporting the above option in `REFER` message.

sip-respect-privacy

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: At the next call

Specifies the content SIP Server will report in the `AttributeANI` extension when an inbound `INVITE` message contains `P-Asserted-Identity` and `Privacy:id` headers. If this option is set to `false`, the content for `AttributeANI` is taken from the `From` header. If this option is set to `true`, the content for `AttributeANI` is taken from the `P-Asserted-Identity` 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`
Changes Take Effect: Immediately

With the option set to `0`, SIP Server connects Stream Manager to a call to play an audio ring tone. With the option set to `1`, SIP Server waits for a response from the called device, and connects Stream Manager to a call to play an audio ring tone *only* when the returned response cannot be used as the offer to a calling device.

Notes: SIP Server does not support internal ring tones in conference scenarios where the `sip-ring-tone-mode` option is set to `1`. In this case, SIP Server provides a ring tone 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-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 232

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 232

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 232

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 the following:

- 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 sigh (`+`) 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 the 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 30 lists the primary cipher aliases.

Table 30: 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 31 lists available cipher group aliases.

Table 31: 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 232

Specifies the name of the file that contains one 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.

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 232

Specifies whether SIP Server will requests the client certificate, initiating 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 232

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 232

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-treatments-continuous

Default Value: `false`
Valid Values:

`true`	A routing strategy treatment is played 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.

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: "Trunk Capacity Support" on page 235
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 88.

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: "Trunk Capacity Support" on page 235
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 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 88.

stranded-call-redirection-limit

Default Value: `4`
Valid Values: `0` to `15`
Changes Take Effect: Immediately
Related Feature: "Trunk Capacity Support" on page 235

Related Options: stranded-calls-overflow and stranded-on-arrival-calls-overflow

Limits the number of times that SIP Server tries to redirect any particular 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

This option 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-timeout

Default Value: 180
Valid Values: 1–3600
Changes Take Effect: Immediately

Specifies the time interval (in seconds) in the Expire header of the 200 OK response message to a subscriber.

timed-acw-in-idle

Default Value: true
Valid Values: true, false
Changes Take Effect: Immediately
Related Feature: “Emulated Agents” on page 162

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.

When 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.

tlib-map-replace-dn

Default Value: `false`
Valid Values: `true, false`
Changed 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.

unknown-bsns-calls

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Feature: “Business-Call Handling” on page 162

Determines whether SIP Server considers calls of unknown call type 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 162

Specifies the threshold (in seconds) at which the timing of ACW stops and manual intervention is required (*untimed ACW*).

user-data-im-enabled

Default Value: `false`
Valid Values: `true`, `false`
Related Feature: “Instant Messaging” on page 171

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 that 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-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 177

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.

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 that 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 the user part as 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.

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 162

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` (limited to ISCC scenarios).
2. In a `DN` configuration object of type `Routing Point`, on the `Annex` tab in the `TServer` section.
3. In a `DN` configuration object of type `ACD Queue`, on the `Annex` tab 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, on the `Annex` tab in the `TServer` section.
6. In a `DN` configuration object of type `Extension`, on the `Annex` tab in the `TServer` section.
7. In a `DN` configuration object of type `ACD Queue` or `Routing Point` that represents logged-in agents (`Agent Group`), on the `Annex` tab in the `TServer` section.
8. In the SIP Server `Application` object, on the `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).

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, and then a numeric value.

extensions-<n>

Default Value: No default value
Valid Values: See description
Changes Take Effect: Immediately
Related Feature: “Mapping SIP Headers and SDP Messages” on page 177

The `extension` prefix instructs SIP Server to put the SIP header (parameter) into the `Extensions` attribute.

The value determines which header within the header parameter of the SIP message is processed. The value of the header/header parameter is added as a key-value pair into the attribute as follows:

```
<header_name>=<header_value>
```

You can use the colon character to address the parameter name of a header. For example, the `extensions-8=From:tag` option puts the `From:tag=979E46B1-0FDF-418F-BA3F-C03F95A4D6E0-2` key-value pair into the `Extensions` attribute of the `EventRouteRequest` message.

userdata-<n>

Default Value: No default value
Valid Values: See description
Changes Take Effect: Immediately
Related Feature: “Mapping SIP Headers and SDP Messages” on page 177

The `userdata` prefix instructs SIP Server to put the SIP header (parameter) into the `UserData` attribute.

Log Section

There is one SIP Server-specific option available for the `Log` section. For the common `Log` options, see Chapter 10 on page 449.

x-sip-log

Default Value: No default value

Valid Values: file name

Changes Take Effect: After SIP Server restart

Specifies whether SIP Server creates a separate log file for SIP processing. If the option is specified, SIP Server creates a separate log file for SIP processing-related messages. T-Library specific messages will be sent to the main log file. Note that the SIP processing log file inherits only the following settings from the common log options: `expire`, `segment`, `verbose`.

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`.

SIP Error Map Section

This section must be called `SipErrorMap`.

sip-<SIP_error_code>

Default Value: No default value
Valid Values: `399 < integer < 700`
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 the following table for a list of available `TEvent` messages and their corresponding identifying number.

Table 32: TEvent Identifiers

Identifying Number	TEvent
3	CallStateGeneralError
4	CallStateSystemError
5	CallStateRemoteRelease
6	CallStateRemoteRelease
7	CallStateBusy
8	CallStateSitDetected
10	CallStateAllTrunksBusy
11	CallStateSitInvalidnum
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

You set configuration options described in this section in the `TServer` section on the `Annex` tab of the relevant `Agent Login` or `DN` object in Configuration Manager.

agent-greeting

Default Value: `NULL`
Valid Values: Any file name that will be played to the agent
Changes Take Effect: Immediately
Related Feature: "Personal Greeting" on page 211

Specifies the media file name that will be used as a greeting for the agent.

Note: 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 the other greeting, but immediately connects the customer and agent. No greetings are played.

audio-codecs

Default Value: `telephone-event, PCMU, PCMA, G711, 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: Immediately

Related Option: `sip-enable-sdp-codec-filter` on page 361

For a description of this option, see the application-level option "audio-codecs" on page 281.

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.

If neither `authenticate-requests` nor `password` configuration options are configured on the DN, no requests will be authenticated.

auto-logout-ready

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: Immediately
Related Feature: "Automatic Agent Logout" on page 97

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 (`Annex` tab)
- DN object (`Annex` tab of the `ACD Position` or `Extension` DN) that represents the device to which the agent is logged in.
- DN object (`Annex` tab of the `Routing Point` or `ACD Queue` DN) that represents the queue to which the agent is logged in.
- SIP Server `Application` object (`Options` tab), 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 Agent Logout" on page 97

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 (`Annex` tab)

- DN object (Annex tab of the ACD Position or Extension DN) that represents the device to which the agent is logged in.
- DN object (Annex tab of the Routing Point or ACD Queue DN) that represents the queue to which the agent is logged in.
- SIP Server Application object (Options tab), 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 of the Annex tab 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, that the 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.

capacity

Default Value: 0
Valid Values: Any positive integer
Changes Take Effect: Immediately

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

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. See the following option configuration examples.

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`.

Example 1 DN of type `Trunk` with the Number `Cisco-8340`

```
[TServer]
capacity=200
prefix=8340
```

With these settings, the number of calls to the `Cisco-8340` trunk will be limited to 200.

Example 2

- DN of type `Trunk` with the Number `Cisco-8340`

```
[TServer]
capacity=200
capacity-group=Cisco-GW
prefix=8340
```

- DN of type `Trunk` with the Number `Cisco-8341`

```
[TServer]
capacity-group=Cisco-GW
prefix=8340
```

With these settings, the number of calls to the `Cisco-8340` and `Cisco-8341` trunks will be limited to 200.

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:

Value	Description
0	Default. Charge type is not set. Charging is based on DN type.
1	Free. Charge type is free (including agent).
2	Paid. Charge type is paid (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 160

Related Trunk Options: “sip-early-dialog-mode” on page 359 and “sip-server-inter-trunk” on page 363

Specifies whether a charge will be incurred for services that are supplied by this 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 represents 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`.

contact

Default Value: No default value
Valid Values Any alphanumerical string
Changes Take Effect: Immediately

Contains the contact URI, specifying the device’s IP address, if this address is fixed. This option is necessary only for stand-alone 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 a SIP Server, and this request is accepted, SIP Server uses the contact information from the `REGISTER` request, and updates (or creates) in Configuration Manager the option `contact` in the `TServer` section of the `Annex` tab of the corresponding `DN` object.

The URI format is:

```
[sip:][number@]hostport[;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.
- `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.

contacts-backup

Default Value: No default value
Valid Values: A comma-separated list of any valid SIP URI
Changes Take Effect: Immediately

Specifies a list of alternative SIP URI addresses to be used in cases were the SIP URI specified in the `contact` option cannot be reached.

For integration with Cisco CallManager (CCM), 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: Immediately
Related Feature: “Class of Service” on page 126, and “Dial Plan” on page 141

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 126
- "Dial Plan" on page 141

cpn

Default Value: No default value
Valid Values: A name that will be used as a user part of the SIP URI
Changes Take Effect: Immediately

When the value is specified for a DN, it will be used as the user part of the SIP URI in the `From` header of the `INVITE` message sent by SIP Server to this DN. This option must not be configured on trunks that are allocated for direct signaling between SIP Servers.

Note: If the `CPNDigits` parameter is specified in the `Extensions` attribute in `TMakeCall`, `TMakePredictiveCall`, `TInitiateConference`, or `TInitiateTransfer` requests, it takes precedence over the `cpn` option setting.

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 -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 that will be played to the customer
Changes Take Effect: Immediately
Related Feature: "Personal Greeting" on page 211

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.

Note: When used in conjunction 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.

default-dn

Default Value: `NULL`
Valid Values: Any valid DN
Changes Take Effect: Immediately

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 be set at 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.

default-music

Default Value: No default value
Valid Value: Name and path of any valid audio file
Changes Take Effect: Immediately for all new calls
Related Feature: “Playing Multiple Media Files” on page 214

Specifies the name of the file that is played for the music on hold treatment when this DN places the call on hold or when the call is waiting on an ACD queue..

Note: You can define this option at both the Application and the DN level. The DN-level option takes precedence.

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 141

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:

- Agent Login—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: For the next call
Related Feature: "Dial Plan" on page 141

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, described below.
`digits;`	Specifies the digits that 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 semi-colon (`;`). 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 142

- "Digit Translation" on page 143
- "Dial Plan Parameters" on page 144

display-name

Default Value: No default value
Valid Values: A string
Changes Take Effect: Immediately

Specifies the string that will be sent as a display name in the `From` header of the `INVITE` request. This option applies only if the `use-display-name` configuration option is set to `true`.

dual-dialog-enabled

Default Value: `true`
Valid Values:

`true`	Set the option to `true` for endpoints that accept more than one SIP dialog and provide remote CTI control by the `NOTIFY` message.
`false`	Set the option to `false` for endpoints that can only accept one active SIP dialog, or cannot provide remote CTI control by the `NOTIFY` message to answer, hold, or retrieve call operations.

Changes Take Effect: Immediately

Enables the SIP dialog functionality for making consultation calls, according to the endpoint type.

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 216 for more information.

Note: You must enable the `subscribe-presence` option before enabling this 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 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 should be set to `true` for all IMS endpoints, as well as for the `Trunk` DN used for routing to IMS.

enforce-trusted

Default Value: Default is applied at the Application-level
Valid Values: `true, false`
Changes Take Effect: Next call
Related Feature: “Network Asserted Identity” on page 192

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.

Note: This option can also be configured on 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.

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: For the next call
Related Features: “Dial Plan” on page 141
Related Option: `privilege-level` (page 352)

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 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: Immediately
Related Feature: “Working with Multiple Devices” on page 247

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).

Note: For gateway `Trunk` DN selection, you must set `find-trunk-by-location` to `true` in order to include the `geo-location` option in the selection procedure.

Table 33 describes the possible DNs that can use this option:

Table 33: DN Configuration Objects

Device Type	Genesys DN Type
Agent SIP endpoint	`Extension`
Media Gateway	`Trunk`
Music-on-hold or Music-in-queue server (such as Stream Manager)	`VoIP Service` with `service-type=music`
Voice Treatment Server (such as Stream Manager)	`VoIP Service` with `service-type=treatment`
Voice Recorder (such as Stream Manager)	`VoIP Service` with `service-type=recorder`
Multipoint Conference Unit (such as Stream Manager)	`VoIP Service` with `service-type=mcu`

Note: Virtual resources such as Routing Points or ACD Queues must not use this option.

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: For the next call
Related Feature: "Dial Plan" on page 141

SIP Server will consider the dial-plan rules that are in the dial-plan 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 remote party.
-	Disables all INFO messages from being sent to the remote party.
<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 remote party.
*,<list>	Allows all INFO messages to be sent to the remote party, 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 through to the remote device, to disable all INFO messages from being sent to the remote device, 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 remote party, except for the following:

- application/vnd.radisys.msml+xml
- application/x-www-form-urlencoded
- application/x-detect (not sent if predictive call is in progress, or if no AudioCodes CPD result header is found)

Note: You can define this option at both the Application and the DN level. The DN-level option takes precedence.

line-type

Default Value: 0
Valid Values: 0, 1
Changes Take Effect: Immediately

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, TSingleStepTransferCall, 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-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 when the option `refer-enabled` is set to `false` for that DN.

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 200

Determines the action taken on an agent if the agent receives a SIP Server business call but fails to answer the call within the time defined in option `agent-no-answer-timeout`. This option is defined on any `Agent Login` object. When set, the value overrides the global `agent-no-answer-action` SIP Server configuration option for that agent.

Notes: If a call is abandoned before either the `agent-no-answer-action`, `no-answer-timeout`, or `supervised-route-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 `agent-no-answer-timeout` expires.
`recall`	SIP Server returns the call to the last distribution device (the device reported in the `ThisQueue` attribute of the call) when `agent-no-answer-timeout` expires.
`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 as defined at the Application level in the main SIP Server section.
Any valid overflow destination	SIP Server returns the call to the specified destination when the value set for the `agent-no-answer-timeout` option expires.

Changes Take Effect: Immediately
Related Feature: “No-Answer Supervision” on page 200

Defines a sequence of overflow destinations (separated by 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`
- `DN` of type `Extension`
- `DN` of type `ACD Position`

When set, this option overrides any of the following global SIP Server configuration options for the object where it has been set (depending on configuration object type):

- `agent-no-answer-overflow` if defined for an `Agent Login` object.
- `extn-no-answer-overflow` if defined for a `DN` of type `Extension` object.
- `posn-no-answer-overflow` if defined for a `DN` of type `ACD Position` object.

no-answer-timeout

Default Value: Same as value in corresponding global option
Valid Value: Any integer from `0`–`600`
Changes Take Effect: Immediately
Related Feature: “No-Answer Supervision” on page 200

Defines the time (in seconds) that SIP Server waits for a call that is ringing on the 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:

- `DN` of type `Extension`
- `DN` of type `ACD Position`
- `Agent Login` object

When set to `0`, the NoAnswer Supervision feature for this device is disabled. When set, this option overrides any of the following global SIP Server configuration options for the object where it has been set (depending on configuration object type):

- `agent-no-answer-timeout` if defined for an `Agent Login` object.
- `extn-no-answer-timeout` if defined for a `DN` of type `Extension` object.
- `posn-no-answer-timeout` if defined for a `DN` of type `ACD Position` object.

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 220

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 login, in order 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 168

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: This option is only supported on the following DN types:

- Voice over IP Service
- Trunk

The oos-check option is not applicable on internal DNs (DNs of type Extension or ACD Position).

oos-force

Default Value: 0
Valid Values: 0–30
Changes Take Effect: Immediately
Related Feature: "Active Out-of-Service Detection" on page 168

Specifies 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.

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 168

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: Immediately

When 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 123.

out-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
Related Feature: “Class of Service” on page 126

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 “Dialing Rule Format” on page 128. See “Examples” on page 132 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:

```
out-rule-01 = in-pattern=0111#CABBB*ccD;out-pattern=ABD
```

override-domain

Default Value: `NULL`
Valid Values: Any computer name
Changes Take Effect: Immediately

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-from

Default Value: `NULL`
Valid Values: Any computer name string
Changes Take Effect: Immediately

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-call-type

Default Value: `0`

Valid Values:

0	CallTypeUnknown
1	CallTypeInternal
2	CallTypeInbound
3	CallTypeOutbound

Changes Take Effect: Immediately

Determines the value SIP Server will use as the CallType attribute for internal calls made directly to a DN of type Routing Point. When set to 0, SIP Server specifies the CallType attribute as Internal.

p-asserted-identity

Default Value: No default value (empty string)

Valid Values: Any string in accordance with RFC 3325

Changes Take Effect: Next call

Related Feature: "Network Asserted Identity" on page 192

Related Option: "privacy" on page 351

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 192. If you configure this option on an Extension DN, the value provides the content of the p-asserted-identity header for this particular D.

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.

password

Default Value: No default value

Valid Values: Any alphanumerical string
Changes Take Effect: Immediately

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.

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.

prefix

Default Value: No default value
Valid Values: Any alphanumerical string
Changes Take Effect: Immediately

In the MCU configuration: Specifies the starting digits of the number that are used when sending calls to MCU. The full number is built as: `<prefix><connid>@<ipaddr>:port`. Typically, MCU servers require a prefix consisting of digits in order to identify a type of conference (for example, voice only, voice and video, and so on). Set the value as `conf=` if Stream Manager is used as the MCU.

In the gateway configuration: Contains the initial digits of the number that must match a particular gateway for that gateway to be selected. If multiple gateways match a number, the gateway with the longest prefix is selected.

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: Immediately for all future calls
Related Feature: “Preview Interactions” on page 226

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 172.

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 backwards compatibility, SIP Server also accepts the following valid values for this option:

- `True`—The protocol is enabled (equivalent to tlib).
- `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: Defined in "RFC 3323: A Privacy Mechanism for the Session Initiation Protocol (SIP)"
Changes Take Effect: Next call
Related Feature: "Network Asserted Identity" on page 192

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 216

Specifies the value of the Private Line User Part ID for the Busy Lamp Field feature when integrating with the BroadWorks softswitch. 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 BroadWorks 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: For the next call
Related Option: `fwd-privilege-level` (page 341)
Related Features: "Dial Plan" on page 141

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 INVTE, `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 alphanumerical 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.

Note: The `public-contact` option is only applicable to Alcatel 5020.

record

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: When the next call is established on the DN

Related Feature: "Call Recording" on page 97

When set to `true`, call recording begins automatically when the call is established on the DN. Call recording stops when the DN leaves the call.

recovery-timeout

Default Value: `0`
Valid Values: `0–86400 seconds`
Changes Take Effect: Immediately
Related Feature: "Passive Out-of-Service Detection" on page 167
Related Options: sip-oos-enabled on page 361

Controls whether a device is taken out of service when an error is encountered, and for how long it is out of service. When 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` (ACDF Position), the zero value does not have any affect.

SIP Server supports this option on the following types of DN:

- `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 168.

rfc-2976-dtmf

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

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.

Note: 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. When 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.

When 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 123.

reinvite-requires-hold

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

When 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.

Notes: This option should 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.

For all other cases, consult Genesys Technical Support for recommendations about enabling this option.

reject-call-incall

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When set to `true`, a call attempt to a DN that is already on a call will be rejected, and the `Invalid Destination State (93)` error message will be generated.

reject-call-notready

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When set to `true`, a call attempt to a DN at which an agent is in a `Logout`, `NotReady`, or `AfterCallWork` state will be rejected, and the `Invalid Destination State (93)` error message will be generated.

Note: The `reject-call-incall` and `reject-call-notready` options are applicable to the following T-Library requests:

- `TMakeCall`
- `TInitiateTransfer`
- `TInitiateConference`
- `TSingleStepTransfer`
- `TSingleStepConference`

replace-prefix

Default Value: `0`
Valid Values: Any non-negative integer
Changes Take Effect**:** Immediately

Contains the digits that are inserted in the DN instead of the prefix for the gateway. If this option value is absent, the number is not modified.

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 Stream Manager, 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.

reuse-sdp-on-reinvite

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

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.

Notes: The value must be set to `true` when using EyeBeam version 1.1.

A limitation to this functionality: if reuse-sdp-on-reinvite is configured on the originating DN (or trunk) for a call, SIP Server cannot send INVITE requests with no SDP, and therefore cannot generate an offer by sending a re-INVITE message to the origination party (or to the MCU).

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.

When set to `true`, SIP Server connects the caller with an audio ringtone from Stream Manager 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` (see page 354)
  - `make-call-rfc3725-flow` set to `1` (see page 344)
- The calling DN initiates a consultation call must be configured with the following option:
  - `dual-dialog-enabled` set to `false` (see page 339)

When the `ring-tone-on-make-call` option is set to `false`, there is no ring tone.

Note: SIP Server does not support internal ring tones in conference scenarios where the `sip-ring-tone-mode` option is set to `1`. In this case, SIP Server provides a ring tone only if the endpoint returns an SDP in the provisional message.

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 34). See Chapter 4, "SIP Devices Support," on page 59 for more information on using this option. For more information about configuring a Class of Service DN, see "Class of Service" on page 126. For more information about configuring a dial-plan DN, see "Dial Plan" on page 141.

Table 34: 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`
Application service	Voice over IP Service	`application`

Table 34: Service-Type Settings for SIP Devices (Continued)

SIP Device Type or Service	Genesys DN Type	Service-Type Setting
Class of service	Voice over IP Service	`cos`
Dial plan	Voice over IP Service	`dial-plan`

sip-add-primary-call-id

Default Value: `false`
Valid Values: `false, true`
Changes Take Effect: Immediately

When set to `true`, any `TMakeCall` request that is initiated for a DN already on call will start with an `INVITE` message that has a proprietary `P-gcti-primary-call-id` header. The value of this header is the `CallID` SIP attribute from the primary SIP call dialog for that DN.

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 171

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-cti-control

Default Value: No default value
Valid Values: `beep`, `dtmf`, `talk`, `hold`

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.
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: Immediately
Related Feature (for values `beep` and `dtmf`): “Remote Media on Genesys SIP Endpoint SDK 8.0” on page 229

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 , see the *SIP Endpoint SDK 8.0 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-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 160
Related Options: "charge-type" on page 333 and "sip-server-inter-trunk" on page 363

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-gdns

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: After SIP Server restart

Specifies the DNS resolution mode. If you set this option to `true`, SIP Server uses its internal DNS client to connect to the DNS server available on the network, in order 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-moh

Default Value: No default value
Valid Value: `true, false, na`
Changes Take Effect: At next call (if value is empty, changes do not take effect until after application restart)
Related Feature: "Playing Multiple Media Files" on page 214

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: You can define this option at both the Application and the DN level. The DN-level option takes precedence.

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: Immediately

Related Option: audio-codecs on page 330.

For a description, see the application-level option sip-enable-sdp-codec-filter (page 314).

sip-hold-rfc3264

Default Value: false
Valid Values:

true	RFC3264-compliant implementation.
false	RFC2543-compliant implementation.

Changes Take Effect: Immediately

Specifies which implementation of hold media SDP is used by SIP Server for hold operations.

Note: When this option is set at the DN level in the Annex tab > TServer section, it overrides the Application-level value.

sip-oos-enabled

Default Value: true
Valid Value: true/false
Related Feature: “Passive Out-of-Service Detection” on page 167
Related Option: recovery-timeout on page 353

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-preserve-contact

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

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.

Note: 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 setting at the Application level.

sip-replaces-mode

Default Value: 0
Valid Values: `0, 1, 2`
Changes Take Effect: Immediately
Related Feature: "TCompleteTransfer using REFER or REFER with Replaces" on page 125

This option specifies the SIP method that will be used by SIP Server to complete a two-step transfer.

- With a value of `0`, SIP Server will use the `REFER` method if the `transfer-complete-by-refer` option is set to `true`.
- With a value of `1`, SIP Server will use 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 will use the REFER method with Replaces to process TCompleteTransfer. The Allow and Supported headers will not be analyzed.

Notes: The sip-replaces-mode option does *not* apply to a DN of type Trunk if the sip-server-inter-trunk option in the DN configuration is set to true.

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-ring-tone-mode

Default Value: 0
Valid Values: 0,1
Changes Take Effect: Immediately

With the option set to 0, SIP Server connects Stream Manager to a call to play an audio ring tone. With the option set to 1, SIP Server waits for a response from the called device, and connects Stream Manager to a call to play an audio ring tone, only when the returned response cannot be used as the offer to a calling device.

Notes: SIP Server does not support internal ring tones in conference scenarios where the sip-ring-tone-mode option is set to 1. In this case, SIP Server provides a ring tone 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 that 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-server-inter-trunk

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately
Related Features: “Trunk Optimization for Multi-Site Transfers” on page 238 and “Early Media for Inbound Calls” on page 160

Related Options: "charge-type" on page 333 and "sip-early-dialog-mode" on page 359

When 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 171

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`, use of SIP Server for IM 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: "Trunk Capacity Support" on page 235

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 88.

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: "Trunk Capacity Support" on page 235

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 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 88.

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 the MSML protocol. In this case, the GVP RM is represented in the Configuration Layer as a `Trunk Group` DN with the required option `subscription-id` properly configured. You must set the value of this option to the name of the tenant to which the SIP Server switch belongs. In this configuration, SIP Server works as a subscribing client—it activates a subscription to the GVP Resource Manager, and then

receives notifications on the number of ports available for the outbound campaign.

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.
- You can configure multiple Resource Manager `Trunk Group` DNs on the same tenant as the SIP Server switch. However, Genesys recommends that you plan your 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 216

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 216

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 216

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 216

Enables presence subscription and mapping of a presence state to an agent state.

- When 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.
- When set to the name of a Trunk DN that contains the subscription parameters is specified, the enable-agentlogin-presence option (see page 339) must also be configured for the same Trunk DN.

transfer-complete-by-refer

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately
Related Feature: "TCompleteTransfer using REFER or REFER with Replaces" on page 125

When 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 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-inter-trunk option is set to true.

use-contact-as-dn

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

Specifies whether SIP Server will use the username of the Contact header as attribute ThisDN.

use-display-name

Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

This option activates the `display-name` configuration option.

userdata-map-filter

Default Value: No default value
Valid Values:

*: All data is mapped.

A list of prefixes A comma-separated list of prefixes used to identify the `UserData` key-value pair to be mapped.

Changes Take Effect: Immediately
Related Feature: "Mapping SIP Headers and SDP Messages" on page 177

Specifies the names of the key-value pairs to be mapped. If this option is not specified, no data will be mapped.

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.

- When 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 that 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, then SIP Server generates a `403 Forbidden` response.

- When 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.

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 on the Options tab for the SIP Server Application object in Configuration Manager.

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 35: 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
clid-withheld-name	Application level > TServer section
correct-rqid	Application level > TServer section
default-dn-type	Application level > TServer section
dn-del-mode	Application level > TServer section
emulate-login	Application level > TServer section

Table 35: Reserved Configuration Options (Continued)

Option Name	Option Section
enable-ims	Application level > `TServer` section
expire-call-tout	Application level > `TServer` section
init-dnis-by-ruri	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
reg-delay	Application level >`link-control` section
reg-interval	Application level > `TServer` section
reg-silent	Application level >`link-control` section
rq-expire-tmout	Application level > `TServer` section
sync-emu-agent	Application level > `TServer` section
unknown-xfer-merge-udata	Application level > `TServer` section
use-display-name	Application level > `TServer` section
wrap-up-threshold	Application level > `TServer` section

Changes from Release 7.6 to Release 8.0

Table 36 lists the configuration options that:

- Are new or changed in the 8.0 release of SIP Server

- Have been added or changed since the most recent 7.6 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 36: Option Changes from Release 7.6 to Release 8.0

Option Name	Option Values	Type of Change	Details
Application Level > Annex Tab > TServer Section			
audio-codecs	String	Changed in 8.0	See the option description on page 281
auto-logout-ready	true, false	New in 8.0	See the option description on page 282.
auto-logout-timeout	Integer	New in 8.0	See the option description on page 282.
busy-tone-duration	Integer	New in 8.0	See the option description on page 284.
cos	String	New in 8.0	See the option description on page 285.
cpd-info-timeout	Integer	New in 8.0.	See the option description on page 285.
dial-plan	String	New in 8.0	See the option description on page 287.
delay-to-start-refresh-on-switchover	Integer	Removed in 8.0	Starting in 7.6, this option is obsolete.
delay-between-refresh-on-switchover	Integer	Removed in 8.0	Starting in 7.6, this option is obsolete.
enable-agentlogin-subscribe	true, false	New in 8.0	See the option description on page 288.
find-trunk-by-location	true, false	New in 8.0	See the option description on page 294.
info-pass-through	String	New in 8.0	See the option description on page 298.
map-sip-errors	true, false	New in 8.0	See the option description on page 301.
music-in-queue-file	String	Changed in 8.0.	See the option description on page 303.
no-response-dn	String	New in 8.0.	See the option description on page 305.
partition-id	String	New in 8.0.	See the option description on page 307.
predictive-call-router-timeout	Integer	Changed in 8.0.	See the option description on page 308.
record-consult-calls	true, false	New in 8.0	See the option description on page 309.
sip-ip-tos	Integer	New in 8.0	See the option description on page 315.
set-notready-on-busy	true, false	New in 7.6.	See the option description on page 312.

Table 36: Option Changes from Release 7.6 to Release 8.0 (Continued)

Option Name	Option Values	Type of Change	Details
sip-preserve-contact	true, false	New in 8.0	See the option description in page 318.
stranded-calls-overflow	String	New in 8.0.	See the option description in page 323.
stranded-on-arrival-calls-overflow	String	New in 8.0.	See the option description in page 323.
stranded-call-redirection-limit	Integer	New in 8.0.	See the option description in page 323.
Application Level > Annex Tab > extrouter Section			
default-network-call-id-matching	sip	New in 8.0	See the option description on page 328.
Application Level > Annex Tab > SipErrorMapr Section			
sip-<SIP_error_code>	Integer	New in 8.0	See the option description on page 329.
DN Level > Options Tab > TServer Section			
authenticate-requests	String	Changed in 8.0.	See the option description on page 330.
auto-logout-ready	String	New in 8.0.	See the option description on page 331.
auto-logout-timeout	Integer	New in 8.0.	See the option description on page 331.
beep-duration	Integer	New in 8.0.	See the option description on page 332
charge-type	0, 1	New in 8.0	See the option description on page 333.
cos	String	Added in 8.0.	See the option description on page 335.
cpd-capability	String	New in 8.0.	See the option description on page 336.
default-music	String	New in 8.0	See the option description on page 337.
dial-plan	String	New in 8.0.	See the option description on page 337.
dial-plan-rule-<n>	String	New in 8.0.	See the option description on page 338.
display-name	String	New in 8.0	See the option description on page 339.
include-dial-plan-<n>	String	New in 8.0.	See the option description on page 342.
info-pass-through	String	New in 8.0.	See the option description on page 342.
geo-location	String	New in 8.0	See the option description on page 341.
line-type	0, 1	New in 8.0	See the option description on page 343.

Table 36: Option Changes from Release 7.6 to Release 8.0 (Continued)

Option Name	Option Values	Type of Change	Details
music-in-queue-file	String	Removed in 8.0.	Use `default-music` (page 287) instead.
oos-options-max-forwards	Integer	New in 8.0.	See the option description on page 347.
oosp-transfer-enabled	Integer	Changed in 8.0.	See the option description on page 347.
privilege-level	String	New in 8.0.	See the option description on page 352.
private-line	String	New in 8.0	See the option description on page 351.
sip-early-dialog-mode	0, 1	New in 8.0	See the option description on page 359.
sip-preserve-contact	true, false	New in 8.0	See the option description on page 362.
sip-refer-to-sst-enabled	true, false	Removed in 8.0	Use `refer-enabled` only. Single step transfers are never rejected See page 354.
sip-replaces-mode	0, 1, 2	New in 8.0	See the option description on page 362.
stranded-calls-overflow	String	New in 8.0.	See the option description in page 364.
stranded-on-arrival-calls-overflow	String	New in 8.0.	See the option description in page 364.
subscription-id	String	New in 8.0.	See the option description in page 365.
transfer-complete-by-refer	true, false	New in 8.0	See the option description on page 367.
use-display-name	true, false	New in 8.0	See the option description on page 368.

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 377, 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 391, describes the variations available for T-Server implementations across geographical locations.
- Chapter 10, "Common Configuration Options," on page 449, describes log configuration options common to all Genesys server applications.
- Chapter 11, "T-Server Common Configuration Options," on page 469, describes configuration options common to all T-Server types including options for multi-site configuration.

New for All T-Servers in 8.0

Before looking at T-Server's place in Genesys solutions and in the architecture of the Genesys Framework, note the following general changes that have been implemented in the 8.0 release of T-Server:

- **Enhanced Event Propagation support for switch partitioning.** T-Server now supports the Event Propagation feature in deployments that use switch partitioning or intelligent trunks. See "Switch Partitioning" on page 430.

- **Enhanced ISCC Transaction Monitoring support.** T-Server now supports new key-value pairs in `AttributeExtensions` with ISCC transaction data requested using `TGetAccessNumber` in the following requests: `TMakeCall`, `TRouteCall`, `TSingleStepTransfer`, `TInitiateTransfer`, `TInitiateConference`, and `TMuteTransfer`. The ISCC Transaction Monitoring allows T-Server clients to monitor ISCC transactions of the call data transfer between T-Servers in a multi-site environment. See "ISCC Transaction Monitoring Feature" on page 434 and the *Genesys 7 Events and Models Reference Manual* for details about key-value pairs in `AttributeExtensions`.
- **Enhanced Agent Reservation support.** T-Server now supports Agent Reservation failure optimization, to ensure that only agent reservation requests of the highest priority are collected. This functionality can now be controlled with the `collect-lower-priority-requests` configuration option. See "Agent Reservation" on page 388 for details.
- **Link bandwidth reporting support.** T-Server now supports notification of link bandwidth utilization. The following two new log events have been introduced:
  - `20009|STANDARD|MSG_TS_COMMON_LINK_ALARM_HIGH`
  - `20010|STANDARD|MSG_TS_COMMON_LINK_ALARM_LOW`

  Refer to *Framework 8.0 Combined Log Events Help* for information about the log events.
- **Notification of failed routing attempts and failed ISCC transactions.** T-Server now supports notification of failed routing attempts and failed ISCC transactions. The following new log events have been introduced:
  - `20011|STANDARD|MSG_TS_COMMON_ALARM_ROUTE_FAILURE_HIGH_WATER_MARK`
  - `20012|STANDARD|MSG_TS_COMMON_ALARM_ROUTE_FAILURE_LOW_WATER_MARK`
  - `21019|STANDARD|ISCC_LOGMSG_TRANSACTION_FAILED`

  Refer to *Framework 8.0 Combined Log Events Help* for information about the log events.
- **Real-time SDN licenses query support.** T-Server can now report how many SDN licenses are currently available and in use, using the following key-value pairs in `AttributeExtensions` in `EventServerInfo` messages: `sdn-licenses-in-use` and `sdn-licenses-available`. See Part Two of this document for details on the use of `AttributeExtensions` in a particular T-Server.

Notes: Configuration option changes common to all T-Servers are described in "Changes from Release 7.6 to 8.0" on page 494.

For information about the new features that are available in your T-Server in the initial 8.0 release, see Part One of this document.

Chapter

8 T-Server Fundamentals

This chapter provides general information that applies to all T-Servers (and Network T-Servers) and their deployment. This chapter is divided into the following sections:

- Learning About T-Server, page 377
- Advanced Disconnect Detection Protocol, page 383
- Redundant T-Servers, page 384
- Multi-Site Support, page 388
- Agent Reservation, page 388
- Client Connections, page 389SIP Server

Learning About T-Server

The *Framework 8.0 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 other two components are Load Distribution Server (LDS) and HA Proxy). 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 41 illustrates the position Framework holds in a Genesys solution.

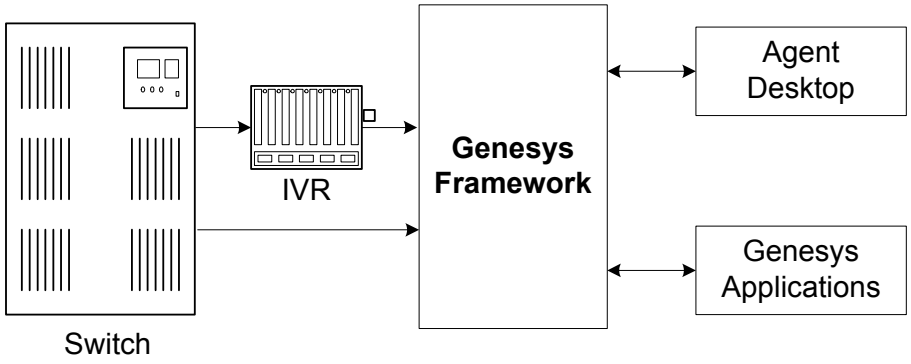

Figure 41: Framework in a Genesys Solution

Moving a bit deeper, Figure 42 presents the various layers of the Framework architecture.

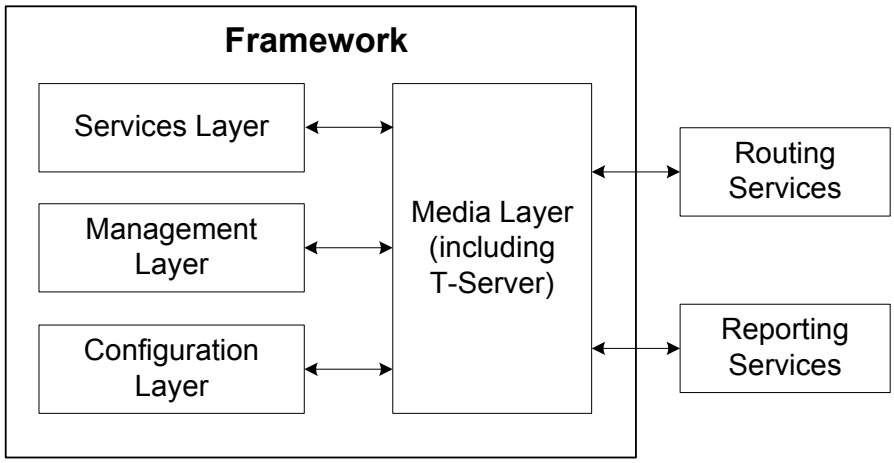

Figure 42: The Media Layer in the Framework Architecture

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 43 presents the generalized architecture of the Media Layer.

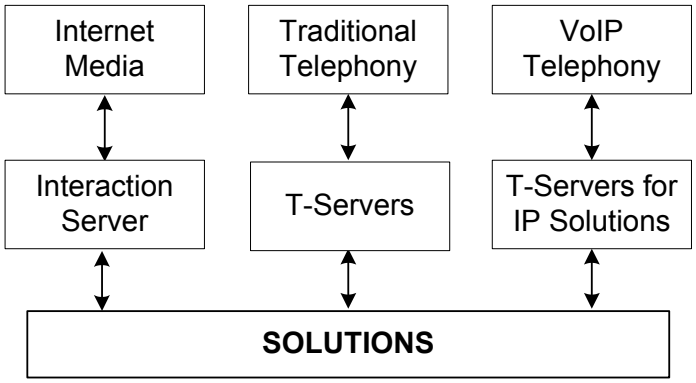

Figure 43: Media Layer Architecture

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, Call 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 7 Events and Models Reference Manual* for complete information on all T-Server events and call models and to the `TServer.Requests` portion of the *Voice Platform SDK 8.0 .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 43 on page 379 (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 44, 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.

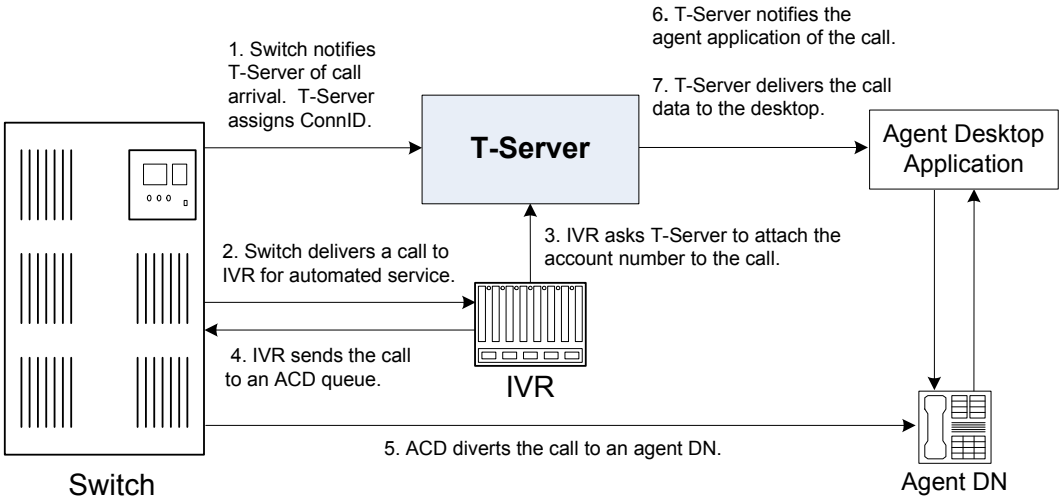

Figure 44: Functional T-Server Steps

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. (See Table 37.)

Notes: IVR Server and some Network T-Servers can be configured for load sharing or `warm` or `hot standby`; however, they do not support any combination of these redundancy types.

Support for Hot Standby Redundancy in Various T-Servers

Use Table 37 to determine whether your T-Server supports the `hot standby` redundancy type. The table also indicates whether HA Proxy components are required for this support, and, if so, how many are required per pair of redundant T-Servers (or per link if so noted).

Table 37 only summarizes `hot standby` redundancy support in various T-Servers. For detailed, up-to-date information on the subject, see the *Genesys Supported Media Interfaces* white paper located on the Technical Support website at http://genesyslab.com/support/dl/retrieve/default.asp?item=A9CB309AF4DEB8127C5640A3C32445A7&view=item.

Table 37: T-Server Support of the Hot Standby Redundancy Type

T-Server Type	Hot Standby Supported	HA Proxy Required	Number of HA Proxy Components
Alcatel A4200/OXO	Yes	No	—
Alcatel A4400/OXE	Yes	No	—
Aspect ACD	Yes	No	—

Table 37: T-Server Support of the Hot Standby Redundancy Type (Continued)

T-Server Type	Hot Standby Supported	HA Proxy Required	Number of HA Proxy Components
Avaya Communication Manager	Yes	No[a]	—
Avaya INDeX	Yes	No	—
Avaya TSAPI	Yes	No	—
Cisco Unified Communications Manager	Yes	No	—
DataVoice Dharma	Yes	No	—
Digitro AXS/20	Yes	No	—
EADS Intecom M6880	Yes	No	—
EADS Telecom M6500	Yes	No	—
eOn eQueue	Yes	No	—
Ericsson MD110	Yes	No	—
Fujitsu F9600	Yes	No	—
Huawei C&C08	Yes	No	—
Huawei NGN	Yes	No	—
Mitel SX-2000/MN-3300	Yes	No	—
NEC NEAX/APEX	Yes	No	—
Nortel Communication Server 2000/2100	Yes	Yes[b], No[c]	1 per link
Nortel Communication Server 1000 with SCCS/MLS	Yes	No	—
Philips Sopho iS3000	Yes	No[d]	1
Radvision iContact	No	—	—
Rockwell Spectrum	Yes	No	—
Samsung IP-PCX IAP	Yes	No	—
Siemens Hicom 300/HiPath 4000 CSTA I	Yes	No	—
Siemens HiPath 3000	Yes	No	—
Siemens HiPath 4000 CSTA III	Yes	No	—

Table 37: T-Server Support of the Hot Standby Redundancy Type (Continued)

T-Server Type	Hot Standby Supported	HA Proxy Required	Number of HA Proxy Components
Siemens HiPath DX	Yes	No	—
SIP Server	Yes	No	—
Tadiran Coral	Yes	No	—
Teltronics 20-20	Yes	Yes	1
Tenovis Integral 33/55	Yes	No	—
Network T-Servers[e]			
AT&T	No	—	—
Concert	No	—	—
CRSP	No	—	—
DTAG	No	—	—
GenSpec	No	—	—
ISCP	No	—	—
IVR Server, using network configuration	Yes	—	—
KPN	No	—	—
MCI	No	—	—
NGSN	No	—	—
Network SIP Server	No	—	—
Sprint	No	—	—
SR3511	No	—	—
Stentor	No	—	—

a. With release 7.1, T-Server for Avaya Communication Manager no longer uses HA Proxy for its support of `hot standby`. Earlier releases of this T-Server require two HA Proxies (for which there is a Configuration Wizard) to support `hot standby`.

b. For T-Server for Nortel Communication Server 2000/2100 in high-availability (`hot standby`) configuration, Genesys recommends that you use link version SCAI14 or above with call-progress and noncontroller-released messages enabled. See the switch-specific information in your *T-Server Deployment Guide* for additional information on HA configurations.

c. Starting with release 7.5, T-Server for Nortel Communication Server 2000/2100 supports HA without HA Proxy when operating in Dual CTI Links mode. See the switch-specific information in your *T-Server Deployment Guide* for additional information on HA configurations.

d. Starting with release 6.5.3, T-Server for Philips Sopho iS3000 supports HA both with and without HA Proxy.

e. Although they do not support high availability per se, Network T-Servers do support a load-sharing schema.

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 391.

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 393), 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 400 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 *Voice Platform SDK 8.0 .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 478 in the "T-Server Common Configuration Options" chapter for more details.

Starting with version 8.0, 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 478).

Client Connections

The number of connections T-Server can accept from its clients depend on the operating system that T-Server runs. Table 38 illustrates the number of client connections that T-Server support.

Table 38: Number of T-Server's Client Connections

Operating System	Number of Connections
AIX 32-bit mode (versions 5.1, 5.2, 5.3)	32767
AIX 64-bit mode (versions 5.1, 5.2, 5.3, 6.1)	32767
HP-UX 32-bit mode (versions 11.11, 11i v2)	2048
HP-UX 64-bit mode (versions 11.11, 11i v2, 11i v3)	2048
Linux 32-bit mode (versions RHEL 3.0, RHEL 4.0, RHEL 5.0)	32768
Solaris 32-bit mode (versions 8, 9)	4096
Solaris 64-bit mode (versions 8, 9, 10)	65536
Tru64 UNIX (versions 4.0F, 5.1, 5.1B)	4096
Windows Server 2003, 2008	4096

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:

- Multi-Site Fundamentals, page 391
- ISCC Call Data Transfer Service, page 393
- ISCC/Call Overflow Feature, page 413
- Number Translation Feature, page 417
- Network Attended Transfer/Conference Feature, page 425
- Event Propagation Feature, page 427
- ISCC Transaction Monitoring Feature, page 434
- Configuring Multi-Site Support, page 434

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 469.

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 39 on page 409 and Table 40 on page 414.

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 400 and "Transfer Connect Service Feature" on page 412.
  - 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 413).
  - Number Translation feature (see page 417).
  - Network Attended Transfer/Conference (NAT/C) feature (see page 425).

  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 428)
  - Party Events propagation (see page 429)

  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 Deployment Guide* for specific configuration details.

Figure 45 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.*

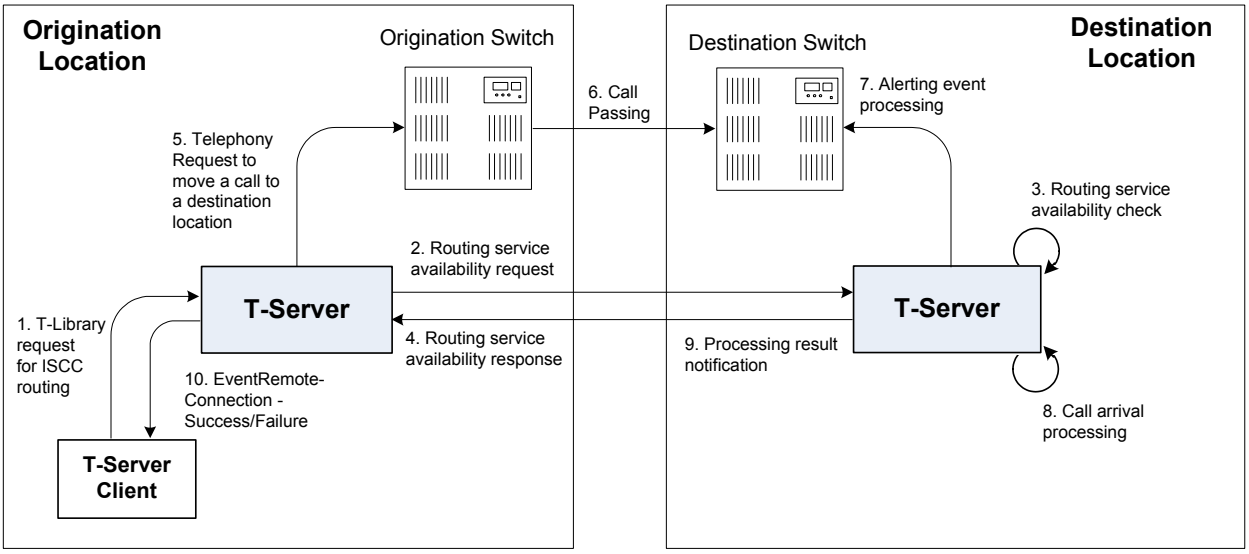

Figure 45: Steps in the ISCC Process

ISCC Call Flows

The following section identifies the steps (shown in Figure 45) 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 *Voice Platform SDK 8.0 .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 436.

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 469 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.

Please 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 401).

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 401. Use Table 39 on page 409 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 469 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 401
- `direct-notoken`, page 403
- `dnis-pool`, page 404
- `pullback`, page 405
- `reroute`, page 406
- `route` (aliased as `route-notoken`), the default transaction type, page 407

The following transaction types are unevenly supported for both the origination and destination T-Server roles:

- `direct-callid` (aliased as `direct`), page 402
- `direct-digits` (reserved for Genesys Engineering)
- `direct-network-callid`, page 402
- `direct-uui`, page 403
- `route-uui`, page 408

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 445 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 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 46.

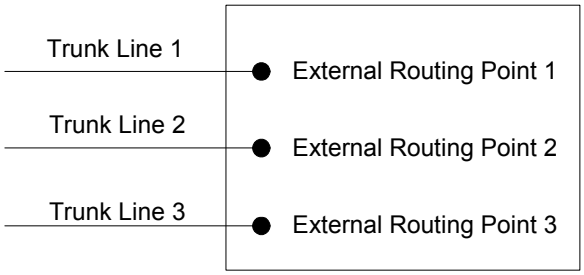

Figure 46: Point-to-Point Trunk Configuration

Note: Dedicated DNs of the `External Routing Point` type must be configured in a switch. See “Configuring Multi-Site Support” on page 434.

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 47.

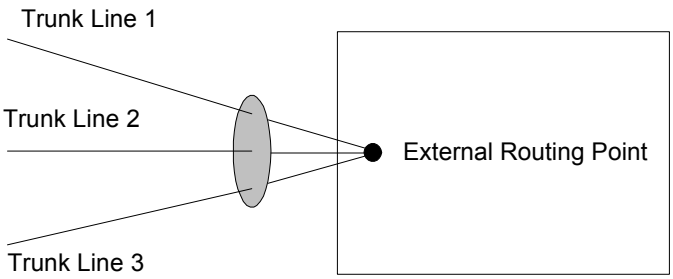

Figure 47: Multiple-to-Point Trunk Configuration

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 407) and the UUI matching feature of the `direct-uui` transaction type (page 403). 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 39 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 39: 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 A4200/OXO	Yes			Yes		Yes	Yes				
Alcatel A4400/OXE	Yes			Yes[a,b,c]	Yes[d]	Yes	Yes[a]		Yes[e]		
Aspect ACD	Yes	Yes		Yes		Yes[f]	Yes[f]				
Avaya Communication Manager	Yes				Yes	Yes	Yes				
Avaya INDeX	Yes					Yes	Yes[b]				
Avaya TSAPI	Yes				Yes	Yes	Yes				
Cisco Unified Communications Manager	Yes			Yes		Yes	Yes				
DataVoice Dharma	Yes			Yes		Yes	Yes				
Digitro AXS/20	Yes			Yes		Yes					
EADS Intecom M6880	Yes			Yes		Yes	Yes				
EADS Telecom M6500	Yes			Yes		Yes	Yes				
eOn eQueue	Yes			Yes		Yes					
Ericsson MD110	Yes			Yes[a]		Yes	Yes[a]				
Fujitsu F9600	Yes					Yes					
Huawei C&C08	Yes			Yes							

Table 39: 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									
Huawei NGN	Yes					Yes	Yes				
Mitel SX-2000/MN3 300	Yes			Yes		Yes	Yes				
NEC NEAX/APEX	Yes			Yes		Yes	Yes				
Nortel Communication Server 2000/2100	Yes			Yes[f]		Yes[f]	Yes[f]				
Nortel Communication Server 1000 with SCCS/MLS	Yes			Yes		Yes	Yes		Yes		
Philips Sopho iS3000	Yes			Yes		Yes	Yes				
Radvision iContact	Yes		Yes								Yes
Rockwell Spectrum	Yes	Yes		Yes		Yes[f]	Yes[f]				
Samsung IP-PCX IAP	Yes			Yes		Yes					
Siemens Hicom 300/HiPath 4000 CSTA I	Yes			Yes	Yes[d]	Yes	Yes				
Siemens HiPath 3000	Yes			Yes		Yes					
Siemens HiPath 4000 CSTA III	Yes				Yes[d]	Yes	Yes				

Table 39: 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									
Siemens HiPath DX	Yes			Yes	Yes	Yes	Yes				
SIP Server	Yes		Yes		Yes[g]	Yes					Yes
Tadiran Coral	Yes			Yes		Yes	Yes				
Teltronics 20-20	Yes			Yes		Yes	Yes				
Tenovis Integral 33/55	Yes			Yes		Yes	Yes				
Network T-Servers											
AT&T											
Concert											
CRSP											Yes
DTAG			Yes								
GenSpec	Yes	Yes	Yes							Yes	
IVR Server, using network configuration	Yes	Yes	Yes							Yes	Yes
KPN			Yes								
ISCP											
MCI											
NGSN	Yes										Yes
Network SIP Server	Yes					Yes	Yes			Yes	
Sprint	Yes										
SR-3511											
Stentor											

a. Not supported in the case of function `TRequestRouteCall` 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. To use this transaction type, you must select the `Use Override` check box on the `Advanced` tab of the DN `Properties` dialog box.

g. SIP Server supports the `direct-uui` type.

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. Open the T-Server `Application's Properties` dialog box.
2. Click the `Options` tab.
3. Set the `tcs-use` configuration option to `always`.
4. 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.
5. When you are finished, click `Apply`.

6. Click OK to save your changes and exit the Properties dialog box.

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 p*assive external routing*, is specifically designed to handle calls delivered between sites by means other than ISCC. 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 networkwide 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 40 shows the T-Server types that provide the `NetworkCallID` of a call.

Table 40: T-Server Support of NetworkCallID for ISCC/COF Feature

T-Server Type	Supported NetworkCallID Attribute
Alcatel A4400/OXE	Yes
Aspect ACD	Yes
Avaya Communication Manager	Yes
Avaya TSAPI	Yes
Nortel Communication Server 2000/2100	Yes
Nortel Communication Server 1000 with SCCS/MLS	Yes
Rockwell Spectrum	Yes
SIP Server	Yes

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 417.

ISCC/COF Call Flow

Figure 48 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.

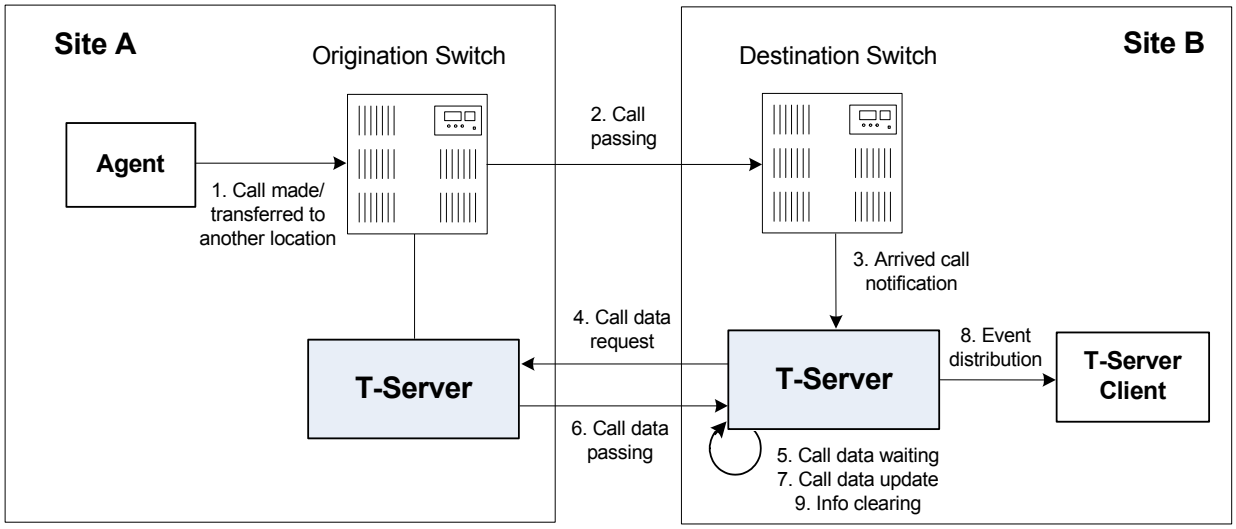

Figure 48: Steps in the ISCC/COF Process

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 was overflowed from any of them.

To determine which calls to check as possibly overflowed, T-Server relies on the `Switch` object configuration:

- If no COF DNs (that is, DNs of the `Access Resources` type with the `Resource Type` set to `cof-in` or `cof-not-in`) are configured for the destination switch, the ISCC/COF feature of the destination T-Server checks all arriving calls.

- If a number of COF DNs are configured for the destination switch, one of three scenarios occurs:
  - If the COF DNs with the `cof-in` setting for the `Resource Type` property are configured, the ISCC/COF checks for overflow only those calls that arrive to those `cof-in` DNs that are `Enabled`.
  - If no DNs with the `cof-in` setting for the `Resource Type` property are configured, but some DNs have the `cof-not-in` setting for the `Resource Type` property, the ISCC/COF checks for overflow only those calls that arrive to those `cof-not-in` DNs that are `Disabled`.
  - If no DNs with the `cof-in` setting for the `Resource Type` property are configured, some DNs have the `cof-not-in` setting for the `Resource Type` property, and some other DNs do not have any setting for the `Resource Type` property, the ISCC/COF checks for overflow only those calls that arrive to the DNs without any setting for the `Resource Type` property.
- In all other cases, no calls are 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.

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 418 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 423 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 425.

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 423) 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 469.
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 49 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).

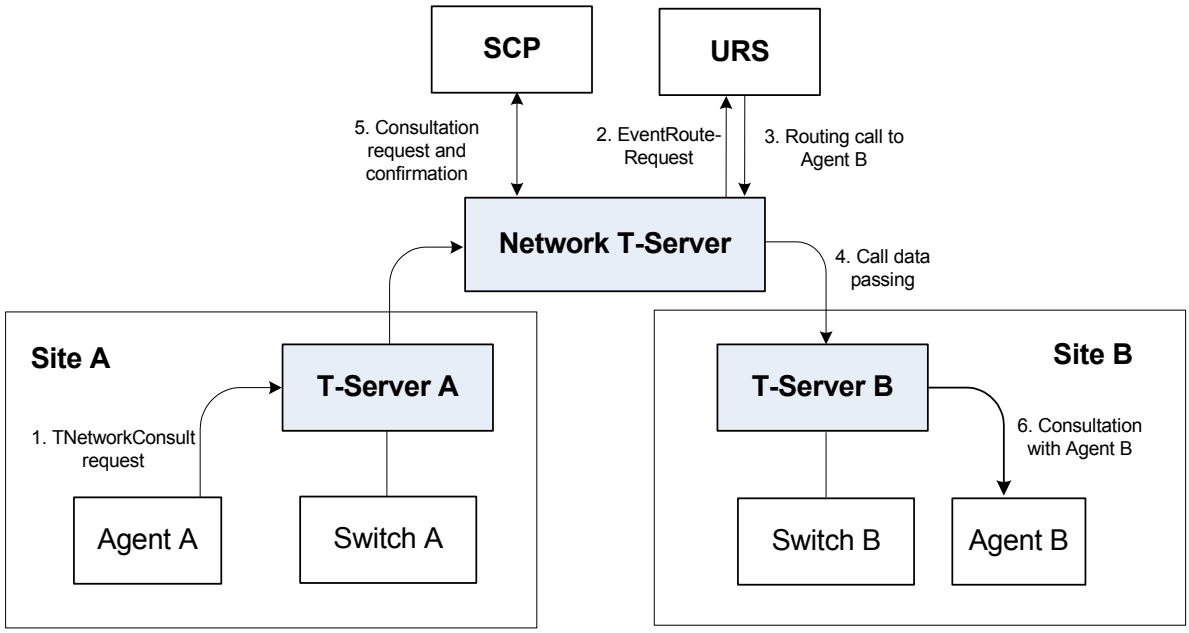

Figure 49: Steps in the NAT/C Process in URS-Controlled Mode

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 *Voice Platform SDK 8.0 .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 393 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 7 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 Configuration Manager 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 50.)

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.

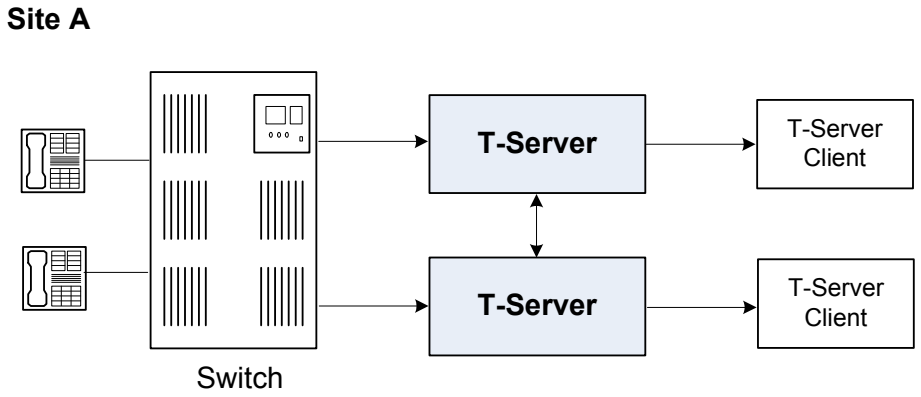

Figure 50: Switch Partitioning Architecture

Starting with version 8.0, in addition to `ConnIDs` and `UserData`, T-Server can now 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 474.

To control race conditions that may occur in the switch-partitioned environment, use the `epp-tout` configuration option (see page 488).

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 41 shows the T-Server types that support switch partitioning.

Table 41: 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

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. Open the T-Server `Application's Properties` dialog box.
2. Click the `Options` tab.
3. Open the `extrouter` section.
4. 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.
5. 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 469.
6. When you are finished, click `Apply`.
7. Click `OK` to save your changes and exit the `Properties` dialog box.

End of procedure

Next Steps

- For advanced feature configuration, do the following procedure:

  Procedure: Modifying Event Propagation: advanced configuration, on page 432

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 432

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 437 for instructions. If no `Access Codes` are configured, see page 438 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[1], type:

     `propagate=yes`

     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:

1. The following are call-party–associated events: `EventPartyChanged`, `EventPartyDeleted`, and `EventPartyAdded`.

```
propagate=party
```

- To disable distribution of both user data associated with the call and call-party–associated events, type:

```
propagate=no
```

5. Click `OK` to save configuration updates and close the `Switch Access Code Properties` dialog box.
6. Click `Apply` and `OK` to save configuration updates and close the `Switch Properties` dialog box.

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 7 Events and Models Reference Manual* and *Voice Platform SDK 8.0 .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 35, as well as previous sections of this chapter on multi-site deployment. In particular, Table 39 on page 409 shows which transaction types are supported by a specific T-Server, while Table 40 on page 414 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 Configuration Manager to create and partially configure each T-Server object. Review multi-site option values in the "Multi-Site Support Section" on page 479 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 442 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 Configuration Manager 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. Open the T-Server Application's Properties dialog box.
2. Click the Connections tab and click Add to add a connection to the appropriate T-Server. The Connection Info Properties dialog box displays.
3. 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

4. Click the 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.

5. Open the extrouter section. Configure the options used for multi-site support.

> **Note:** For a list of options and valid values, see "Multi-Site Support Section" on page 479, in the "T-Server Common Configuration Options" chapter.

6. When you are finished, click Apply.
7. 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 8.0, or when using T-Servers of different releases (including 8.0) in the same environment, see "Compatibility Notes" on page 441.

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).
7. When you are finished, click Apply.

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 42:

Table 42: 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 432.
direct-network-callid=<>	For configuration information, see your *T-Server Deployment Guide*. (Use Table 40 on page 414 to determine if your T-Server supports the direct-network-callid transaction type.)

   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 43:

Table 43: 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 44 contains cross-reference information on transaction types that the Configuration Layer and T-Server use.

Table 44: 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

Table 44: Route Type and ISCC Transaction Type Cross-Reference (Continued)

Route Type Field Value	ISCC Transaction Type
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

8. When you are finished, click Apply.

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 8.0, or when using T-Servers of different releases (including 8.0) 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

Overview

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.

Procedure:
Configuring access resources for ISCC/COF

Purpose: To configure dedicated DNs required for the ISCC/COF feature.

Start of procedure

Note: Use Table 40 on page 414 to determine if your T-Server supports the ISCC/COF feature.

1. Under a configured `Switch`, select the `DNs` folder. From the main menu, select `File > New > DN` to create a new `DN` object.

   Note: The number of the access resource must match the name of a DN configured on the switch (usually, an ACD Queue) so that T-Server can determine whether the calls arriving to this DN are overflowed calls.

2. On the `General` tab of the DN `Properties` dialog box, specify the number of the configured DN as the value for the `Number` field.
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 436.

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 442.
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 438.

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 438.

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.

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:

- Setting Configuration Options, page 450
- Mandatory Options, page 450
- Log Section, page 450
- Log-Extended Section, page 463
- Log-Filter Section, page 465
- Log-Filter-Data Section, page 465
- Security Section, page 465
- SML Section, page 466
- Common Section, page 467
- Changes from 7.6 to 8.0, page 467

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 in the `Options` tab of the `Application` object.

Warning! Configuration section names, configuration option names, and predefined option values are case-sensitive. Type them in Configuration Manager 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 455.

Note: For definitions of the Standard, Interaction, Trace, and Debug log levels, refer to the *Framework 8.0 Management Layer User's Guide*, *Framework 8.0 Genesys Administrator Help,* or to *Framework 8.0 Solution Control Interface 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 455). 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`.

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 8.0 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 455). 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 455.

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.

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 459.

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 Technical Support 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.

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 8.0 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 8.0 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 8.0 Security Deployment Guide* for complete information about this option.

SML Section

This section must be called `sml`.

Options in this section are defined in the `Annex` tab of the `Application` object.

Warning! Use the three options in this section (`heartbeat-period`, `heartbeat-period-thread-class-<n>`, and `hangup-restart`) with great care, and only with those applications for 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 the Management Layer 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 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 notify that the application has stopped responding.

Common Section

This section must be called `common`.

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 Technical Support.

Changes from 7.6 to 8.0

Table 45 on page 468 provides all the changes to common configuration options between release 7.6 and the latest 8.0 release.

Table 45: Common Configuration Option Changes from 7.6 to 8.0

Option Name	Option Values	Type of Change	Details
log Section			
expire	Increased range of value	Modified	See description on page 451.
log-filter Section			
default-filter-type	Additional option values	Modified	See description on page 465.
log-filter-data Section			
<key name>	Additional option values	Modified	See description on page 465.
security Section (New Section)			
disable-rbac	true, false	New	See description on page 465.
sml Section			
heartbeat-period	3–604800 seconds	New	See description on page 466.
heartbeat-period-thread-class-<n>	3–604800 seconds	New	See description on page 466.
hangup-restart	true, false	New	See description on page 467.

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:

- Setting Configuration Options, page 469
- Mandatory Options, page 470
- T-Server Section, page 470
- License Section, page 475
- Agent-Reservation Section, page 478
- Multi-Site Support Section, page 479
- Translation Rules Section, page 489
- Backup-Synchronization Section, page 490
- Call-Cleanup Section, page 491
- Security Section, page 493
- Timeout Value Format, page 493
- Changes from Release 7.6 to 8.0, page 494

T-Server also supports common log options described in Chapter 10, "Common Configuration Options," on page 449.

Setting Configuration Options

Unless it is specified otherwise, you set configuration options in Configuration Manager in the corresponding sections on the `Options` tab for the T-Server `Application` object.

Mandatory Options

Except as noted for certain environments, the configuration of common options is not required for basic T-Server operation.

T-Server Section

The T-Server section contains the configuration options that are used to support the core features common to all T-Servers.

TServer 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: `false`
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.

Note: Use of this option can negatively impact T-Server processing speed.

background-timeout

Default Value: `60 msec`
Valid Values: See "Timeout Value Format" on page 493.
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 checks whether a client provides the correct name and password of a tenant. If it does, 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`

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-8.0 T-Server behavior applies.

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 471.)

propagated-call-type

Default Value: `false`

Valid Values: `true, false`

Changes Take Effect: Immediately

When set to `false`, T-Server reports a value in the `CallType` attribute as it did in pre-8.0 releases and extends distribution of call-related TEvents that contain the `PropagatedCallType` attribute (if known). This provides backward compatibility with existing T-Server clients.

When set to `true`, T-Server extends distribution of call-related TEvents that contain a call type value in the `LocalCallType` attribute (as in a single-site T-Server deployment) and replaces the value of the regular `CallType` attribute with the `PropagatedCallType` value.

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 476.

license This section must be called `license`.

Notes: T-Server also supports the `license-file` option described in the *Genesys Licensing Guide*.

The `license` section is not applicable to Network T-Server for DTAG.

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` (all available licenses)
Valid Values: `0` or string `max`
Changes Take Effect: Immediately

Specifies how many DN licenses T-Server checks out. T-Server treats a value of `0` (zero) the same as it treats `max`—that is, it checks out all available licenses.

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 from `0`–`9999`
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 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 46 shows how to determine the number of seat-related DN licenses that T-Server attempts to check out. See the examples on page 477.

Table 46: License Checkout Rules

Options Settings[a]			License Checkout[b]
num-of-licenses	num-sdn-licenses		Seat-related DN licenses
max (or 0)	max		all available
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` is the maximum number of licenses that T-Server can check out; `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, and it is limited to 9999.

Examples

This section presents examples of option settings in the `license` section.

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		

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		

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		

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 388 section for details on this feature.

agent-reservation 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-8.0 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 493.
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 493.
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.

Multi-Site Support Section

The Multi-Site Support 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:

- ISCC Transaction Options, page 481
- Transfer Connect Service Options, page 485
- ISCC/COF Options, page 486
- Event Propagation Options, page 488
- Number Translation Option, page 489

extrouter 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.

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: Genesys does not recommend `match-call-once` option to `false` . to excessive or inconsistent call data updates

reconnect-tout

Default Value: `5 sec`
Valid Values: See "Timeout Value Format" on page 493.
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 39 on page 409 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 493.
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 493.
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 493.
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 `Resource Type 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 493.
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.

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 the `UserData` 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 493.
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 493.
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 493.
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 493.
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-8.0 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 493.
Changes Take Effect: Immediately

Specifies the time interval 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.

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 418. See "Configuring Number Translation" on page 425 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
```

Backup-Synchronization Section

The Backup-Synchronization section contains the configuration options that are used to support a high-availability (`hot standby` redundancy type) configuration.

backup-sync 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 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 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 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.

sync-reconnect-tout

Default Value: `20 sec`
Valid Values: See "Timeout Value Format" on page 493.
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 "Stuck Call Management" chapter in the *Framework 8.0 Management Layer User's Guide*.

call-cleanup This section must be called `call-cleanup`.

cleanup-idle-tout

Default Value: `0`
Valid Values: See "Timeout Value Format" on page 493.
Changes Take Effect: Immediately

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 493.
Changes Take Effect: Immediately

Specifies the time interval 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 493.
Changes Take Effect: Immediately

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.

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 8.0 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 7.6 to 8.0

Table 47 lists the configuration options that:

- Are new or changed in the 8.0 release of T-Server
- Have been added or changed since the most recent 7.6 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 47: Option Changes from Release 7.6 to 8.0

Option Name	Option Values	Type of Change	Details
TServer Section			
dn-scope	`switch, office, tenant`	New in 8.0	See the option description on page 472.
propagated-call-type	`true, false`	New in 8.0	See the option description on page 474.
extrouter Section			
compound-dn-representation	`true, false`	New in 8.0	See the option description on page 488.
default-network-call-id-matching	No default value	See Details	This option is undocumented in previous versions. See the option description on page 487.
epp-tout	Timeout value format	New in 8.0	See the option description on page 488.
use-data-from	`active, original, current, active-data-original-call`	New default value	New default value: `current`. Old default value: `active`. See the option description on page 481.
agent-reservation Section			
collect-lower-priority-requests	`true, false`	New in 8.0	See the option description on page 478.

Related Documentation Resources

The following resources provide additional information that is relevant to this software. Consult these additional resources as necessary.

SIP Server

- The *Framework 8.0 Deployment Guide*, which will help you configure, install, start, and stop Framework components.
- The *Framework 7.6 Stream Manager Deployment Guide*, which will help you configure, install, start, and stop Stream Manager.

Management Framework

- The *Framework 8.0 Configuration Options Reference Manual*, which will provide you with descriptions of configuration options for other Framework components.
- The *Framework 8.0 Configuration Manager Help*, which will help you use Configuration Manager.

Genesys

- The *Genesys 7 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 ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms used in this document.

- *Genesys Migration Guide*, which ships on the Genesys Documentation Library DVD, and which provides documented migration strategies for Genesys product releases. Contact Genesys Technical Support for more information.
- Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at http://genesyslab.com/support.

Information about supported hardware and third-party software is available on the Genesys Technical Support website in the following documents:

- *Genesys Supported Operating Environment Reference Manual*
- *Genesys Supported Media Interfaces Reference Manual*

Consult these additional resources as necessary:

- *Genesys Hardware Sizing Guide,* which provides information about Genesys hardware sizing guidelines for the Genesys 7.x and 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 relevant to the Genesys licensing system.
- *Genesys Database Sizing Estimator 7.6 Worksheets,* which provides a range of expected database sizes for various Genesys products.

For additional system-wide planning tools and information, see the release-specific listings of System Level Documents on the Genesys Technical Support website, accessible from the system level documents by release tab in the Knowledge Base Browse Documents Section.

Genesys product documentation is available on the:

- Genesys Technical Support website at http://genesyslab.com/support.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.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:

80fr_ref_06-2008_v8.0.001.00

You will need this number when you are talking with Genesys Technical Support 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 48 describes and illustrates the type conventions that are used in this document.

Table 48: 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 498).	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 . . .

Table 48: Type Styles (Continued)

Type Style	Used For	Examples
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>`

Index

Symbols

[] (square brackets) 498
< > (angle brackets) 498
<key name>
  common log option 465, 468

Numerics

1pcc transfer 120
3pcc transfer 120
3rd-party known limitations 275

A

accept-dn-type
  configuration option 369
Access Code
  configuration 438
  defined 45, 436
acw-in-idle-force-ready
  configuration option 277
addp-remote-timeout
  configuration option 490
addp-timeout
  configuration option 490
addp-trace
  configuration option 490
Advanced Disconnect Detection Protocol . . 383
after-routing-timeout
  configuration option 278
Agent
  agent no-answer supervision 201
  agent-greeting 330
  agent-no-answer-action 279
  agent-no-answer-overflow 279
  agent-no-answer-timeout 280
  agent-strict-id
    configuration option 280
Agent Login objects 45
agent reservation
  defined 388
agent-greeting
  configuration option 330
agent-group
  configuration option 278
  emulated agent options 278
agent-no-answer-action
  configuration option 279
agent-no-answer-overflow
  configuration option 279
agent-no-answer-timeout
  configuration option 280
Agent-Reservation section
  configuration options 478–479
alarm
  common log option 456
all
  common log option 455
alternate gateway selection 248
am-detected
  configuration option 281
angle brackets 498
ANI 401
ani-distribution
  configuration option 470
app
  command line parameter 51
Application objects
  multi-site operation 435
application server mode 28
AssistMode 265
associating ACD Queue with Routing Point . . 93
Asterisk-based Voice Mail
  agent group registration 96
  agent login registration 96
  extension registration 95
  feature configuration 96
  Genesys voice mail configuration adapter . 95
  message waiting indicator 95
  overview 94

audio-codecs
configuration option 281, 330
authenticate-requests
configuration option 330
auto-logout-ready (application level)
configuration option 282
auto-logout-ready (switch/DN level)
configuration option 331
auto-logout-timeout (application level)
configuration option 282
auto-logout-timeout (switch/DN level)
configuration option 331
auto-redirect-enabled
configuration option 332

B

background-processing
configuration option 470
background-timeout
configuration option 471
Back-to-Back User Agent 25
backup-mode
configuration option 369
Backup-Synchronization section
configuration option 490–491
backwds-compat-acw-behavior
configuration option 282
bandwidth requirements 39
beep-duration
configuration option 332
brackets
angle 498
square 498
buffering
common log option 451
busy-tone
configuration option 284
busy-tone-duration
configuration option 284

C

Call Recording
emergency 100
feature configuration 99
overview 97
Call Supervision
assistance request 106
configuration options 108
feature configuration 104
feature limitations 109
hiding supervisor presence 107
intrusion 104
monitoring session 103
multi-site supervision 109
scopes 102
supervisor auto-release 106
types 102
Call-Cleanup section
configuration option 491–493
call-max-outstanding
configuration option 369
call-rq-gap
configuration option 284
Calls Outside the Premise
find-trunk-by-location 294
cancel-monitor-on-disconnect
configuration option 284
capacity
configuration option 332
capacity-group
configuration option 333
cast-type
configuration option 400, 481
CDN 407
changes
SIP configuration options 370
changes from 7.6 to 8.0
common configuration options 467
configuration options 494
charge-type
configuration option 333
check-point
common log option 454
check-tenant-profile
configuration option 471
Class of Service 126–137
dialing rule format 128
feature configuration 134
outbound dialing rules 127
ring-through rules 134
transformation 131
cleanup-idle-tout
configuration option 491
clid-withheld-name
configuration option 369
Code property 438, 439
cof-ci-defer-create
configuration option 486
cof-ci-defer-delete
configuration option 486
cof-ci-req-tout
configuration option 416, 486
cof-ci-wait-all
configuration option 486
cof-feature
configuration option 487
cof-rci-tout
configuration option 487
collect-lower-priority-requests

configuration option . . . 478
collect-tone
configuration option . . . 284
command line parameters . . . 51
app . . . 51
host . . . 51
l . . . 52
lmspath . . . 52
nco X/Y . . . 52
port . . . 51
V . . . 52
commenting on this document . . . 16
common configuration options . . . 450–468
changes from 7.6 to 8.0 . . . 467
common section . . . 467
disable-rbac . . . 465, 468
hangup-restart . . . 467, 468
heartbeat-period . . . 466, 468
heartbeat-period-thread-class-<n> . . . 466, 468
log section . . . 450–463
log-extended section . . . 463–465
log-filter section . . . 465
log-filter-data section . . . 465
mandatory . . . 450
rebind-delay . . . 467
security section . . . 465
setting . . . 450
sml section . . . 466–467
common log options . . . 450–465
<key name> . . . 465, 468
alarm . . . 456
all . . . 455
buffering . . . 451
check-point . . . 454
debug . . . 458
default-filter-type . . . 465, 468
expire . . . 451, 468
interaction . . . 457
level-reassign-<eventID> . . . 463
level-reassign-disable . . . 465
log section . . . 450–463
log-extended section . . . 463–465
log-filter section . . . 465
log-filter-data section . . . 465
mandatory options . . . 450
memory . . . 454
memory-storage-size . . . 454
message_format . . . 452
messagefile . . . 452
print-attributes . . . 453
segment . . . 451
setting . . . 450
spool . . . 455
standard . . . 457
time_convert . . . 453
time_format . . . 453
trace . . . 457
verbose . . . 450
x-conn-debug-all . . . 462
x-conn-debug-api . . . 462
x-conn-debug-dns . . . 462
x-conn-debug-open . . . 460
x-conn-debug-security . . . 462
x-conn-debug-select . . . 461
x-conn-debug-timers . . . 461
x-conn-debug-write . . . 461
common options
common log options . . . 450–465
common section . . . 467
mandatory options . . . 450
sml section . . . 466–467
common section
common options . . . 467
compound-dn-representation
configuration option . . . 488
Configuration Manager
configuring T-Server . . . 46
configuration options
accept-dn-type . . . 369
acw-in-idle-force-ready . . . 277
addp-remote-timeout . . . 490
addp-timeout . . . 490
addp-trace . . . 490
after-routing-timeout . . . 278
agent-greeting . . . 330
agent-group . . . 278
agent-no-answer-action . . . 279
agent-no-answer-overflow . . . 279
agent-no-answer-timeout . . . 280
Agent-Reservation section . . . 478–479
agent-strict-id . . . 280
am-detected . . . 281
ani-distribution . . . 470
audio-codecs . . . 281, 330
authenticate-requests . . . 330
auto-logout-ready (application level) . . . 282
auto-logout-ready (switch/DN level) . . . 331
auto-logout-timeout (application level) . . . 282
auto-logout-timeout (switch/DN level) . . . 331
auto-redirect-enabled . . . 332
background-processing . . . 470
background-timeout . . . 471
backup-mode . . . 369
Backup-Synchronization section . . . 490–491
backwds-compat-acw-behavior . . . 282
beep-duration . . . 332
busy-tone . . . 284
busy-tone-duration . . . 284
Call-Cleanup section . . . 491–493
call-max-outstanding . . . 369
call-rq-gap . . . 284
cancel-monitor-on-disconnect . . . 284

capacity . . . 332
capacity-group . . . 333
cast-type . . . 481
changes from 7.6 to 8.0 . . . 494
charge-type . . . 333
check-tenant-profile . . . 471
cleanup-idle-tout . . . 491
clid-withheld-name . . . 369
cof-ci-defer-create . . . 486
cof-ci-defer-delete . . . 486
cof-ci-req-tout . . . 486
cof-ci-wait-all . . . 486
cof-feature . . . 487
cof-rci-tout . . . 487
collect-lower-priority-requests . . . 478
collect-tone . . . 284
common log options . . . 450–465
common options . . . 450–468
compound-dn-representation . . . 488
consult-user-data . . . 471
contact . . . 334
contacts-backup . . . 335
correct-rqid . . . 369
cos . . . 335
cpd-info-timeout . . . 285
cpn . . . 336
customer-greeting . . . 336
customer-id . . . 472
default-dn . . . 482
default-dn (application level) . . . 286
default-dn (switch/DN level) . . . 337
default-dn-type . . . 369
default-monitor-mode . . . 286
default-monitor-scope . . . 287
default-music . . . 337
default-music (application level) . . . 287
default-music (switch/DN level) . . . 337
default-network-call-id-matching . . . 328, 487
default-video-file . . . 287
dial-plan (application level) . . . 287
dial-plan (switch/DN level) . . . 337
direct-digits-key . . . 482
display-name . . . 339
divert-on-ringing . . . 288
dn-del-mode . . . 369
dn-for-unexpected-calls . . . 482
dn-scope . . . 430, 472
dtmf-payload . . . 288
dual-dialog-enabled . . . 339
emergency-recording-cleanup-enabled . . . 288
emergency-recording-filename . . . 288
emulated-login-state . . . 289
emulate-login . . . 369
enable-agentlogin-presence . . . 339
enable-agentlogin-subscribe . . . 290
enable-ims . . . 340, 370
enable-unknown-gateway . . . 290, 291
encoding . . . 290
enforce-external-domains . . . 291
enforce-trusted . . . 291, 340
epp-tout . . . 431, 488
event-propagation . . . 489
event-ringing-on-100trying . . . 291
expire-call-tout . . . 370
extensions-<n> . . . 180, 327
external-registrar . . . 292
extn-no-answer-overflow . . . 292
ext-no-answer-action . . . 293
fast-busy-tone . . . 293
fax-detected . . . 293
find-trunk-by-location . . . 294
forced-notready . . . 294
force-register . . . 340
geo-location . . . 341
greeting-after-merge . . . 295
greeting-call-type-filter . . . 295
greeting-delay-events . . . 295
greeting-notification . . . 295
greeting-repeat-once-party . . . 296
handle-vsp . . . 369
ims-3pcc-prefix . . . 296
ims-default-icid-prefix . . . 297
ims-default-icid-suffix . . . 297
ims-default-orig-ioi . . . 297
ims-route . . . 297
ims-skip-ifc . . . 298
inbound-bsns-calls . . . 298
inbound-translator-<n> . . . 489
include-dial-plan- . . . 342
info-pass-through (application level) . . . 298
info-pass-through (switch/DN level) . . . 342
inherit-bsns-type . . . 299
init-dnis-by-ruri . . . 370
internal-bsns-calls . . . 299
internal-registrar-domains . . . 299
internal-registrar-enabled . . . 299
internal-registrar-persistent . . . 300
intrusion-enabled . . . 300
kpl-interval . . . 370
kpl-loss-rate . . . 370
kpl-tolerance . . . 370
legal-guard-time . . . 300
License section . . . 475–478
local-node-id . . . 487
logout-on-disconnect . . . 301
log-trace-flags . . . 473
make-call-alert-info . . . 301
make-call-rfc3725-flow . . . 344
management-port . . . 473
mandatory options . . . 450
map-sip-errors . . . 301
match-call-once . . . 480

max-legs-per-sm . . . 301
max-pred-req-delay . . . 370
merged-user-data . . . 473
monitor-internal-calls . . . 301
Multi-Site Support section . . . 479–489
music-in-conference-file . . . 302
music-in-queue-file . . . 303
mwi-agent-enable . . . 303
mwi-domain . . . 303
mwi-extension-enable . . . 303
mwi-group-enable . . . 304
mwi-host . . . 304
mwi-mode . . . 304
mwi-port . . . 304
nas-indication . . . 370
nas-private . . . 305
network-request-timeout . . . 482
no-answer-action . . . 344
no-answer-overflow . . . 344
no-answer-timeout . . . 345
notify-idle-tout . . . 492
num-of-licenses . . . 475
num-sdn-licenses . . . 476
observing-routing-point . . . 305
ocs-dn . . . 346
oos-check . . . 346
oos-force . . . 347
oosp-transfer-enabled . . . 347
outbound-bsns-calls . . . 305
out-rule-<n> . . . 348
overload-ctrl-call-rate-capacity . . . 306
overload-ctrl-dialog-rate-capacity . . . 306
overload-ctrl-threshold . . . 306
override-call-type . . . 348
override-domain . . . 348
override-domain-from . . . 348
override-switch-acw . . . 370
override-to-on-divert . . . 306
parking-music . . . 307
partition-id (application level) . . . 307
partition-id (switch/dn level) . . . 349
p-asserted-identity . . . 349
password . . . 349
periodic-check-tout . . . 492
posn-no-answer-overflow . . . 307
posn-no-answer-timeout . . . 308
prd-dist-call-ans-time . . . 370
predictive-call-router-timeout . . . 308
prefix . . . 350
preview-expired . . . 309
preview-interaction . . . 350
priority . . . 351
privacy . . . 351
private-line . . . 351
privilege-level . . . 352
propagated-call-type . . . 430, 474
protocol . . . 491
public-contact . . . 352
quiet-cleanup . . . 370
quiet-startup . . . 370
recall-no-answer-timeout . . . 370
reconnect-tout . . . 480
record . . . 352
record-consult-calls . . . 309
recording-filename . . . 309
recovery-timeout . . . 353
refer-enabled . . . 354
reg-delay . . . 370
reg-interval . . . 370
register-attempts . . . 483
register-tout . . . 483
registrar-default-timeout . . . 310
reg-silent . . . 370
reinvite-requires-hold . . . 354
reject-call-incall . . . 355
reject-call-notready . . . 355
reject-subsequent-request . . . 478
replace-prefix . . . 355
report-connid-changes . . . 480
request-collection-time . . . 479
request-tout . . . 483
request-uri . . . 356
reservation-time . . . 479
resource-allocation-mode . . . 483
resource-load-maximum . . . 484
restricted . . . 369
reuse-sdp-on-reinvite . . . 356
rfc-2976-dtmf . . . 353
ringing-on-route-point . . . 311
ring-tone . . . 310
ring-tone-on-make-call . . . 356
route-dn . . . 484
router-timeout . . . 311
rq-expire-tmout . . . 370
rule-<n> . . . 489
Security section . . . 493
server-id . . . 474
server-role . . . 311
service-type . . . 357
session-refresh-interval . . . 311
set-notready-on-busy . . . 312
setting . . . 469
  common . . . 450
silence-tone . . . 312
sip-add-primary-call-id . . . 358
sip-address . . . 312
sip-block-headers . . . 312
sip-busy-type . . . 358
sip-call-retain-timeout . . . 313
sip-chat-format . . . 358
sip-cti-control . . . 359
sip-dtmf-send-rtp . . . 313

sip-early-dialog-mode . . . 359
sip-enable-100rel . . . 313
sip-enable-call-info . . . 313
sip-enable-gdns . . . 360
sip-enable-moh (application level) . . . 314
sip-enable-moh (switch/DN level) . . . 360
sip-enable-sdp-codec-filter . . . 314, 361
sip-hold-rfc3264 (application level) . . . 314
sip-hold-rfc3264 (switch/DN level) . . . 361
sip-invite-timeout . . . 315
sip-invite-treatment-timeout . . . 315
sip-ip-tos . . . 315
sip-pass-check . . . 317
sip-port . . . 317
sip-port-tls . . . 317
sip-preserve-contact (application level) . . . 318
sip-preserve-contact (switch/DN level) . . . 362
sip-proxy-headers-enabled . . . 318
sip-replaces-mode . . . 362
sip-retry-timeout . . . 319
sip-ring-tone-mode (application level) . . . 319
sip-ring-tone-mode (switch/DN level) . . . 363
sip-route (DN level) . . . 363
sip-server-inter-trunk . . . 363
sip-signaling-chat . . . 364
sip-tls-cert . . . 319
sip-tls-cert-key . . . 319
sip-tls-cipher-list . . . 320
sip-tls-crl . . . 321
sip-tls-mutual . . . 322
sip-tls-target-name-check . . . 322
sip-tls-trusted-ca . . . 322
sip-treatments-continuous . . . 322
stranded -calls-overflow . . . 323
stranded-call-redirection-limit (application level) . . . 323
stranded-calls-overflow (switch/DN level) . . . 364
stranded-on-arrival-calls-overflow (application level) . . . 323
stranded-on-arrival-calls-overflow (switch/DN level) . . . 364
subscribe-presence . . . 367
subscribe-presence-domain . . . 366
subscribe-presence-expire . . . 366
subscribe-presence-from . . . 366
subscription-delay . . . 324
subscription-id . . . 365
subscription-timeout . . . 324
sync-emu-agent . . . 370
sync-reconnect-tout . . . 491
tcs-queue . . . 485
tcs-use . . . 485
timed-acw-in-idle . . . 324
timeout . . . 484
timeout value format . . . 493
tlib-map-replace-dn . . . 325
transfer-complete-by-refer . . . 367
Translation Rules section . . . 489
T-Server section . . . 470–475
unknown-bsns-calls . . . 325
unknown-xfer-merge-udata . . . 370
untimed-wrap-up-value . . . 325
use-contact-as-dn . . . 367
use-data-from . . . 481
use-display-name . . . 368, 370
use-implicit-access-numbers . . . 485
userdata-<n> . . . 328
user-data-im-enabled . . . 325
user-data-limit . . . 475
userdata-map-filter . . . 368
userdata-map-trans-prefix . . . 326
use-register-for-service-state . . . 368
wrap-up-threshold . . . 370
wrap-up-time . . . 326
configure
ACD Queue . . . 63
application service . . . 74
device types . . . 60
emergency recording . . . 76
endpoint . . . 65
external music servers . . . 70
gateway . . . 68
MCU . . . 63
proxies . . . 72
Recorder servers . . . 75
Routing Point . . . 72
soft switches . . . 72
Stream Manager . . . 80
switch objects . . . 62
treatment service . . . 77
configuring
Agent Login options . . . 330
DN options . . . 330
INFO . . . 327
INVITE . . . 327
multi-site operation . . . 435
steps . . . 435
T-Server . . . 46
TServer section . . . 277
UPDATE . . . 327
consult-user-data
configuration option . . . 471
contact
configuration option . . . 334
contacts-backup
configuration option . . . 335
conventions
in document . . . 497
type styles . . . 497
correct-rqid
configuration option . . . 369
cos

configuration option 335
cpd-info-timeout
configuration option 285
cpn
configuration option 336
CPNDigits 268
customer-greeting
configuration option 336
customer-id
configuration option 472
customer-side proxy mode 28
customized music files 214

D

debug
common log option 458
Default Access Code
configuration 437
defined 436
default-dn
configuration option 482
configuration option (application level) . . 286
configuration option (switch/DN level) . . . 337
default-dn-type
configuration option 369
default-filter-type
common log option 465, 468
default-monitor-mode
configuration option 286
default-monitor-scope
configuration option 287
default-music
configuration option 337
configuration option (switch/DN level) . . . 337
default-music (application level)
configuration option 287
default-network-call-id-matching
configuration option 328, 487
default-video-file
configuration option 287
destination location 394
destination T-Server 400
dial-plan
configuration option (application level) . . 287
configuration option (switch/DN level) . . . 337
direct-ani
ISCC transaction type 401, 409
direct-callid
ISCC transaction type 402, 409
direct-digits
transaction type 409
direct-digits-key
configuration option 482
direct-network-callid
ISCC transaction type 402, 409
direct-notoken
ISCC transaction type 403, 409
direct-uui
ISCC transaction type 403, 409
disable-rbac
common configuration option 465, 468
display-name
configuration option 339
divert-on-ringing
configuration option 288
DN objects 45
dn-del-mode
configuration option 369
dn-for-unexpected-calls
configuration option 482
dnis-pool
in load-balancing mode 405
ISCC transaction type 396, 404, 409
DNs
configuring for multi-sites 442
dn-scope
configuration option 430, 472
document
conventions 497
errors, commenting on 16
version number 497
dtmf-payload
configuration option 288
dual-dialog-enabled
configuration option 339

E

emergency-recording-cleanup-enabled
configuration option 288
emergency-recording-filename
configuration option 288
emulated agent options
agent-group 278
untimed-wrap-up-value 325
Emulated Agents 162–166
emulated agents
timed-acw-in-idle 324
emulated-login-state
configuration option 289
emulate-login
configuration option 369
enable-agentlogin-presence
configuration option 339
enable-agentlogin-subscribe
configuration option 290
enable-ims
configuration option 340, 370
enable-unknown-gateway
configuration option 290, 291
encoding

configuration option 290
endpoint monitoring 167
enforce-external-domains
configuration option 291
enforce-trusted
configuration option 291, 340
epp-tout
configuration option 431, 488
error messages 272
Event Propagation
defined 427
EventAttachedDataChanged 428
event-propagation
configuration option 489
event-ringing-on-100trying
configuration option 291
expire
common log option 451, 468
expire-call-tout
configuration option 370
Extension
AssistMode 265
attribute 264
CPNDigits 268
MonitorMode 264
MonitorScope 265
music 266
NO_ANSWER_ACTION 267
NO_ANSWER_OVERFLOW 267
NO_ANSWER_TIMEOUT 267
no-answer supervision 201
record 264
extension-<n>
configuration option 180, 327
external-registrar
configuration option 292
extn-no-answer-overflow
configuration option 292
ext-no-answer-action
configuration option 293
extrouter
configuration section 425, 432, 436

F

fast-busy-tone
configuration option 293
fax-detected
configuration option 293
Features Summary 30
figures
Multiple-to-Point mode 408
Point-to-Point mode 407
steps in ISCC/Call Overflow 415
find-trunk-by-location
configuration option 294
first-party call control transfer 120
font styles
italic 497
monospace 498
forced-notready
configuration option 294
force-register
configuration option 340

G

geo-location
configuration option 341
greeting-after-merge
configuration option 295
greeting-call-type-filter
configuration option 295
greeting-delay-events
configuration option 295
greeting-notification
configuration option 295
greeting-repeat-once-party
configuration option 296
gvm_group_mailbox 96
GVP integration 253
option 368

H

handle-vsp
configuration option 369
hangup-restart
common configuration option 467, 468
heartbeat-period
common configuration option 466, 468
heartbeat-period-thread-class-<n>
common configuration option 466, 468
high availability 30
host
command line parameter 51
hot standby 30, 384
defined 384

I

IM-related options
sip-chat-format 358
sip-signaling-chat 364
ims-3pcc-prefix
configuration option 296
ims-default-icid-prefix
configuration option 297
ims-default-icid-suffix
configuration option 297

ims-default-orig-ioi
configuration option . . . 297
IMS-related options
enable-ims . . . 297, 340
ims-3pcc-prefix . . . 296
ims-default-orig-ioi . . . 297
ims-route . . . 297
ims-skip-ifc . . . 298
override-domain . . . 348
refer-enabled . . . 354
registrar-default-timeout . . . 310
server-role . . . 311
sip-route (DN level) . . . 363
ims-route
configuration option . . . 297
ims-skip-ifc
configuration option . . . 298
inbound-bsns-calls
configuration option . . . 298
inbound-translator-<n>
configuration option . . . 489
include-dial-plan-
configuration option . . . 342
info-pass-through (application level)
configuration option . . . 298
info-pass-through (switch/DN level)
configuration option . . . 342
inherit-bsns-type
configuration option . . . 299
init-dnis-by-ruri
configuration option . . . 370
Instant Messaging
calling features . . . 172
feature configuration . . . 176
overview . . . 171
supported call operations . . . 172
transcript . . . 171
treatments . . . 174
Inter Server Call Control . . . 394–413
Inter Server Call Control/Call Overflow . . . 413–417
interaction
common log option . . . 457
internal-bsns-calls
configuration option . . . 299
internal-registrar-domains
configuration option . . . 299
internal-registrar-enabled
configuration option . . . 299
internal-registrar-persistent
configuration option . . . 300
intrusion-enabled
configuration option . . . 300
ISCC . . . 32
destination T-Server . . . 400
origination T-Server . . . 400
ISCC transaction types . . . 395, 400
direct-ani . . . 401, 409
direct-callid . . . 402, 409
direct-digits . . . 409
direct-network-callid . . . 402, 409
direct-notoken . . . 403, 409
direct-uui . . . 403, 409
dnis-pool . . . 404, 409
in load-balancing mode . . . 405
pullback . . . 405, 409
reroute . . . 406, 409
route . . . 407, 409
route-uui . . . 408
supported . . . 409
ISCC/COF
supported . . . 414
iscc-xaction-type . . . 395
italics . . . 497

K

known limitations . . . 274
kpl-interval
configuration option . . . 370
kpl-loss-rate
configuration option . . . 370
kpl-tolerance
configuration option . . . 370

L

l
command line parameter . . . 52
legal-guard-time
configuration option . . . 300
level-reassign-<eventID>
common log option . . . 463
level-reassign-disable
common log option . . . 465
License section
configuration options . . . 475–478
lmspath
command line parameter . . . 52
Load Balancing . . . 30
local-node-id
configuration option . . . 487
location parameter . . . 394
log configuration options . . . 450–??
log section
common log options . . . 450–463
log-extended section
common log options . . . 463–465
log-filter section
common log options . . . 465
log-filter-data section

common log options 465
logout-on-disconnect
configuration option 301
log-trace-flags
configuration option 473

M

make-call-alert-info
configuration option 301
make-call-rfc3725-flow
configuration option 344
Management Layer 40
management-port
configuration option 473
mandatory options
configuration options 470
mapping
extract data from INVITE message 177
line in SDP message body 177, 182
SIP headers 177
whole SDP message body 177, 182
map-sip-errors
configuration option 301
match-call-once
configuration option 480
max-legs-per-sm
configuration option 301
max-pred-req-delay
configuration option 370
media files, playing multiple 214
memory
common log option 454
memory-storage-size
common log option 454
merged-user-data
configuration option 473
Message Waiting Indicator
mwi-agent-enable 303
mwi-domain 303
mwi-extension-enable 303
mwi-group-enable 304
mwi-host 304
mwi-mode 304
mwi-port 304
message_format
common log option 452
messagefile
common log option 452
mode
application server 28
customer-side proxy 28
stand-alone 27
Monitor
MonitorMode 264
MonitorScope 265
monitor-internal-calls
configuration option 301
monospace font 498
Multiple-to-One mode 407
Multiple-to-Point mode 407, 408
multi-site supervision 109
Multi-Site Support section
configuration options 479–489
music 266
Music and Announcements
announcement treatments 186
music treatments 188
music-in-queue-file 303
other treatments 189
overview 186
music-in-conference-file
configuration option 302
music-in-queue-file
configuration option 303
mwi-agent-enable
configuration option 303
mwi-domain
configuration option 303
mwi-extension-enable
configuration option 303
mwi-group-enable
configuration option 304
mwi-host
configuration option 304
mwi-mode
configuration option 304
mwi-port
configuration option 304

N

nas-indication
configuration option 370
nas-private
configuration option 305
NAT/C feature 425
nco X/Y
command line parameter 52
Network
architecture 26
Network Asserted Identity 192
network attended transfer/conference 425
Network Considerations 38
bandwidth requirements 39
firewalls 40
remote agents 39
tuning 39
voice quality 38
network objects 40
network-request-timeout
configuration option 482

NO_ANSWER
NO_ANSWER_ACTION . . . 267
NO_ANSWER_OVERFLOW . . . 267
NO_ANSWER_TIMEOUT . . . 267
No-Answer Supervision
agent configuration . . . 201
agents . . . 201
extension configuration . . . 201
extensions . . . 201
position . . . 202
position configuration . . . 202
no-answer-action
configuration option . . . 344
no-answer-overflow
configuration option . . . 344
no-answer-timeout
configuration option . . . 345
notify-idle-tout
configuration option . . . 492
Number Translation feature . . . 417–425
number translation rules . . . 418
num-of-licenses
configuration option . . . 475
num-sdn-licenses
configuration option . . . 476

O

objects
Agent Logins . . . 45
DNs . . . 45
network . . . 40
Switches . . . 45
Switching Offices . . . 44
observing-routing-point
configuration option . . . 305
ocs-dn
configuration option . . . 346
One-to-One mode . . . 407
oos-check
configuration option . . . 346
oos-force
configuration option . . . 347
oosp-transfer-enabled
configuration option . . . 347
origination location . . . 394
origination T-Server . . . 400
Out Of Signaling Path
oosp-transfer-enabled . . . 347
outbound-bsns-calls
configuration option . . . 305
Out-of-Service
oos-check . . . 346
oos-force . . . 347
out-rule-<n>
configuration option . . . 348
overload control-related options
overload-ctrl-call-rate-capacity . . . 306
overload-ctrl-dialog-rate-capacity . . . 306
overload-ctrl-threshold . . . 306
overload-ctrl-call-rate-capacity
configuration option . . . 306
overload-ctrl-dialog-rate-capacity
configuration option . . . 306
overload-ctrl-threshold
configuration option . . . 306
override-call-type
configuration option . . . 348
override-domain
configuration option . . . 348
override-domain-from
configuration option . . . 348
override-switch-acw
configuration option . . . 370
override-to-on-divert
configuration option . . . 306

P

parking-music
configuration option . . . 307
partition-id (application level)
configuration option . . . 307
partition-id (switch/DN level)
configuration option . . . 349
p-asserted-identity
configuration option . . . 349
password
configuration option . . . 349
periodic-check-tout
configuration option . . . 492
Personal greeting
feature configuration . . . 212
overview . . . 211
Personal Greeting-related options
greeting-after-merge . . . 295
greeting-call-type-filter . . . 295
greeting-delay-events . . . 295
greeting-notification . . . 295
greeting-repeat-once-party . . . 296
playapplication, parameters
GSIP_APP_ID . . . 156
GSIP_DTMF_DURATION . . . 157
GSIP_DTMF_TO_DIAL . . . 156
Point-to-Point mode . . . 407
port
command line parameter . . . 51
position no-answer supervision . . . 202
posn-no-answer-overflow
configuration option . . . 307
posn-no-answer-timeout
configuration option . . . 308

prd-dist-call-ans-time
configuration option . . . 370
predictive-call-router-timeout
configuration option . . . 308
prefix
configuration option . . . 350
Presence Subscription
LCS . . . 217
overview . . . 216
processing userdata . . . 176
PUBLISH request . . . 218
subscribe-presence . . . 367
subscribe-presence-domain . . . 366
subscribe-presence-expire . . . 366
subscribe-presence-from . . . 366
subscription-id . . . 365
subscription-timeout . . . 324
updating agent state . . . 218
Preview interactions . . . 226
preview-expired
configuration option . . . 309
preview-interaction
configuration option . . . 350
print-attributes
common log option . . . 453
priority
configuration option . . . 351
privacy
configuration option . . . 351
privacy mechanism . . . 192
private-line
configuration option . . . 351
privilege-level
configuration option . . . 352
propagated-call-type
configuration option . . . 430, 474
protocol
configuration option . . . 491
Providing a caller ID . . . 228
public-contact
configuration option . . . 352
pullback
ISCC transaction type . . . 405, 409

Q

quiet-cleanup
configuration option . . . 370
quiet-startup
configuration option . . . 370

R

rebind-delay
common configuration option . . . 467
recall-no-answer-timeout
configuration option . . . 370
reconnect-tout
configuration option . . . 480
record . . . 264
configuration option . . . 352
record-consult-calls
configuration option . . . 309
recording-filename
configuration option . . . 309
recovery-timeout
configuration option . . . 353
redundancy
hot standby . . . 30, 384
warm standby . . . 30, 384
redundancy types
hot standby . . . 384
refer-enabled
configuration option . . . 354
reg-delay
configuration option . . . 370
reg-interval
configuration option . . . 370
register-attempts
configuration option . . . 483
register-tout
configuration option . . . 483
registrar-default-timeout
configuration option . . . 310
reg-silent
configuration option . . . 370
reinvite-requires-hold
configuration option . . . 354
reject-call-incall
configuration option . . . 355
reject-call-notready
configuration option . . . 355
reject-subsequent-request
configuration option . . . 478
remote agents . . . 39
Remote Server Registration . . . 230
Remote Supervision
feature configuration . . . 112
feature limitations . . . 120
overview . . . 112
use of Extensions attribute . . . 120
Remote Talk . . . 230
replace-prefix
configuration option . . . 355
report-connid-changes
configuration option . . . 480
request-collection-time
configuration option . . . 479
request-tout
configuration option . . . 396, 483
request-uri

configuration option 356
reroute
ISCC transaction type 406, 409
reservation-time
configuration option 479
resource-allocation-mode
configuration option 483
resource-load-maximum
configuration option 484
restricted options
configuration options 369
reuse-sdp-on-reinvite
configuration option 356
rfc-2976-dtmf
configuration option 353
ringing-on-route-point
configuration option 311
ring-tone
configuration option 310
ring-tone-on-make-call
configuration option 356
route
ISCC transaction type . . . 396, 407, 409, 442
route-dn
configuration option 484
router-timeout
configuration option 311
route-uui
ISCC transaction type 408
routing
Inter Server Call Control 400–413
rq-expire-tmout
configuration option 370
rule-<n>
configuration option 489
run.bat 55
run.sh 54

S

Security section
configuration option 493
security section
common configuration options 465
segment
common log option 451
selecting an alternate gateway 248
server-id
configuration option 474
server-role
configuration option 311
service-type
configuration option 357
session-refresh-interval
configuration option 311
set-notready-on-busy
configuration option 312
setting
configuration options 469
setting configuration options
common 450
silence-tone
configuration option 312
sip-add-primary-call-id
configuration option 358
sip-address
configuration option 312
sip-block-headers
configuration option 312
sip-busy-type
configuration option 358
sip-call-retain-timeout
configuration option 313
sip-chat-format
configuration option 358
sip-cti-control
configuration option 359
sip-dtmf-send-rtp
configuration option 313
sip-early-dialog-mode
configuration option 359
sip-enable-100rel
configuration option 313
sip-enable-call-info
configuration option 313
sip-enable-gdns
configuration option 360
sip-enable-moh
configuration option (application level) . . 314
configuration option (switch/DN level) . . 360
sip-enable-sdp-codec-filter
configuration option 314, 361
sip-hold-rfc3264 (application level)
configuration option 314
sip-hold-rfc3264 (switch/DN level)
configuration option 361
sip-invite-timeout
configuration option 315
sip-invite-treatment-timeout
configuration option 315
sip-ip-tos
configuration option 315
sip-pass-check
configuration option 317
sip-port
configuration option 317
sip-port-tls
configuration option 317
sip-preserve-contact (application level)
configuration option 318
sip-preserve-contact (switch/DN level)
configuration option 362

sip-proxy-headers-enabled
configuration option . . . 318
sip-replaces-mode
configuration option . . . 362
sip-retry-timeout
configuration option . . . 319
sip-ring-tone-mode
configuration option (application level) . . 319
configuration option (switch/DN level) . . . 363
sip-route (DN level)
configuration option . . . 363
sip-server-inter-trunk
configuration option . . . 363
sip-signaling-chat
configuration option . . . 364
sip-tls-cert
configuration option . . . 319
sip-tls-cert-key
configuration option . . . 319
sip-tls-cipher-list
configuration option . . . 320
sip-tls-crl
configuration option . . . 321
sip-tls-mutual
configuration option . . . 322
sip-tls-target-name-check
configuration option . . . 322
sip-tls-trusted-ca
configuration option . . . 322
sip-treatments-continuous
configuration option . . . 322
sml section
common options . . . 466–467
spool
common log option . . . 455
square brackets . . . 498
stand-alone mode . . . 27
standard
common log option . . . 457
stranded-call-redirection-limit (application level)
configuration option . . . 323
stranded-calls-overflow
configuration option . . . 323
stranded-calls-overflow (switch/DN level)
configuration option . . . 364
stranded-on-arrival-calls-overflow (application level)
configuration option . . . 323
stranded-on-arrival-calls-overflow (switch/DN level)
configuration option . . . 364
Stream Manager
appropriate codec format . . . 81
audio files . . . 81
defined . . . 80
Multipoint Conference Unit . . . 80
Music-on-Hold (MoH) server . . . 80
subscribe-presence
configuration option . . . 367
subscribe-presence-domain
configuration option . . . 366
subscribe-presence-expire
configuration option . . . 366
subscribe-presence-from
configuration option . . . 366
subscription-delay
configuration option . . . 324
subscription-id
configuration option . . . 365
subscription-timeout
configuration option . . . 324
Supported Functionality table . . . 256–263
Switch objects . . . 45
multi-site operation . . . 435
switch partitioning
defined . . . 430
T-Server support . . . 431
Switching Office objects . . . 44
multi-site operation . . . 436, 437, 438, 442
Switch-Specific Configuration . . . 59
sync-emu-agent
configuration option . . . 370
sync-reconnect-tout
configuration option . . . 491

T

Target ISCC
Access Code configuration . . . 439
Default Access Code configuration . . . 438
tcs-queue
configuration option . . . 485
tcs-use
configuration option . . . 485
third-party
known limitations . . . 275
Music-on-Hold server architecture . . . 29
third-party call control transfer . . . 120
time_convert
common log option . . . 453
time_format
common log option . . . 453
timed-acw-in-idle
configuration options . . . 324
emulated agents . . . 324
timeout
configuration option . . . 396, 484
timeout value format
configuration option . . . 493
TInitiateConference . . . 394
TInitiateTransfer . . . 394
tlib-map-replace-dn

configuration options . . . 325
T-Library Functionality . . . 255
TLS-related options
sip-port-tls . . . 317
sip-tls-cert . . . 319
sip-tls-cert-key . . . 319
sip-tls-cipher-list . . . 320
sip-tls-crl . . . 321
sip-tls-mutual . . . 322
sip-tls-target-name-check . . . 322
sip-tls-trusted-ca . . . 322
TMakeCall . . . 394
TMuteTransfer . . . 394
trace
common log option . . . 457
transaction types (ISCC) . . . 395, 400
supported . . . 409
transfer connect service . . . 412
transfer-complete-by-refer
configuration option . . . 367
Translation Rules section
configuration option . . . 489
treatments
Instant Messaging . . . 174
TRouteCall . . . 394
keys . . . 267
trunk lines . . . 407
trunk versus trunk group, differences . . . 61
T-Server
configuring Application objects . . . 46
for multi-sites . . . 435
multi-site operation . . . 435
starting . . . 56
using Configuration Manager . . . 46
TServer Section
configuration options . . . 277
T-Server section
configuration options . . . 470–475
TSingleStepTransfer . . . 394
TXRouteType . . . 395
type styles
conventions . . . 497
italic . . . 497
monospace . . . 498
typical SIP Server network . . . 26
typographical styles . . . 497

U

UDP
SIP message size limit . . . 335
SIP signaling port . . . 317
transport selection . . . 335
UNIX
installing T-Server . . . 42, 47
starting applications . . . 55
starting T-Server . . . 56
starting with run.sh . . . 54
unknown-bsns-calls
configuration option . . . 325
unknown-xfer-merge-udata
configuration option . . . 370
untimed-wrap-up-value
configuration option . . . 325
emulated agent options . . . 325
use-contact-as-dn
configuration option . . . 367
use-data-from
configuration option . . . 481
use-display-name
configuration option . . . 368, 370
use-implicit-access-numbers
configuration option . . . 485
user data propagation . . . 428
userdata-<n>
configuration option . . . 328
user-data-im-enabled
configuration option . . . 325
user-data-limit
configuration option . . . 475
userdata-map-filter
configuration option . . . 368
userdata-map-trans-prefix
configuration option . . . 326
use-register-for-service-state
configuration option . . . 368

V

V
command line parameters . . . 52
VDN . . . 407
verbose
common log option . . . 450
version numbering, document . . . 497
Via header
geo-location labeling . . . 249
Video
feature configuration . . . 244
Push Video . . . 242
start . . . 242
stop . . . 243
voice quality . . . 38

W

warm standby . . . 30, 384
Windows
installing T-Server . . . 43, 49
starting applications . . . 55

starting T-Server 56
starting with run.bat 55
wrap-up-threshold
configuration option 370
wrap-up-time
configuration option 326

X

x-conn-debug-all
common log option 462
x-conn-debug-api
common log option 462
x-conn-debug-dns
common log option 462
x-conn-debug-open
common log option 460
x-conn-debug-security
common log option 462
x-conn-debug-select
common log option 461
x-conn-debug-timers
common log option 461
x-conn-debug-write
common log option 461