SIP Server Deployment Guide

SIP Server 8.1.1

5/8/2024

Table of Contents

Supplement to SIP Server Deployment Guide

This supplement provides descriptions of new features introduced in SIP Server 8.1.1 as part of the SIP Server Continuous Delivery project. For information on new or updated configuration options, see here.

The current documentation set for SIP Server can be found here.

8.1.1 Features Support

The following features are described in this supplement:

Released in Version	Feature Name/Article	Date Released
8.1.104.49	Re-establishing connection on TCP transport exception	June 28, 2022
8.1.104.34	Remote agent deletion from a conference (TDeleteFromConference)	November 23, 2021
8.1.103.95	HTTP Monitoring Interface is updated to include statistics for T-Library client connections of the following SIP Server threads statistics: Session Controller (the main T-Server thread), T-Controller, Interaction Proxy, and Smart Proxy	July 21, 2020
8.1.103.88	Masking/unmasking specific SIP headers contained in SIP Server logs	May 12, 2020
	Enabling office-based agents to work from home - sample configuration	March 27, 2020
8.1.103.80	Enhanced handling of XS requests	March 20, 2020
8.1.103.35	HTTP Monitoring Interface is updated to include Extended Services (XS) statistics	February 5, 2019
8.1.103.35	Treating incoming calls as inbound calls	February 5, 2019
8.1.103.25	HTTP Monitoring Interface is updated to include SIP Feature Server statistics	November 6, 2018
8.1.103.08	Secure SIP Signaling	May 15, 2018
	Migrating from Network SIP	December 13, 2017

Released in Version	Feature Name/Article	Date Released
	Server to SIP Server	
8.1.102.93	Remote agents with non-provisioned phone numbers	November 14, 2017
8.1.102.58	Enhanced procedure for upgrading SIP Server HA pair	February 21, 2017
8.1.102.51	Masking sensitive data in SIP messages	November 15, 2016
8.1.102.50	SRV address support in Contact and Record-Route headers	November 9, 2016
8.1.102.39	Metadata support for IVR recording	August 29, 2016
8.1.102.38	No-Answer Supervision: After Routing Timeout Action support in multi-site deployments	August 23, 2016
8.1.102.33	Overload Control: Logging Level	July 26, 2016
8.1.102.31	Customizing Music on Hold	July 8, 2016
8.1.102.29	Setting SIP INVITE timeout for individual DNs	June 28, 2016
8.1.102.28	HTTP Live Streaming	June 22, 2016
8.1.102.26	Recording an Agent Greeting	June 7, 2016
8.1.102.25	Controlling Early Media with a Routing Strategy	May 24, 2016
8.1.102.22	Dial Plan enhancements including support for SIP Feature Server Dial Plan	May 3, 2016
8.1.102.20	Muting/Unmuting a Party in Two-way Calls	April 26, 2016
8.1.102.13	HTTP Monitoring Interface	March 29, 2016
8.1.102.13	Improved presentation of multiple routing attempts in historical reporting	March 29, 2016
8.1.102.09	Enhanced Reporting for Early ISCC Transaction Completion	March 8, 2016
8.1.102.05	Modifying the From Header in SIP INVITE	February 8, 2016
8.1.102.02	Muting/Unmuting a Party in a Conference	January 19, 2016
8.1.102.01	No-Answer Supervision: After Routing Timeout Action	December 29, 2015
8.1.102.00	Caller Information Delivery Content for AT&T Trunks for GVP	December 18, 2015
8.1.101.97	Instant Messaging in Business Continuity deployments	December 8, 2015
8.1.101.95	DTMF Clamping in a Conference	November 24, 2015

Released in Version	Feature Name/Article	Date Released
	support in multi-site deployments	
8.1.101.87	Enable Customer-on-Hold Privacy	September 29, 2015
8.1.101.85	Providing Origination DN Name and Location in EventRinging	September 16, 2015
8.1.101.83	Sending Outgoing INVITEs with Multipart Body	September 9, 2015
8.1.101.78	Private Conversations During Conference	August 7, 2015
8.1.101.75	Alternate Routing for unresponsive URS/ORS	July 24, 2015
8.1.101.75	Find Me Follow Me	July 24, 2015
8.1.101.75	Shared Call Appearance support in Business Continuity deployments	July 24, 2015
8.1.101.74	Informing Agents of Supervisor Presence	July 8, 2015
8.1.101.68	DTMF Clamping in a Conference	June 8, 2015
8.1.101.66	Caller Information Delivery Content for AT&T Trunks for URS/ ORS	May 21, 2015
8.1.101.62	Geo-location for MSML-based services: Strict Matching enhancement	March 30, 2015
8.1.101.62	Geo-location support by GVP enhancement	March 30, 2015
8.1.101.61	Nailed-up connection on agent login or when an agent in Ready state	March 17, 2015
8.1.101.57	Shared Call Appearance	February 12, 2015
8.1.101.57	Deleting Party From Conference	February 12, 2015
8.1.101.57	VXML support for agent greeting in multi-site and Business Continuity deployments	February 12, 2015
8.1.101.56	Agent Login and State Update on SIP Phones	January 30, 2015
8.1.101.50	Disabling Media Before Greeting	November 11, 2014
8.1.101.49	Hunt Groups support in Business Continuity deployments	October 31, 2014
8.1.101.47	Geo-location for MSML-based services: Strict Matching	October 21, 2014
8.1.101.43	Keep Alive for TCP Connections	September 19, 2014
8.1.101.38	Switching Between Supervision Modes	August 6, 2014
8.1.101.29	VXML support for agent greeting	June 20, 2014

Released in Version	Feature Name/Article	Date Released
8.1.101.29	Nailed-up connections in Business Continuity deployments	June 20, 2014
8.1.101.27	Hunt Groups enhancement to support sequential ringing	June 4, 2014
8.1.101.20	IMS Integration: Routed calls as originating or terminating	April 21, 2014
8.1.101.15	DN Recording Override	February 21, 2014
8.1.101.15	Dial Plan Support for Overdial	February 21, 2014
8.1.101.10	Trunk Capacity Control	January 31, 2014
8.1.101.10	Video Blocking	January 31, 2014

Re-establishing connection on TCP transport exception

SIP Server is now enhanced to re-establish connection when a TCP transport exception occurs. The new enhancement allows SIP Server to overcome a temporary single transport failure of a reliable SIP transport.

New procedures

The following two procedures are now implemented for handling a reliable transport failure:

Handling of Reliable Transport Failure for Standalone Deployment

If SIP Server detects a TCP transport failure of a single existing reliable transport while sending a re-INVITE request, it re-sends a request as soon as the preconfigured timeout expires. It repeats this action until a response is received successfully to the request but not for longer than the timeout defined by the **sip-invite-timeout** option.

Handling of Reliable Transport Failure for Deployment with SIP Proxy

> **Important**
>
> Coming soon. Check SIP Proxy Release Notes for availability of this feature.

SIP Proxy, on detecting a TCP transport failure of a forwarded request terminates that request transaction (client transaction). It returns a 503 error with the header, `Error-Info`, and value, `transport-error`, to the original request (server transaction). SIP Server processes the 503 response by waiting for the time defined by the pre-configured timeout and then re-sends the INVITE. It repeats these actions until a response is sent successfully to the request. Actions are not repeated after the timeout defined by the sip-invite-timeout option expires.

Unimpacted existing procedures

The following existing procedures are not impacted by this improvement and will continue to work the same way.

Handling of Failure with More than One Reliable Transport

If a failure occurs when more then one reliable transport is present, SIP Server processes the failure by re-sending the request through another transport. This behavior is preserved.

Handling of UDP Failure

There is no change in handling of UDP failures.

SIP workflow

There are no changes for cases with new INVITEs and for cases with non-INVITE requests.

Note on backward compatibility

Prior to this improvement, while receiving a 503 response with the Retry-After header, SIP Server waited for the retry-after time to expire, then re-sends the INVITE request and repeated that process without a limitation. That was not an acceptable behavior. Now, SIP Server re-tries the INVITE only within the period defined by **sip-invite-timeout** option.

Configuration options

A new configuration option, **retry-after-reliable-transport-error**, is introduced to specify the number of milliseconds SIP Server will wait before re-sending a request after transport failure on INVITE.

retry-after-reliable-transport-error
Setting: **[TServer]** section, Application level or Device level
Default Value: `0`
Valid Values: `0` to `34000` (recommended - `500` to `5000`)
Changes Take Effect: Immediately

The option specifies the number of milliseconds that SIP Server will wait before re-sending a request after a transport failure on re-INVITE. If the option is set to `0` or not configured, SIP Server disconnects the SIP peer after a transport error.

> **Important**
> Values at the device level take precedence over the values at the application level.

The existing sip-invite-timeout option is extended to specify the number of seconds for which SIP Server tries to re-send the request after the first failed attempt is detected after a transport error or a

`503` response.

sip-invite-timeout
Setting: **[TServer]** section, Application level
Default Value: `0` (in effect, 32 seconds)
Valid Values: `0` to 34
Changes Take Effect: Immediately

> Important
>
> In an environment with SIP Proxy, it is expected that SIP Proxy reacts to the transport failure of a forwarded request with a `503` response to SIP Server. Such a response should have `Error-Info: transport-error` in the header.

HA considerations

In case a SIP Server HA switchover happens during the time period configured in the *retry-after-reliable-transport-error* option, SIP Server, after becoming the primary instance, terminates the SIP dialog in which the reliable transport error failed before the switchover. The corresponding T-Library party will be released in this case.

Limitations

This improvement is only for transport failure during re-INVITEs (failures which may lead to dropped calls). There is no change for other SIP transport failure cases such as:

- New INVITEs. Failed initial INVITEs normally do not break established calls. They are followed by corresponding T-Library errors and can be processed in a regular way.
- Processing of non-INVITE transactions such as, SIP NOTIFY with talk or hold events. Failure of these transactions does not lead to dropped calls and will be processed in a regular way.
- This improvement covers only cases with single transport. If a failure occurs when more then one reliable transport is present, as usual, SIP Server will continue to process the failure by re-sending the request through another transport.

Remote agent deletion from a conference (TDeleteFromConference)

Starting with version 8.1.104.34, SIP Server supports remote agent deletion from a conference (TDeleteFromConference) in the mixed SIP Server and T-Server for Cisco Unified Communications Manager (CUCM) multisite environment.

SIP Server supports the following scenarios:

- When remote agent deletion from a conference is invoked on the SIP Server while a party is terminated on the other SIP Server peer.
- When remote agent deletion from a conference is invoked on the SIP Server while a party is terminated on the T-Server for CUCM peer.
- When remote agent deletion from a conference is invoked on the T-Server for CUCM while a party is terminated on the SIP Server peer.

A remote TDeleteFromConference request is a TDeleteFromConference request with a special key-value pair **location**=`remote_tserver_location` in AttributeExtensions.

While processing the TDeleteFromConference request, SIP Server does not verify AttributeOtherDN or the **location** value. The remote server executes this request only if the **location** value matches the server location and OtherDN could be found on the call. If the remote server ignores the request, the initiating server responds with Error Code 57 (ErrorTimeout) after the timeout expires.

This feature does not depend on the LCT party (Call Participant Info) functionality.

The new value of hybrid is added to the existing **sip-remote-del-from-conf** configuration option. In multisite deployments, when this option is set to `true`, SIP Server processes a TDeleteFromConference request to remove a remote party (specified in OtherDN) from a conference.

With a value of `hybrid`, SIP Server processes a TDeleteFromConference request even if the remote server is not a SIP Server (for example, T-Server for CUCM). The remote deletion from a conference could be done on the SIP Server peer, on the T-Server for CUCM peer, and any other T-Server that supports the same remote deletion from a conference rules as T-Server for CUCM supports.

Enhanced handling of XS requests

Starting with version 8.1.103.80, SIP Server can handle different HTTP error responses from SIP Feature Server for Dial Plan extended service (XS) requests in an enhanced way to address connection instabilities and provide a quality response to the origination side.

SIP Server sends an XS request to one of the SIP Feature Server URLs, starts the timer configured by the **xs-post-timeout** option, and waits for a Feature Server response. When the timeout expires, SIP Server sends an XS request to an alternative Feature Server URL. If SIP Server receives an error response within the timer period, it sends an XS request to an alternative Feature Server URL. In both cases, SIP Server sends an XS request to an alternative Feature Server URL only once.

When a Feature Server URL becomes out of service, SIP Server does not send subsequent requests to it until the Feature Server URL becomes in service. The Feature Server URL remains out of service, if the number of failed heartbeat requests exceeds the configured threshold (set in the **xs-missed-heartbeat-threshold** option), and that URL will not be selected for request processing, until it responds with a 200 OK message for a heartbeat request.

This table summarizes SIP Server actions for handling certain error responses received from Feature Server.

Error Code	Description	Action
400 Bad Request	Invalid Request Format	SIP Server responds to a caller with the 503 message. It doesn't resend a request and doesn't mark the Feature Server URL as out of service.
404 Not Found	Invalid API	SIP Server responds to a caller with the 503 message. It doesn't resend a request and doesn't mark the Feature Server URL as out of service.
501 Not Implemented	Unsupported operation type	SIP Server responds to a caller with the 503 message. It doesn't resend a request and doesn't mark the Feature Server URL as out of service.
503 Service Unavailable	Feature Server is unable to provide a response	SIP Server resends a request with an alternative Feature Server URL and marks the Feature Server URL that responded with 503 as out of service.
Any error response or Request Timeout	Feature Server internal error or unable to process a request	SIP Server resends a request with an alternative Feature Server URL and marks the Feature Server URL that responded with 503 as out of service, and responds to a caller with 503 on receiving an error or a timeout for retry.

When none of the Feature Server URLs are available and, as a result, the Feature Server VOIP Service DN is placed out of service, SIP Server starts rejecting call requests with a 503 Service Unavailable message.

SIP Server running in primary mode switches over to backup mode if there is no active connection to any of the configured Feature Server URLs. If the **switchover-on-xs-oos** option is set to `true`, SIP Server reports the SERVICE_UNAVAILABLE status to LCA/SCS to switch over to backup mode instead of rejecting requests. This behavior ensures availability of the dial plan resolution in case of network instabilities.

SIP Server starts the switchover process after the timeout defined by the **time-before-switchover-on-xs-oos** option expires.

To control how long an XS request is considered active, use the **xs-request-timeout** option. If no response is received within this timeout, SIP Server rejects the request immediately with a 503 Service Unavailable message.

All the above features can be enabled by setting the **enable-enhanced-dialplan-handling** option to `true` in the Feature Server VOIP Service DN (**service-type**=`extended`).

Configuration Options

enable-enhanced-dialplan-handling

Setting: **[TServer]** section, the SIP Server Application (standalone SIP Server) or the VOIP Service DN with **service-type**=`sip-cluster-nodes`
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After restart

When set to `true`, enables enhanced handling of Dial Plan extended service (XS) requests by SIP Server. This includes:

- Handling of various error codes sent as responses from Feature Server.
- Resending XS requests once on recoverable error responses from Feature Server.
- Setting a timeout (**xs-request-timeout**) for each dial plan request.
- Setting a timeout specific to the heartbeat requests (**xs-heartbeat-timeout**).
- Marking the URL as out of service on heartbeat failures based on the threshold set by **xs-missed-heartbeat-threshold**.
- Rejecting any XS request or switching SIP Server to backup mode when no active Feature Server URLs are available (the **switchover-on-xs-oos** option).

xs-request-timeout

Setting: **[TServer]** section, the VOIP Service DN with **service-type**=`extended`
Default Value: 8

Valid Values: 4-32
Changes Take Effect: For the next XS dial plan request

Specifies the timeout, in seconds, that SIP Server waits for a Feature Server response on an XS request. The timeout starts when the XS request is added to the queue and stops when a response is received from Feature Server. When the timeout expires, SIP Server rejects the XS request with a corresponding error. The request timeout must be at least twice as long as the **xs-post-timeout** option value.

xs-post-timeout

Setting: **[TServer]** section, the VOIP Service DN with **service-type**=extended
Default Value: 4
Valid Values: 2-16
Changes Take Effect: For the next XS dial plan request

Specifies the timeout, in seconds, for an XS request in transit. The timeout starts when the XS request is sent out and stops when a response is received from Feature Server. When the timeout expires, SIP Server either resends the XS request to an alternative Feature Server URL or rejects with a corresponding error if the limit of retries (more than 1) has exceeded. The post timeout must not be more than half of the **xs-request-timeout** option value.

xs-heartbeat-timeout

Setting: **[TServer]** section, the VOIP Service DN with **service-type**=extended
Default Value: 5
Valid Values: 2-120
Changes Take Effect: For the next XS heartbeat request

Specifies the timeout, in seconds, for an XS heartbeat request. The timeout starts when an XS heartbeat request is posted to a Feature Server URL and stops when a response for a heartbeat is received from Feature Server. When the timeout expires, SIP Server counts the number of failures and marks the URL as out of service if the threshold specified by the **xs-missed-heartbeat-threshold** option is reached. The heartbeat timeout must be greater than the **xs-post-timeout** option value.

xs-missed-heartbeat-threshold

Setting: **[TServer]** section, the VOIP Service DN with **service-type**=extended
Default Value: 3
Valid Values: 1-10
Changes Take Effect: Immediately

Specifies the maximum number of failed heartbeat requests that SIP Server receives from a Feature Server URL, before marking the corresponding URL as out of service.

switchover-on-xs-oos

Setting: **[TServer]** section, the SIP Server Application (standalone SIP Server) or the VOIP Service DN with **service-type**=`sip-cluster-nodes`
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies the SIP Server action in case of losing connectivity with all Feature Server URLs. SIP Server marks a URL as out of service when the threshold of failed heartbeat requests set by the **xs-missed-heartbeat-threshold** option is reached. When set to `true` and all configured Feature Server URLs become out of service, SIP Server reports the SERVICE_UNAVAILABLE status to LCA/SCS to switch over to backup mode. When set to `false`, SIP Server responds with a 503 Service Unavailable message to all calls, until one of the Feature Server URLs becomes available.

time-before-switchover-on-xs-oos

Setting: **[TServer]** section, the SIP Server Application (standalone SIP Server) or the VOIP Service DN with **service-type**=`sip-cluster-nodes`
Default Value: `1`
Valid Values: `0-60`
Changes Take Effect: Immediately

Specifies the timeout, in seconds, that SIP Server waits before reporting the SERVICE_UNAVAILABLE status in a scenario described in the **switchover-on-xs-oos** option. When set to `0` (zero), SIP Server reports the SERVICE_UNAVAILABLE status immediately after the Feature Server VOIP Service DN is detected as out of service.

xs-pool-size

Setting: **[TServer]** section, the SIP Server Application (standalone SIP Server) or the VOIP Service DN with **service-type**=`extended`
Default Value: `10`
Valid Values: Any number of connections that is possible for the system
Changes Take Effect: For the next XS request

Specifies the maximum number of connections to one Feature Server URL. The setting at a DN level takes priority.

xs-heartbeat-interval

Setting: **[TServer]** section, the SIP Server Application (standalone SIP Server) or the VOIP Service DN with **service-type**=`extended`
Default Value: `10`
Valid Values: `0-65535`
Changes Take Effect: For the next XS request

Specifies the heartbeat messages interval, in seconds. Value of `0` (zero) disables heartbeats. The setting at a DN level takes priority.

Feature Limitations

- SIP Server rejects the Dial Plan XS requests with a 503 Service Unavailable message instead of a 603 Decline message, when:
    - A retry limit for a request is exceeded.
    - None of the Feature Server URLs are available to provide a service.
- This feature depends on support from a specific version of SIP Feature Server. Consult corresponding documentation for the availability of this new feature in that component.

Enabling office-based agents to work from home

Important

The purpose of this article is to provide recommendations to our customers for moving office-based agents to their remote home-based locations.

Sample configuration of office-based agents

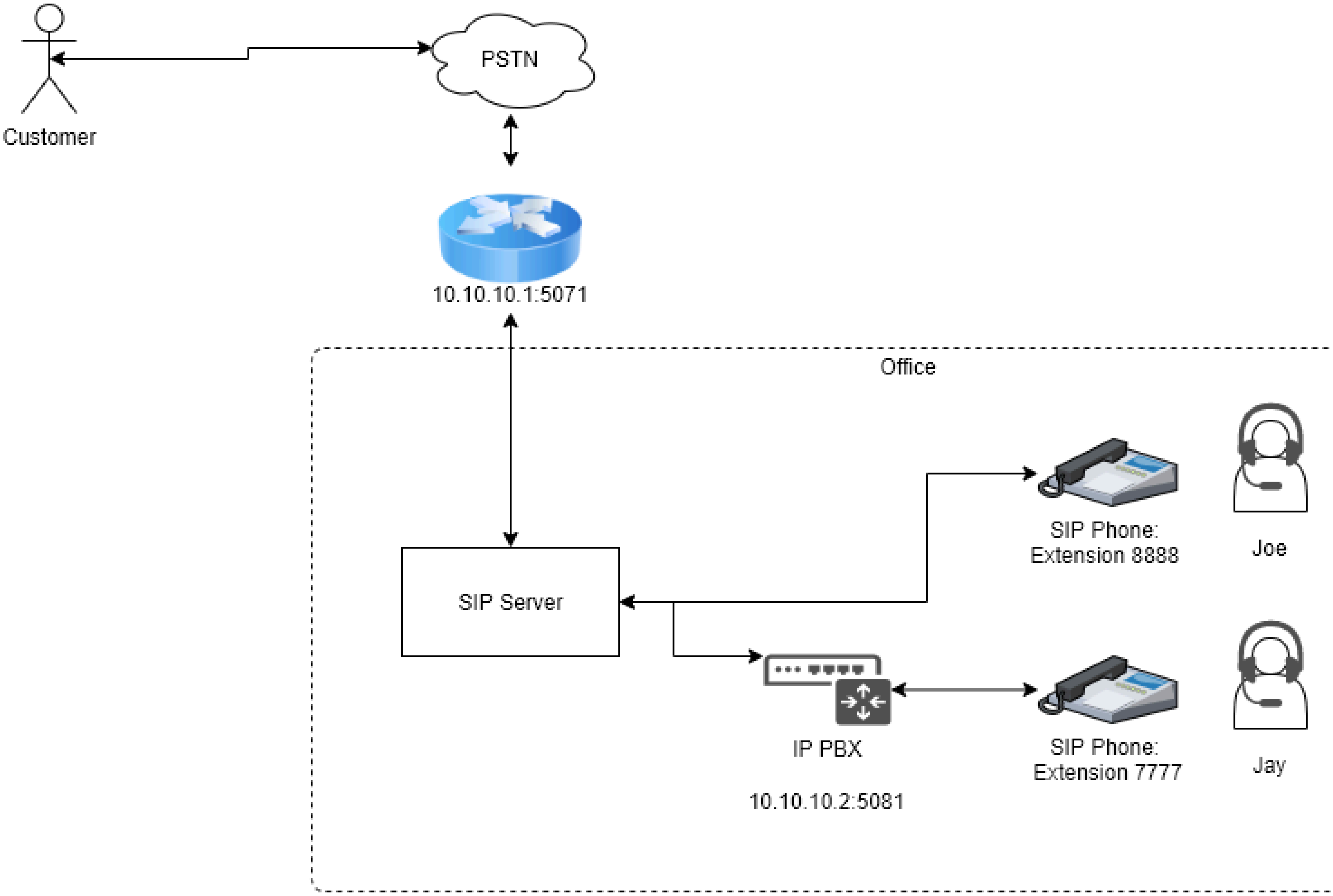


Figure 1: Reference configuration of office-based agents (click to expand)

In the sample configuration of office-based agents:

- Agent Joe has a SIP Phone (Extension 8888) that is directly registered on the SIP Server.
- Agent Jay has a SIP Phone (Extension 7777) that is located behind the office IP PBX.

Extension 8888 is a SIP Phone that is SIP-registered on the SIP Server and must have the following configuration of the **contact** option (other options are excluded):

Agent: Joe **Extension DN: 8888**
[TServer] contact = *

Extension 7777 is a SIP Phone that is located behind the softswitch (IP PBX) and must have the following configuration objects representing the softswitch and SIP Phone:

VOIP Service DN: Softswitch **(some reference configuration)**	**Agent: Jay** **Extension DN: 7777**

[TServer]
contact = 10.10.10.2:5081
service-type = softswitch
prefix = 77
dual-dialog-enabled = false
make-call-rfc3725-flow = 1
oos-check = 10
oos-force = 2
	[TServer]
	<there must be no "contact" configured for the DN>

Sample configuration of home-based agents

There are several possibilities for moving office-based agents to their home locations:

- Remote agents located behind the softswitch
- Remote agents with nailed-up connections located behind the softswitch
- Remote agents with non-provisioned phone numbers

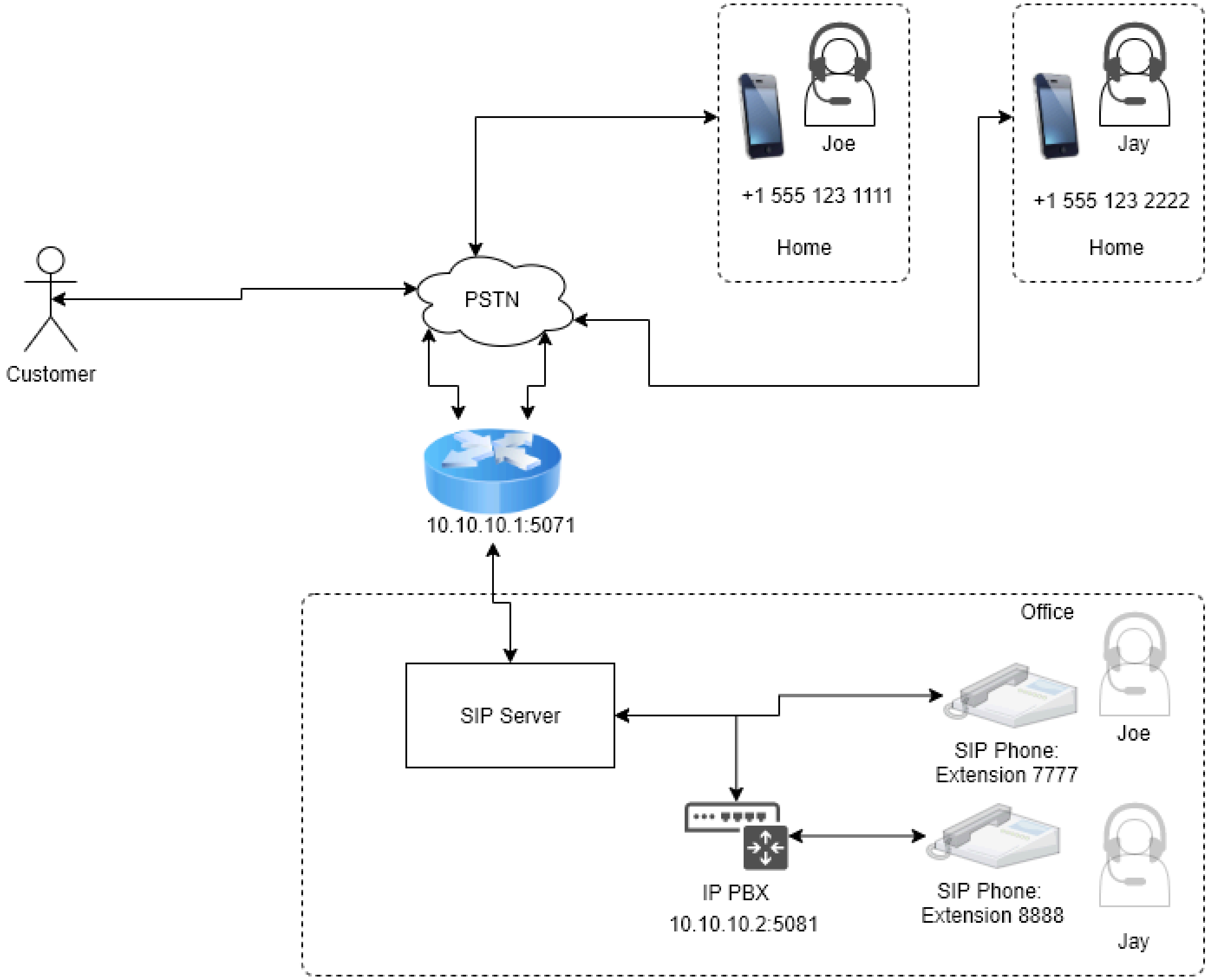


Figure 2: Reference configuration of home-based agents (click to expand)

Remote agents behind the softswitch

In this sample configuration:

- Agent Joe has a phone +1 555 123 1111 that is reachable over the PSTN.
- Agent Jay has a phone +1 555 123 2222 that is reachable over the PSTN.

To reconfigure agents to their home locations:

1. Configure the new or existing softswitch representing a trunk to the SBC: 10.10.10.1.5071
2. Configure the new Extension for Joe: +1 555 123 1111

3. Configure the new Extension for Jay: +1 555 123 2222

VOIP Service DN: Softswitch (some reference configuration)	**Agent: Joe** **Extension: +1 555 123 1111**	**Agent: Jay** **Extension: +1 555 123 2222**

[TServer]
contact = 10.10.10.1:5071
service-type = softswitch
prefix = +1
dual-dialog-enabled = false
make-call-rfc3725-flow = 1
oos-check = 10
oos-force = 2
	[TServer]
	<there must be no "contact" configured for the DN>
		[TServer]
		<there must be no "contact" configured for the DN>

When a call is routed to Joe's new extension +1 555 123 1111, SIP Server locates the Softswitch configuration by prefix +1 and sends an INVITE message through the SBC 10.10.10.1.5071 to DN +1 555 123 1111.

Remote agents with nailed-up connections behind the softswitch

To configure remote agents with nailed-up connections behind the softswitch, use the same configuration procedure as described in the Remote agents behind the softswitch section. In addition, the agent Extensions are configured with option **line-type** = 1.

Note: Configuration with **line-type** = 1 on the Extension behind the softswitch was introduced in SIP Server version 8.1.102.93. If you run a SIP Server version prior to 8.1.102.93, use the workaround solution for configuring the Extension DN with the **contact** pointing to the SBC (see Remote agents with nailed-up connections behind a trunk prior to SIP Server version 8.1.102.93).

VOIP Service DN: Softswitch (some reference configuration)	**Agent: Joe** **Extension: +1 555 123 1111**	**Agent: Jay** **Extension: +1 555 123 2222**

[TServer]
contact = 10.10.10.1:5071
service-type = softswitch
prefix = +1
dual-dialog-enabled = false
make-call-rfc3725-flow = 1
oos-check = 10
oos-force = 2
	[TServer]
	line-type = 1
	<there must be no "contact" configured for the DN>
		[TServer]
		line-type = 1
		<there must be no "contact" configured for the DN>

Remote agents with nailed-up connections behind a trunk prior to SIP Server version 8.1.102.93

Agent: Joe **Extension: +1 555 123 1111**	**Agent: Jay** **Extension: +1 555 123 2222**
[TServer]	[TServer]

contact = 10.10.10.1:5071
line-type = 1
reject-call-notready = true
dual-dialog-enabled = false
make-call-rfc3725-flow = 1
	contact = 10.10.10.1:5071
	line-type = 1
	reject-call-notready = true
	dual-dialog-enabled = false
	make-call-rfc3725-flow = 1

See detailed steps describing configuration and functionality of nailed-up connections in Nailed-Up Connections for Agents.

Remote agents with non-provisioned phone numbers

For Agent Jay with the DN configured behind the softswitch, a configuration described in this section could be used only if a deployment meets the following prerequisites:

- Agent desktop is Workspace Web Edition (WWE) version 8.5.201.95 or later
- SIP Server version 8.1.102.93 or later

Such configuration does ***not*** require provisioning of a new Extension DN +1 555 123 2222 for Agent Jay. Agent Jay continues to use Extension DN 8888 through the WWE desktop; however, voice calls are directed to the PSTN number +1 555 123 2222.

A special configuration must be enabled in WWE, which enables WWE to prompt an agent for the remote DN during login. The entered remote DN, +1 555 123 2222 in this case, is passed to the SIP Server in the **agent-phone** key-value pair of AttributeExtensions of RequestAgentLogin.

When an inbound call is routed to an agent logged in on DN 8888, SIP Server uses the provided remote DN to reach that agent.

For SIP Server to be able to reach an agent at the remote DN +1 555 123 2222, the softswitch VOIP Service DN must be created with the **contact** pointing to the SBC gateway.

VOIP Service DN: Softswitch (some reference configuration)

[TServer]
contact = 10.10.10.2:5081
service-type = softswitch
prefix = +1
dual-dialog-enabled = false
make-call-rfc3725-flow = 1
oos-check = 10
oos-force = 2

SIP Server resolves the softswitch by the prefix (in this example, +1) and sends INVITE to +1 555 123 2222 towards the SBC 10.10.10.2:5081.

For more information about this feature, see Remote agents with non-provisioned phone numbers.

System performance warning

For large-scale systems ranging 2000+ agents per SIP Server switch, bulk Extension DN re-configuration can adversely impact performance for Configuration Server and SIP Server. Bulk configuration changes generate high rate of configuration update notifications and associated load on the components processing configuration changes.

Observe the following guidelines should to prevent significant service degradation:

- Plan to do large-scale configuration changes during off-peak hours, when production traffic is the lowest.
- Monitor CPU consumption of Configuration Server, SIP Server, and other components deemed to be critical during the implementation.
- Throttle configuration changes using a staggered approach, dividing changes to the smaller batches of acceptable size.
- Start with a small conservative batch, for example 50-100 DNs, and observe increased CPU load.
- Validate successful reconfiguration of the initial batch.
- Estimate excess system capacity, and increase the batch size based on the estimated excess capacity.

Treating incoming calls as inbound calls

Starting with version 8.1.103.35, SIP Server can treat incoming calls from external callers (agents behind SIP trunks) as inbound calls. Previously, if the username in the **From** header of the initial INVITE request matched the name of the existing DN in the SIP Switch, SIP Server treated that inbound call as an internal call.

Feature Configuration

To enable this feature:

1. Set the **enforce-1pcc-inbound** option to `true`.
2. (Optional) Set the **internal-call-domains** option to a list of IPv4 CIDR blocks or FQDN separated by semicolons (;).

SIP Server feature processing logic

To take advantage of this feature and, if you use the **enforce-external-domains** option in your environment, Genesys recommends that you gradually transition from using the **enforce-external-domains** option to using the **enforce-1pcc-inbound** option.

The **enforce-external-domains** option has higher priority than the **enforce-1pcc-inbound** option. If configuration options of both approaches are applied, SIP Server verifies the new incoming call INVITE message multiple times, as follows:

1. SIP Server verifies the domain part of the **From** header of the INVITE message against the value of the **enforce-external-domains** option:
   - If a match is found in the **enforce-external-domains** option, SIP Server treats the call as inbound.
   - If a match is *not* found, SIP Server proceeds to Step 2.
2. SIP Server verifies the value of the **enforce-1pcc-inbound** option:
   - If the value of the **enforce-1pcc-inbound** option is set to `true`, SIP Server proceeds to Step 3.
   - Otherwise, SIP Server proceeds to Step 5.
3. SIP Server verifies the value of the **internal-call-domains** option:
   - If the value of the **internal-call-domains** option is empty, SIP Server treats the call as inbound.
   - If the value of the **internal-call-domains** option is not empty, SIP Server proceeds to Step 4.
4. SIP Server verifies the **Via** header of the INVITE message against the value of the **internal-call-domains** option:
   - If a match is found in the **internal-call-domains** option, SIP Server proceeds to Step 5.
   - If a match is *not* found in the **internal-call-domains** option, SIP Server treats the call as inbound.

5. SIP Server verifies only the username part in the **From** header in the INVITE message against the internal DNs:
   - If the username matches an Extension or ACD Position DN, SIP Server treats the call as internal.
   - If the username matches a Routing Point or Trunk Group DN, SIP Server rejects the call.
   - If a match is *not* found, SIP Server treats the call as inbound.

Configuration Options

enforce-1pcc-inbound

Setting: **TServer** section, Application level
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

When set to `true`, SIP Server treats 1pcc/incoming calls from external callers as inbound calls. A call is considered internal if both conditions are met:

1. A username in the **From** header matches the Extension DN configured in the SIP Switch.
2. A network address of the caller in the first **Via** header matches the IPv4 CIDR blocks or FQDN listed in the **internal-call-domains** option.

If the **internal-call-domains** option is empty, all incoming calls are treated as inbound calls.

internal-call-domains

Setting: **TServer** section, Application level
Default Value: An empty string
Valid Values: A list of IPv4 CIDR blocks or FQDN separated by semicolons (;). The IP address without a wildcard means the host address—for example, "1.2.3.0" means "1.2.3.0/32".
Changes Take Effect: On the next call

If the **enforce-1pcc-inbound** option is set to `true` and the **internal-call-domains** option is set to a list of IP addresses, SIP Server does the following for the incoming calls:

1. SIP Server verifies the **Via** header of the INVITE message against the value of the **internal-call-domains** option:
   - If a match is found, SIP Server proceeds to Step 2.
   - If a match is *not* found, SIP Server treats the call as inbound.
2. SIP Server verifies only the username part in the **From** header in the INVITE message against the internal DNs and classifies the calls as follows:
   - If the username matches an Extension or ACD Position DN, SIP Server treats the call as internal.
   - If the username matches a Routing Point or Trunk Group DN, SIP Server rejects the call.
   - If a match is *not* found, SIP Server treats the call as inbound.

All other 1pcc/incoming calls are treated as inbound calls. If the option is empty, all 1pcc/incoming calls are treated as inbound calls.

Feature Limitations

- IPv6 addresses are not supported in the list of the **internal-call-domains** option.

Secure SIP Signaling

Starting with version 8.1.103.08, SIP Server supports the secure SIP signaling schema, or **sips**, in accordance with RFC 5630.

When enabled, SIP Server forms the **Request-URI**, **From**, **To**, and **Contact** headers to include the **sips** schema when sending a SIP message to a device that requires that **sips** schema. The **Via** header of the message contains the transport TLS. When generating a response to an incoming message containing the **sips** schema, SIP Server forms the header **Contact** to include **sips**.

If the Request-URI with the **sips** schema also contains the transport parameter **transport=tcp** or **transport=tls**, communication will be established in secure TLS over TCP.

SIP Server applies the **sips** schema rules selectively, on a per call leg basis. In other words, if one SIP peer must communicate using secure SIP signaling while the other SIP peer does not support it, SIP Server is able to interconnect these peers using their supported protocol. However, devices communicating with SIP Server using the **sips** schema must be configured to enforce the **sips** schema.

Examples

Example of the INVITE message with the **sips** schema arrived to SIP Server:

```
INVITE sips:5000@172.21.83.50:5314;transport=TCP SIP/2.0
From: "7789"<sips:7789@172.21.83.24>;tag=74cc50-185315ac-13c4-55013-38-2147ec74-38
To: <sips:5000@172.21.83.50:5314>
Call-ID: 75b148-185315ac-13c4-55013-38-4004bd76-38
CSeq: 1 INVITE
Via: SIP/2.0/TLS 172.21.83.24:5061;branch=z9hG4bK-38-dd24-c4644b6
Max-Forwards: 70
Supported: replaces,100rel,eventlist,timer
Allow: REGISTER, INVITE, ACK, BYE, REFER, NOTIFY, CANCEL, INFO, OPTIONS, PRACK, SUBSCRIBE,
UPDATE, PUBLISH
User-Agent: AUDC-IPPhone/2.2.12.172 (420HD-Rev1; 00908F567540)
Contact: <sips:7789@172.21.83.24:5061;transport=TCP>
Session-Expires: 1800
Min-SE: 90
Content-Type: application/sdp
Content-Length: 299
...
```

Example of the 200 OK SIP Server response with the **sips** schema:

```
SIP/2.0 200 OK
From: "7789"<sips:7789@172.21.83.24>;tag=74cc50-185315ac-13c4-55013-38-2147ec74-38
To: <sips:5000@172.21.83.50:5314>;tag=EBDFD947-8988-4831-9FFF-051C3B626FFA-2
Call-ID: 75b148-185315ac-13c4-55013-38-4004bd76-38
CSeq: 1 INVITE
Via: SIP/2.0/TLS 172.21.83.24:5061;branch=z9hG4bK-38-dd24-c4644b6;received=172.21.83.24
Contact: <sips:5000@172.21.83.50:5314;transport=TCP>
X-Genesys-CallUUID: 8AH5H0H7054R93EBKC9ICTN8A8000001
Allow: INVITE, ACK, PRACK, CANCEL, BYE, REFER, INFO, MESSAGE, NOTIFY, OPTIONS
```

```
User-Agent: PolycomVVX-VVX_300-UA/5.2.0.8330
Allow-Events: conference,talk,hold
Accept-Language: en
Session-Expires: 1800;refresher=uas
Supported: uui,timer
Content-Type: application/sdp
Content-Length: 193
 ...
```

Feature Configuration

To enable the **sips** schema for secure SIP signaling, add the **sips** parameter to the **contact** option of the required device, as follows:

- **contact**=`sips:[number@]hostport[;transport={tls/tcp}]`

Genesys recommends that you configure **transport=tls**.

The **sips** schema is supported on the following types of DNs:

- Trunk
- Extension
- ACD Position
- Voice over IP Service with **service-type**=`softswitch`

Examples of the **contact** values with the **sips** schema:

- sips:fly.example.com;transport=tls
- sips:192.168.8.57;transport=tcp

Enforcing the sips schema by SIP registration

Self-registered DNs are configured with the option **contact**="*". When an incoming (from an endpoint) SIP REGISTER request contains the **sips** schema, SIP Server communicates with that endpoint using the **sips** schema. The **transport** parameter will be removed from the SIP REGISTER request.

Feature Limitations

- The **sips** schema is not yet supported by SIP Proxy.
- SIP Server guarantees consistency in using the **sips** schema only if it is configured and matches incoming traffic. In other words, the trunk through which an INVITE request containing **sips** arrives must have the **sips** schema configured and the self-registered DN must have the option **contact** ="*" configured.
- If required to communicate with Media Server over TLS, Genesys recommends using the **sip** schema

(not **sips** in the contact) to keep it backward compatible.

Migrating from Network SIP Server to SIP Server

If you have been using Network SIP Server (version 7.5 or earlier) for a while, you might consider migrating your environment to use 8.1.1 SIP Server that offers more capabilities. This article describes differences between the two components (Network SIP Server and SIP Server), and describes migration use cases, with migration steps, that might apply to your environment.

Network SIP Server was the first Genesys application supporting the Session Initiation Protocol (SIP). Network SIP Server is used for load balancing among premise SIP Servers. It utilizes simple functionality that can be summarized as follows:

1. Network SIP Server receives a SIP INVITE message from SIP media gateways.
2. Based on that INVITE, Network SIP Server generates EventRouteRequest containing T-Library attributes with elements of the INVITE message that Network SIP Server maps into that event.
3. Network SIP Server receives RequestRouteCall from a routing application.
4. Network SIP Server sends a 302 response to the INVITE message. The Contact of that request is the same as AttributeOtherDN of RequestRouteCall.
5. When it receives a SIP ACK message, Network SIP Server generates EventRouteUsed, which matches corresponding RequestRouteCall.

Feature differences between Network SIP Server and SIP Server

How Network SIP Server features are supported in SIP Server

	Functionality Of Network SIP Server	**Current Support in SIP Server**	**Comments**
1	Receiving the incoming INVITE and generating EventRouteRequest based on that	Supported with some difference in the interface	Attributes of the Event in SIP Server and Network SIP Server are mapped differently. All the information presented by Network SIP Server can be delivered by SIP Server; however some configuration and URS strategies might need be changed.
2	Responding to INVITE with 302	Supported with some difference in the interface	SIP Server cannot put the Contact value of AttributeOtherDN from

			TRouteCall into the 302 message as is. SIP Server instead applies matching rules to the value of OtherDN (evaluating internal DN or Trunk DN prefixes, or dial plans).
3	Option **load-balancing**=yes	Not supported, but there is a workaround.	Network SIP Server in this mode ignores the username of the URI in the incoming INVITE and distributes EventRouteRequest in a round-robin fashion among configured routing points. SIP Server does the opposite, always matching the username to available routing points. Some special configuration is required to enable sending any call to a default routing point in spite of the value of the username.
4	Option **load-balancing**=no	Supported	Network SIP Server (and SIP Server) distributes EventRouteRequest on the routing point that matches the username of the URI in the incoming INVITE.
5	Option **sip-port**	Supported	Value of **sip-port** is configurable for both applications.
6	Option **t1-timeout**	Not supported	Value of the SIP T1 timeout is hard-coded to `500` ms in the SIP library.
7	Option **t2-timeout**	Not supported	Value of the SIP T2 timeout is hard-coded to `4000` ms in the SIP library.
8	Option **router-timeout**	Supported	Supported by both applications.
9	Option **udp-packets-to-read**	Not supported	
10	Option **udp-packets-to-write**	Not supported	
11	Option **udp-recvbuf**	Not supported	Currently, SIP Server uses the default size of the buffer, which is 8 KB

			for Windows. Increasing this value up to 128 KB is known to improve performance for a highly loaded application. This can be done at the OS level.
12	Option **udp-sendbuf**	Not supported	SIP Server sets the size of the send buffer to 1 MB. The value is hard-coded.
13	Option **enable-sip-port-on-backup**	Not supported	Initially, Network SIP Server did not close the sip-port on backup assuming that the script performing an IP address takeover will take care the SIP traffic. Now, to support a more reliable switchover, the option was added for backward-compatibility. SIP Server always closes the sip-port on the backup server.
14	Mapping values of AttributeExtensions to 302	Not supported	Both applications use the same key SIP_HEADERS to define which key-value pairs to map into a SIP routing message. SIP Server supports it only for routing a call by INVITE or REFER (but not for a 302 response). SIP Server cannot map key-value pairs from the RequestRouteCall AttributeExtensions as additional headers of a 302 response.
15	Option **resolve-sip-address**	Supported	If set to `true`, Network SIP Server resolves the host name provided by TRouteCall into an IPv4 address. SIP Server does the same.
16	Option **default-routing-destinations**	Not supported	This option provides a list of default URIs to which the server redirects incoming calls (selecting them on a round-robin fashion) if URS is down.

17	Support of OPTIONS request	Supported	Supported the same way for both applications.
18	Option **final-timeout**	Not supported	Network SIP Server keeps a SIP call in memory within this timeout. SIP Server removes a call from memory right after its redirection or rejection. So, any incoming INVITE with the same SIP Call-ID will be treated as a new call.
19	Option **num-of-rerouting**	Not supported	SIP Server cannot limit re-routing attempts for the same SIP Call-ID.

Differences in mapping of INVITE elements into EventRouteRequest attributes

Attribute	**Network SIP Server**	**SIP Server**
AttributeOtherDN	Contact URI	From Username
AttributeDNIS	To URI	To Username
AttributeANI	From URI	From Username
AttributeExtensions	Controlled by the Extensions application-level section	Controlled by the INVITE application-level section
AttributeUserData	Not supported	Controlled by the INVITE application-level section

Network SIP Server maps basic data from INVITE headers to EventRouteRequest attributes as full URIs. SIP Servers maps the same data as usernames. It is unlikely that a URS strategy relies on attributes that are presented by Network SIP Server. However, Genesys recommends reviewing URS strategies during migration. If a strategy expects a URI, configure the INVITE section of the SIP Server application.

In addition to basic attributes, INVITE elements can be configured to be mapped as key-value pairs into AttributeExtensions. Configurations are different but the result is similar. In Network SIP Server, the **Extensions** application-level section is used. In SIP Server, the **INVITE** application-level section is used. SIP Server can distribute everything that Network SIP Server can. SIP Server can also distribute INVITE elements as UserData key-value pairs.

Differences in mapping RequestRouteCall attributes

Attribute	**Network SIP Server**	**SIP Server**
AttributeOtherDN	Contact URI is mapped as is in a 302 message	Target DN that is resolved as the Contact in a 302 message by configuration or a dial plan.

AttributeExtensions (key SIP_HEADERS)	Any key-value pair of AttributeExtensions can be mapped in a custom header of the 302 message	Not supported for a 302 message. Supported only for INVITE and REFER messages.

Migration Use Cases

Two general migration use cases are covered in this section:

- Network SIP Server routes calls to SIP Server through ISCC (route type =`route`) with load balancing off
- Network SIP Server with load balancing on

Network SIP Server routes calls to SIP Server through ISCC (route type =`route`) with load balancing off

When load balancing is turned off, Network SIP Server matches the username of the incoming INVITE Request-URI to an available routing point and distributes EventRoutePoint only if there is a successful match. SIP Server does exactly the same.

Here is an example of Network SIP Server actions before the migration:

1. A call is routed from a routing point of Network SIP Server to the routing point on a premise SIP Server using the External Routing Point on the premise server as an intermediate target.
2. URS sends RequestRouteCall with AttributeLocation of the premise destination switch.
3. AttributeOtherDN of that request is formed as a regular DN (routing point on the premise SIP Server).
4. To provide the origination server with the target in the form acceptable by Network SIP Server, External Routing Points on premise SIP Servers are configured with the Use Override property. The value of this property consists of the entire SIP URI with the Username equal to the name of the External Routing Point and the hostport equal to the SIP address of the premise SIP Server.

Migration includes the following steps:

1. Creating new Application objects for SIP Server
2. Connecting new applications with their peer applications
3. Modifying Network Switch Access Code
4. Configuring Trunk DN objects for inbound gateways
5. Configuring Trunk DN objects for destination trunks
6. Modifying External Routing Point DN objects on destination switches

Creating new Applications objects for SIP Server

Create two new Application objects (for high availability) using the SIP Server application template.

Most of the default application parameters will work correctly for SIP Server running in redirect mode. Ensure the following options are configured:

- **sip-link-type**=`0`
- **ringing-on-route-point**=`false`
- **sip-address**= <IP address or host name of the host where SIP Server will be running>
- **sip-port** = <port used to receive SIP messages>

Configure the new SIP Server applications to work in Warm Standby HA mode.

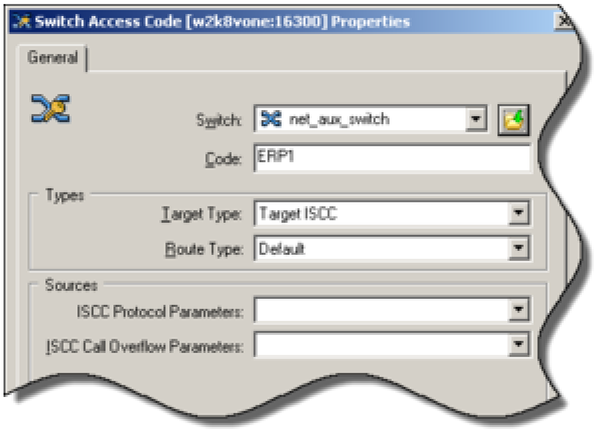
Use the existing Switch object to associate it with these new applications.

Connecting new applications with their peer applications

Connections of the new applications must replicate Network SIP Server existed connections. You might want to review existing connections to remove the obsolete ones.

Modifying Network Switch Access Code

In the **Access Codes** tab of the Network SIP Server Switch Properties window, configure a unique **Code** for each destination switch. In the example, the **Code** is set to ERP1.



Switch Access Code [w2k8vone:16300] Properties

Switch: net_aux_switch

Code: ERP1

Types

Target Type: Target ISCC

Route Type: Default

Sources

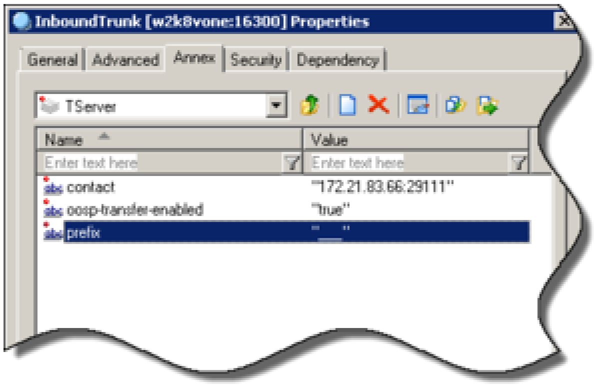
ISCC Protocol Parameters:

ISCC Call Overflow Parameters:

Configuring Trunk DN objects for inbound gateways

Configure a DN object of type Trunk for each inbound gateway in your environment, with the following configuration options for each Trunk DN:

- **contact**—Set to the IP address of the corresponding inbound gateway.
- **oosp-transfer-enabled**—Set to `true`.
- **prefix**—Set to a value different from any access codes configured in Modifying Network Switch Access Code. That will guarantee that this trunk will not be used for call routing to the destination.

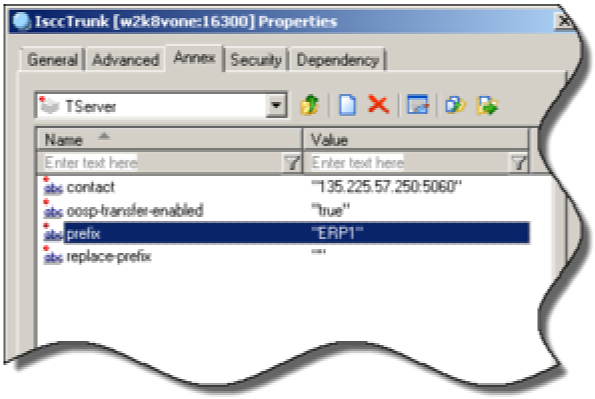


Name	Value
contact	"172.21.83.66:29111"
oosp-transfer-enabled	"true"
prefix	"_"

Configuring Trunk DN objects for destination trunks

Configure a DN object of type Trunk for each destination SIP Server in your environment, with the following configuration options for each Trunk DN:

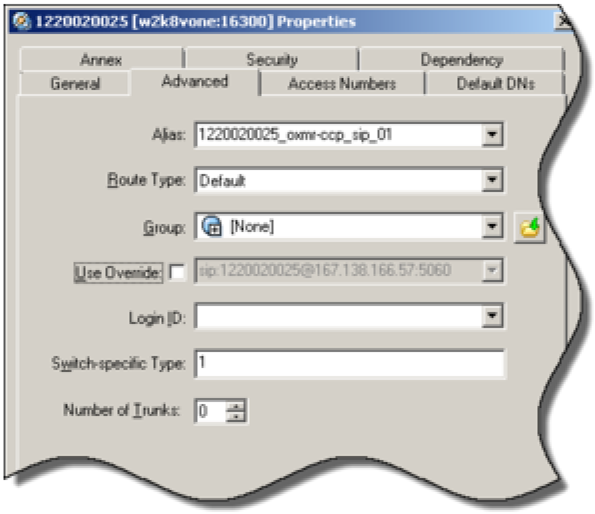
- **contact**—Set to the IP address of the corresponding destination SIP Server.
- **oosp-transfer-enabled**—Set to `true`.
- **prefix**—Set to the same value as the access code configured for the switch.
- **replace-prefix**—Set to an empty value. That will guarantee that an access code provided by ISCC will be used to find a proper SIP address of the destination SIP Server but will not be included into the URI username of a SIP 302 message.



Name	Value
contact	"135.225.57.250:5060"
oosp-transfer-enabled	"true"
prefix	"ERP1"
replace-prefix	""

Modifying External Routing Point DN objects on destination switches

Network SIP Server used the Use Override property to provide the entire URI as AttributeOtherDN of the ISCC RequestRouteCall. Selection of the Use Override property is no longer required. SIP Server resolves a DN into the URI through the Switch's Trunk DNs configuration.



Alias: 1220020025_oxmr-ccp_sip_01

Route Type: Default

Group: [None]

Use Override: □ sip:1220020025@167.138.166.57:5060

Login ID:

Switch-specific Type: 1

Number of Trunks: 0

Migration of deployments with load balancing

When Network SIP Server is used for load-balancing it means that:

- Network SIP Server does not attempt to match the username of the incoming INVITE Request-URI. It distributes EventRouteRequest to the available routing point.
- Routing points are selected in a round-robin fashion, so any consecutive event is distributed to a different DN.

To satisfy the first bullet point, the Alternate Routing for Calls to an External Destination feature must be enabled in SIP Server. The SIP Server switch must not have any internal resources that match possible usernames in the INVITE Request-URI and the From header. In that case, a call is qualified as an inbound call to be routed to an external destination. SIP Server sends the call to a routing point defined by the **default-route-point** option.

If an incoming call rate is too high for a single routing point to handle, you can create a fast strategy, which will provide a round-robin call distribution among additional configured routing points.

Performance Considerations

The SIP Server Sizing Tool is a spreadsheet providing for the input of sizing parameters to calculate CPU usage and network bandwidth of a SIP Server application. Use this tool to calculate CPU usage and network bandwidth of SIP Server that has replaced the former Network SIP Server.

The **Input & Calculation(Redirect)** tab of the Tool is dedicated to the sizing of SIP Server working as a SIP redirect application.

The Tool is located on this page: https://docs.genesys.com/Documentation/SIPS.

Examples of EventRouteRequest for Network SIP Server and SIP Server

```
INVITE sip:8001@172.21.83.50:5060 SIP/2.0
From: <sip:29111@172.21.83.50:29111>;tag=6CA770E8-0266-415F-A118-CEB42F39C605-1
To: sip:8001@172.21.83.50:5060
Call-ID: 082D1164-7D88-4863-8701-40FCA11340F8-1@172.21.83.50
CSeq: 1 INVITE
Content-Length: 145
Content-Type: application/sdp
Via: SIP/2.0/UDP 172.21.83.50:29111;branch=z9hG4bK57EEFBEE-F235-45FD-924A-C941A9D1942D-1
Contact: <sip:172.21.83.50:29111>

v=0
o=PhoneSimulator 1 1 IN IP4 172.21.83.50
s=incoming INVITE
c=IN IP4 172.21.83.50
t=0 0
m=audio 39111 RTP/AVP 0
a=rtpmap:0 PCMU/8000/1
```

Network SIP Server:

```
@16:54:59.9870 [0] 7.5.000.23 distribute_event: message EventRouteRequest
    AttributeEventSequenceNumber    0000000000000007
    AttributeTimeinuSecs    987000
    AttributeTimeinSecs    1504914899 (16:54:59)
    AttributeExtensions    [53] 00 01 00 00..
        'From:tag'    'E009AE79-BFB2-408B-AD4D-CF3F908FEA82-1'
    AttributeOtherDN    '172.21.83.50:29111'
    AttributeThisDNRole    2
    AttributeThisQueue    '8001'
    AttributeThisDN    '8001'
    AttributeCustomerID    'gregb'
    AttributeANI    '29111@172.21.83.50:29111'
    AttributeDNIS    '8001@172.21.83.50:5080'
    AttributeCallUUID    'KB7JRPUN5P1D12AK6V7SNCQJC4000001'
    AttributeConnID    04f402aacccd3001
    AttributeCallID    1
    AttributePropagatedCallType    2
    AttributeCallType    2
```

SIP Server:

```
@16:01:19.3540 [0] 8.1.102.XXX_Debug distribute_event: message EventRouteRequest
    AttributeEventSequenceNumber    0000000000000037
    AttributeTimeinuSecs    354000
    AttributeTimeinSecs    1505430079 (16:01:19)
    AttributeExtensions    [135] 00 03 00 00..
        'OtherTrunkName'    'trunk_29111'
        'From'    '<sip:29111@172.21.83.50:29111>;tag=6CA770E8-0266-415F-A118-CEB42F39C605-1'
        'BusinessCall'    1
    AttributeOtherDNRole    1
    AttributeOtherDN    '29111'
    AttributePartyUUID    'JTNRVQG16D2FBEV53LIVKD6AB0000003'
    AttributeThisQueue    '8001'
    AttributeThisDNRole    2
    AttributeThisDN    '8001'
    AttributeANI    '29111'
    AttributeDNIS    '8001'
    AttributeCallUUID    '1VJ3ICNE8D4HLBAJ7TRPDIMF84000001'
    AttributeConnID    006c02ab4a93f001
    AttributeCallID    16777217
    AttributePropagatedCallType    2
    AttributeCallType    2
    AttributeCallState    0
```

Remote Agents Support

SIP Server supports remote agents that use legacy PSTN phones. These agents could be working from their homes, or in a branch office that has simple PSTN connectivity.

SIP Server supports the following configurations for remote agents, depending on the remote agent locations:

- Remote agents located behind the softswitch
- Remote agents located behind the SBC/gateway
- Remote agents with non-provisioned phone numbers

To learn about benefits of nailed-up connections and how to configure them, refer to the Nailed-Up Connections for Agents topic.

To reconfigure office-based agents to their remote home-based locations, refer to the Enabling office-based agents to work from home topic.

For general information about configuring SIP devices, refer to the Configuring Devices and Services section in the *SIP Server Deployment Guide*.

Configuring remote agents located behind the softswitch

Remote agent location	VOIP Service DN: Softswitch configuration	Extension DNs configuration
Behind the softswitch	**[TServer]** • **contact** = <the contact URI that SIP Server uses for communication with the softswitch> • **prefix** = <the initial characters of the number that must match a particular softswitch for that softswitch to be selected> • **service-type** = `softswitch` • **refer-enabled** = `false` • **dual-dialog-enabled** = `false`	The Extension DN (Number property) for each remote agent must be configured with the PSTN number—for example, +1 555 123 1111, and contain no options. **[TServer]** <no options>

Remote agent location	VOIP Service DN: Softswitch configuration	Extension DNs configuration
	- **reject-call-notready** = `true` (recommended, not mandatory) - **sip-cti-control** = <ensure that this option is not configured>	
With nailed-up connections behind the softswitch	**[TServer]** - **contact** = <the contact URI that SIP Server uses for communication with the softswitch> - **prefix** = <the initial characters of the number that must match a particular softswitch for that softswitch to be selected> - **service-type** = `softswitch` - **refer-enabled** = `false` - **dual-dialog-enabled** = `false` - **reject-call-notready** = `true` (recommended, not mandatory) - **sip-cti-control** = <ensure that this option is not configured>	The Extension DN (Number property) for each remote agent must be configured with the PSTN number—for example, +1 555 123 1111, and contain the following options: **[TServer]** - **line-type** = 1

Configuring remote agents located behind the SBC/gateway

Remote agent location	Extension DNs configuration
Behind the SBC/gateway	The Extension DN (Number property) for each remote agent must be configured with the PSTN number—for example, +1 555 123 1111, and contain the following options: **[TServer]**

Remote agent location	Extension DNs configuration
	• **contact** = <the contact URI of the PSTN SBC/gateway, depending on the agent location> • **refer-enabled** = `false` • **dual-dialog-enabled** = `false` • **reject-call-notready** = `true` (recommended, not mandatory) • **sip-cti-control** = <ensure that this option is not configured>
With nailed-up connections behind the SBC/gateway	The Extension DN (Number property) for each remote agent must be configured with the PSTN number—for example, +1 555 123 1111, and contain the following options: **[TServer]** • **contact** = <the contact URI of the PSTN gateway/SBC, depending on the agent location> • **refer-enabled** = `false` • **dual-dialog-enabled** = `false` • **reject-call-notready** = `true` (recommended, not mandatory) • **sip-cti-control** = <ensure that this option is not configured> • **line-type** = 1

Limitations

Due to the specifics of gateway behavior in performing SIP REFER methods, support for remote agents has some limitations. In order to use remote agents, you must perform one of the two following steps:

- Provision customers and remote agents to use physically separate gateways (otherwise, calls from agents to customers take shortcuts within gateways, which means that SIP Server loses track of the call and therefore cannot perform call control). Even in this configuration, direct calls between two remote agents on the same gateway are not visible to SIP Server.
- 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.

Remote agents with non-provisioned phone numbers

Starting with version 8.1.102.93, SIP Server improves provisioning of remote agent DNs in the Configuration Database. It is no longer required to provision external phone numbers (for example, agent’s PSTN numbers) in the Configuration Database. You must create an Extension DN for each remote agent where a DN number can be a primary office DN number or any other number if an agent doesn't have a primary office DN.

The external phone number is used to reach the agent during the agent session only, thereby limiting the lifetime of the external phone number to a particular agent session. In other words, after the agent is logged out, any associations with that external phone number are removed.

The non-provisioned phone number to be used for the agent session is passed to SIP Server in the TAgentLogin request in AttributeExtensions as the **agent-phone** key. AttributeThisDN of that request will contain the agent DN configured in the Configuration Database.

This feature requires Workspace Web Edition (WWE) version 8.5.201.95 or later.

Configuring remote agents with non-provisioned phone numbers

Remote agent location	VOIP Service DN: Softswitch configuration	Extension DNs configuration
Behind the softswitch	**[TServer]** • **contact** = <the contact URI that SIP Server uses for communication with the softswitch> • **prefix** = <the initial characters of the number that must match a particular softswitch for that softswitch to be selected> • **service-type** = `softswitch` • **refer-enabled** = `false` • **dual-dialog-enabled** = `false` • **reject-call-notready** = `true` (recommended, not mandatory) • **sip-cti-control** = <ensure that this option is not configured>	**[TServer]** <no options>
With nailed-up connections behind the	**[TServer]**	**[TServer]**

Remote agent location	VOIP Service DN: Softswitch configuration	Extension DNs configuration
softswitch	• **contact** = <the contact URI that SIP Server uses for communication with the softswitch> • **prefix** = <the initial characters of the number that must match a particular softswitch for that softswitch to be selected> • **service-type** = `softswitch` • **refer-enabled** = `false` • **dual-dialog-enabled** = `false` • **reject-call-notready** = `true` (recommended, not mandatory) • **sip-cti-control** = <ensure that this option is not configured>	• **line-type** = 1 • **connect-nailedup-on-login** = `gcti::park`

Limitations

- If a non-provisioned phone number is used for the agent session, the agent can only initiate calls using the agent desktop. 1pcc calls originated from the non-provisioned phone number are not supported.
- For agents with nailed-up connections that use a non-provisioned number for the agent session, an establishment of the nailed-up connection by calling into a contact center routing point is not supported.
- Hunt Groups in Business Continuity (BC) functionality are not supported by this feature. That is, in the BC deployment, agent logging with a non-provisioned external phone number to a DN that is a member of the Hunt Group is not supported.

Masking sensitive data in SIP messages

Starting with version 8.1.102.51, SIP Server can mask sensitive data in SIP messages. When enabled, SIP Server replaces:

- All private SIP header values with a single asterisk
- SIP message body content with the phrase CONTENT FILTERED

SIP Server does not replace the content of type `application/sdp`, and it replaces `application/vnd.radisys.msml+xml` in the SIP message body only when it contains user data.

Starting with version 8.1.103.88, SIP Server can unmask specific SIP headers contained in SIP Server logs. This feature is enabled by the x-sip-unmask-headers and x-sip-unmask-headers-default configuration options.

Feature Configuration

To enable masking sensitive data in SIP messages, set the **x-sip-mask-sensitive-data** configuration option to `true` in the **[log]** section of the SIP Server Application.

Configuration Options

x-sip-mask-sensitive-data

Setting: **[log]** section, Application level
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether SIP Server masks sensitive data in SIP messages contained in SIP Server logs.

- If set to `true`, SIP Server masks all private SIP header values and SIP message body content of all types, except for `application/sdp` and `application/vnd.radisys.msml+xml`. If the message contains `application/vnd.radisys.msml+xml`, SIP Server masks it only when it contains user data.
- If set to `false`, SIP Server does not mask sensitive data in SIP messages contained in SIP Server logs.

x-sip-unmask-headers

Setting: **[log]** section, Application level
Default Value: No default value
Valid Values: A list of comma-separated SIP headers
Changes Take Effect: Immediately

Specifies a list of private SIP headers that SIP Server does not mask in SIP messages contained in SIP

Server logs. These headers are unmasked in addition to the headers specified in the **x-sip-unmask-headers-default** option. If the value of this option is not configured or empty, headers specified in the **x-sip-unmask-headers-default** are unmasked. Example: `X-Genesys-UUID,X-ISCC-Id`.

x-sip-unmask-headers-default

Setting: **[log]** section, Application level
Default Value: `X-Genesys-strict-location,X-Genesys-peer-proxy-contact,X-Genesys-CallUUID,X-Genesys-PartyInfo,X-Genesys-GVP-Session-ID,X-Genesys-CallInfo,X-Genesys-Route,X-Genesys-geo-location,X-Genesys-bypass-resource-list,X-ISCC-Id,X-ISCC-CofId,X-Detect,Event,presence,Answer-Mode`
Valid Values: A list of comma-separated SIP headers
Changes Take Effect: Immediately

Specifies a list of private SIP headers that SIP Server does not mask in SIP messages contained in SIP Server logs, by default. To unmask other SIP headers that are not included in the default value of this option, use the **x-sip-unmask-headers** option. If the value of this option is empty, the private SIP headers remain masked/unmasked based on the value of **x-sip-unmask-headers** and **x-sip-mask-sensitive-data**.

SRV address support in Contact and Record-Route headers

Starting with version 8.1.102.50, SIP Server supports the SRV FQDN—FQDN resolving to SRV records—received in the Contact or Record-Route headers of a SIP message. SIP Server also supports the SRV FQDN in the **contact** option on a Trunk DN.

If the target destination received in the URI of the Contact or Record-Route headers of a 200 OK message is not a numeric IP address, and no port is present, SIP Server performs an SRV query to obtain the target's IP address:port. The OPTIONS messages are sent over all transports representing SRV records. The ACK messages and all further SIP requests are sent to the active transport with the highest priority. SIP Server uses the original transport if it is among the active transports with the highest priority. If no active SRV records are found, the SIP transaction fails.

If the target destination received in the URI of the Contact header of an INVITE message is not a numeric IP address, and no port is present, SIP Server performs an SRV query to obtain the target's IP address:port. The OPTIONS messages are sent over all transports representing SRV records. Further SIP requests are sent to the active transport with the highest priority. SIP Server uses the original transport if it is among the active transports with the highest priority. If no active SRV records are found, SIP Server uses the transport of the original INVITE message for further SIP requests.

When SIP Server is deployed with SIP Proxy (the Application-level option **sip-outbound-proxy** is set to `true`) and it must send a SIP request to a destination configured with the SRV FQDN or list of active transports, SIP Server selects an active target destination and adds the private X-Genesys-Route header with a value of `sip:IpAddress:Port[;transport=tcp/tls]`. SIP Proxy uses the value of the X-Genesys-Route header as the next destination for forwarding the request. For SIP Proxy, this header has priority over the target specified in Request-URI or Route headers. SIP Server uses the same transport value for the X-Genesys-Route header until a transport becomes out of service.

Feature Configuration

To configure SIP Server to perform an SRV query:

- Set **sip-enable-gdns** to `true` (at an Application level).
- Set **sip-enable-rfc3263** to `true` (at an Application level).
- If SIP Proxy is used, set **sip-enable-x-genesys-route** to `true` (on an Application level).
- In a multisite SRV/DNS-based configuration:
    - Set **sip-address-srv** to the SRV FQDN.
    - Set **sip-address** to the hostname of the SIP Server interface.
    - Set **sip-port** to any valid port.
    - Set the **contact** option to the SRV FQDN on inter-site Trunk DNs.

Configuration Options

sip-enable-x-genesys-route

Setting: **TServer**, Application level
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies if SIP Server adds the private X-Genesys-Route header to SIP messages when deployed with SIP Proxy. This is for backward compatibility to disable new functionality in old deployments.

sip-disable-via-srv

Setting: **TServer**, Application level
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When set to `true`, SIP Server inserts a value of the **sip-address** option in the Via header. This option applies when the **sip-address-srv** option is configured.

contact

Setting: **TServer**, DN level
Default Value: No default value
Valid Values: Any alphanumeric string
Changes Take Effect: For the next call

Contains the contact URI in the following format:

```
[sip:][number@]srvFQDN[;transport={tcp/udp}]
```

Where:

- `sip`: is an optional prefix.
- `number` is the DN number. This is currently ignored.
- `srvFQDN` is the SRV FQDN.
- `transport=tcp` or `transport=udp` is used to select the network transport. The default value is `udp`.

SIP Private Header

Header name: **X-Genesys-Route**

Value: `sip:host_ip:port[;transport=tcp/tls]`

This header applies only to a configuration with SIP Proxy. SIP Server adds this header when sending a SIP request to a SIP device located behind SIP Proxy, for which an active out-of-service check is enabled (**oos-check** is set to `true`). SIP Proxy uses the header value as the next destination to which the request is forwarded.

Feature Limitations

SIP Server does not support the SRV FQDN in REGISTER messages.

Metadata Support for IVR Recording

SIP Server can now pass its Application name in the custom X-Genesys-sipsAppName header of the INVITE message to GVP/Media Server. This metadata—along with ANI, CallUUID, and DNIS—is used by GVP to get information about the call for a proper retrieval of recording files.

Feature Configuration

To enable this feature, set the **sip-enable-ivr-metadata** configuration option at the Application or DN level as required.

Note the following: If the IVR recording feature is enabled, then it is not required to explicitly enable the recording by setting the **record** option to `true` on DNs representing GVP, such as Trunk, Trunk Group, or Voice Treatment Port. Recording will be started by the VXML application running on the Media Server.

Configuration Options

sip-enable-ivr-metadata

Setting: **TServer** section, DN level
Default Value: No default value
Valid Values: `true, false`
Changes Take Effect: For the next call

Specifies whether SIP Server passes its Application name in the initial INVITE message (in the X-Genesys-sipsAppName header) to Media Server. If this option is set to `true`, SIP Server includes its Application name in the custom header of the INVITE that it sends to Media Server. If this option is set to `false`, SIP Server does not include its Application name in the initial INVITE sent to Media Server. This option applies to DNs of type Trunk, Voice over IP Service (msml), Trunk Group, and Voice Treatment Port. This DN-level setting takes priority over the Application-level setting.

sip-enable-ivr-metadata

Setting: **TServer** section, Application level
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: For the next call

Specifies whether SIP Server passes its Application name in the initial INVITE message (in the X-Genesys-sipsAppName header) to Media Server. If this option is set to `true`, SIP Server includes its Application name in the custom header of the INVITE that it sends to Media Server. It also enables the default behavior of the feature depending on the DN type, as follows:

- Voice over IP Service (msml), Trunk Group, and Voice Treatment Port—SIP Server sends the custom header.

- Trunk—SIP Server does not send the custom header.

If this option is set to `false`, SIP Server does not include its Application name in the initial INVITE sent to Media Server.

To enable this feature for a DN of type Trunk, set the **sip-enable-ivr-metadata** option to `true` on the corresponding Trunk DN.

Overload Control: Logging Level

This feature provides the ability to control SIP Server's CPU usage overload by decrementing the server's log level when the CPU usage overload threshold is reached. Overload is detected by per-thread CPU usage measurement. CPU usage is checked every 10 seconds. If the CPU usage of any core SIP Server thread exceeds the value specified in the **log-reduce-cpu-threshold** configuration option, the log level is decremented to allow SIP Server to handle traffic more efficiently. Once the load drops below 40% of the **log-reduce-cpu-threshold** configuration option setting, it remains at that level for approximately 300 seconds; after that the logging level is restored to the initially configured level.

When configuring the overload threshold, keep in mind the following:

- The threshold value must not be configured too high; otherwise, the reduced logging can bring a risk not being enabled at all.
- The threshold must not be configured too low; otherwise, the lack of logging will make troubleshooting impossible in case of any failure.

Genesys recommends monitoring the SIP Server usage during a typical load spike period, detecting both the start and finish of the period, so the overload threshold is set appropriately.

In HA deployments, the primary and backup SIP Servers monitor and process overload conditions independently. For example, the primary server might be overloaded, while the backup server is not.

log-reduce-cpu-threshold

Setting: **overload** section, Application level
Default Value: `0`
Valid Values: `0, 5-100`
Changes Take Effect: Immediately

Specifies the CPU usage overload threshold in percent. When the SIP Server CPU usage increases beyond the specified value, SIP Server is considered overloaded and the log level is decremented. The default value of `0` (zero) disables the dynamic overload control feature.

Customizing Music on Hold

Starting with version 8.1.102.31, SIP Server lets you customize music for music-on-hold treatments. When the music-on-hold feature is activated, it applies to scenarios when the hold action is performed by an agent within the duration of the call explicitly (by THoldCall), or implicitly (by TAlternateCall, TInitiateTransfer, or TInitiateConference).

New options are supported on the Switch object level:

- **music-on-hold** for Routing Point DNs
- **default-music** for Agent Logins

When custom music-on-hold is enabled on the Routing Point with the **music-on-hold** configuration option, or with the **music-on-hold** key in AttributeExtensions of TRouteCall, it remains attached (sticks) to the call until the call is released. If a TRouteCall request arrives with an empty value of the **music-on-hold** key in AttributeExtensions, the custom music-on-hold stickiness is removed from the call. If call routing fails, the custom music-on-hold setting is rolled back to the previous value.

The value of the **music-on-hold** option is attached to calls distributed via this Routing Point and used for playing the music-on-hold later.

When the **default-music** option is set for an Agent Login object, the setting applies only to a call established by the agent who activated the Hold operation.

Music File Priority

The following settings determine the order of priority—from highest to lowest—in which a music file is played for a call on hold:

1. The **music** key of AttributeExtensions in THoldCall, TAlternateCall, TInitiateTransfer, TInitiateConference requests, which initiate the Hold operation for a call.
2. The **music-on-hold** key of AttributeExtensions in TRouteCall (if there are several TRouteCall requests for this call containing this key, the value from the last one is applied).
3. The **music-on-hold** option on a Routing Point DN (if a call is passed through several Routing Points containing this option, the value from the last one is applied).
4. The **default-music** option on an Agent Login level.
5. The **default-music** option on an agent's Extension DN level.
6. The **default-music** option on a SIP Server Application level.

How it Works in Conferences

The custom music-on-hold setting is not applied to conferences and not shared when a consultation call is merged with the main call. However, the custom music-on-hold setting remains associated with the call, and if only two participants are left on the call, the custom music-on-hold setting will be applied if the caller is placed on hold. When a new party joins the conference, the custom music-on-hold setting is not applied.

For multi-site conferences support, SIP Servers must propagate full information about call parties. See "Providing Call Participant Info" in the SIP Server Deployment Guide for information on how to enable it.

How it Works in Transfers

The custom music-on-hold setting is transferred with the call, which includes call routing, single-step transfers, two-step transfers, and call forwarding. In multi-site transfers, the ISCC connection is used.

If a call is transferred through a Routing Point that has a custom music-on-hold setting, the new music-on-hold setting will be applied to the next Hold scenario.

Configuration Options

music-on-hold

Setting: **[TServer]** section, Routing Point DN
Default Value: An empty string
Valid Values: The subdirectory and name of the audio file in the MCP root directory, using the following format: `<subdirectory>/<music file name>`; for example: `music/in_queue_welcome.wav`
Changes Take Effect: For the next call

Specifies the name of the file that is played for the music-on-hold treatment when one of the parties in the call is put on hold. The option applies to calls that are passed through this Routing Point, unless a call is distributed with the TRouteCall request that contains the **music-on-hold** key in AttributeExtensions.

default-music

Setting: **[TServer]** section, Agent Login
Default Value: An empty string
Valid Values: The subdirectory and name of the audio file in the MCP root directory, using the following format: `<subdirectory>/<music file name>`; for example: `music/in_queue_welcome.wav`
Changes Take Effect: For the next call

Specifies the name of the file that is played for the music-on-hold treatment to a caller when a respective agent places the call on hold. The option applies to calls distributed to this agent, unless a call is passed through a Routing Point with the **music-on-hold** option, or a call is distributed with the TRouteCall request that contains the **music-on-hold** key in AttributeExtensions.

AttributeExtensions

Key: **music-on-hold**
Type: String
Valid Values: The subdirectory and name of the audio file in the MCP root directory, using the following format: `<subdirectory>/<music file name>`; for example: `music/in_queue_welcome.wav`
Request: TRouteCall

Specifies the name of the file that is played for the music-on-hold treatment when one of the parties in the call is put on hold.

Feature Limitations

- In multi-site deployments with the **music-on-hold** setting enabled in AttributeExtensions, the **iscc-pass-extensions** key in AttributeExtensions must not be set to a value of `local`, because it prevents extensions being passed through ISCC to a remote site.

- In Business Continuity (BC) deployments, the custom music-on-hold setting is propagated with a call transfer in DR-forward scenarios only if the Call Overflow feature is enabled. That is, the following SIP Server Application options must be set in the **[extrouter]** section:
    - **cof-feature**=`true`
    - **default-network-call-id-matching**=`sip`

Setting SIP INVITE timeout for individual DNs

With this enhancement, you can limit how long a SIP transaction will remain in Proceeding state if the only provisional response received was 100 Trying. When this timeout expires, the call is either sent to the DN configured in the **no-response-dn** option, or released if that option is not configured. (See the "Alternate Routing for Unresponsive DNs" section in the SIP Server Deployment Guide.)

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

Feature Configuration

In the SIP Server Switch > DNs > individual DN > TServer section, configure the **sip-trying-timeout** option.

sip-trying-timeout

Setting: **TServer** section, DN level
Default Value: An empty string
Valid Values: `0-256`
Changes Take Effect: For the next call

Specifies the period of time (in seconds) that a SIP call remains in an active state if the only provisional response received was 100 Trying. When this timeout expires, the call is either sent to the DN configured in the **no-response-dn** option, or is released if that option is not configured. If the **sip-trying-timeout** option is not specified, the value of the Application-level option **sip-invite-timeout** is used instead. If the **sip-invite-timeout** option is set to `0`, the default value of 32 seconds is used.

The **sip-trying-timeout** option can be set on the following DN types:

- Extension
- Trunk
- Voice over IP Service with **service-type=**`softswitch`

HTTP Live Streaming

SIP Server supports HTTP Live Streaming (HLS) in the following scenarios:

- When treatments are applied on a Routing Point (TreatmentPlayAnnouncement/TreatmentMusic)
- For music-on-hold treatments, when a call on the DN is placed on hold, or when a call is waiting on an ACD Queue

The feature is available through the MSML protocol.

Feature Configuration

To use this feature, SIP Server must be integrated with MCP version 8.5.161.34 or later.

In the SIP Server configuration, do the following as required:

- For music-on-hold treatments, either at an Application or DN level, specify the proper URL to HTTP Live Streaming server in the **default-music** option.
  For example: **default-music**=`http://123.45.678.90/hls/audio`
- For music treatments, in a routing strategy, specify the URI to the HTTP Live Streaming server in the MUSIC_DN treatment parameter.
- For announcements, in a routing strategy, specify the URI to the HTTP Live Streaming server in the TEXT treatment parameter.

In the MCP configuration, specify the format of audio segments in the **transcoders** parameter.
For example: if audio segments are encoded in the MP3 format, you must add `MP3` into the list of transcoders, as follows:
[mpc].transcoders=`G722 GSM G726 G729 MP3`

> ### Important
> Media Server supports only packed audio segments in MP3 format. It does not support media segments formatted as MPEG-2 Transport Stream or WebVTT.

Recording an Agent Greeting

Starting with SIP Server release 8.1.102.26, you can configure SIP Server to record the agent call leg during the personal greeting. This feature works only when both recording and greeting are enabled for the call.

Feature Configuration

To enable recording of the agent call leg during the personal greeting:

1. In the **TServer** section of the SIP Server Application, configure the following options:
   - Set the **msml-support** option to `true`.
   - Set the **msml-record-support** option to `true`.
   - Set the **record-agent-greeting** option to `true`.
2. Do one of the following:
   - Set the **record** option to `true` on the agent's DN.
   - Set the **record** key to `source` or `destination` in AttributeExtensions of the TRouteCall request.
3. Enable personal greetings by specifying **agent-greeting** and **customer-greeting keys** in AttributeExtensions of the TRouteCall request.

record-agent-greeting

Setting: **TServer** section, Application level
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: For the next call

Specifies whether the agent greeting or the customer greeting must be recorded when both recording and greeting are enabled for the call.

- If set to `true`, the agent greeting is recorded.
- If set to `false`, the customer greeting is recorded.

AttributeExtensions

Key: **record-agent-greeting**
Type: String
Valid Values: `true, false`
Request: TRouteCall

Specifies whether the agent greeting or the customer greeting must be recorded when both recording and greeting are enabled for the call.

- If set to `true`, the agent greeting is recorded.
- If set to `false`, the customer greeting is recorded.

Feature Limitations

- This feature is supported for MSML-based integration only.
- This feature is supported only for greetings played for inbound calls.
- This feature is not supported for greetings configured in the Agent Login object.

Controlling Early Media with a Routing Strategy

In deployments with the Early Media for Inbound Calls feature enabled, this new enhancement (a new **charge-type** extension key) enables you to create a routing strategy that does the following for an inbound call:

- Switch audio treatments from cost-free early media to an established state (charged) in a SIP dialog, which can be made at the initial TApplyTreatment or at any sequential TApplyTreatment. All consecutive audio treatments in this dialog will be charged.
- Play initial audio treatments in cost-free early media in deployments that are configured to play audio treatments at a cost, until a TApplyTreatment request containing the **charge-type** key set to 2 (charged) arrives.

The transition from early media to an established state can be made only once within a SIP dialog and only when changing from cost-free audio treatments to charged audio treatments.

This functionality is supported for MSML deployments and is not supported for NETANN deployments.

Feature Configuration

1. Enable the Early Media for Inbound Calls feature as described in the SIP Server Deployment Guide.
2. In the routing strategy, specify the **charge-type** key in AttributeExtensions of the TApplyTreatment request.

AttributeExtensions

Key: **charge-type**
Type: Integer
Values: 1, 2
Request: TApplyTreatment

If set, the value of this key overrides the value set in the **charge-type** configuration option for the current call.

- 1—Free. When SIP Server receives this key in the initial TApplyTreatment request, SIP Server forces audio treatments to be played in early media, free of charge, instead of media in the established state, in deployments where the **charge-type** option is set to **2** (charged). Consecutive audio treatments are played in early media until the new TApplyTreatment request containing the **charge-type** key set to 2 (charged) arrives.
- 2—Charged. SIP Server forces audio treatments to be played in the established state and ignores the **charge-type** key value in consecutive TApplyTreatment requests.

Dial Plan enhancements including support for SIP Feature Server Dial Plan

SIP Server now offers the option to use SIP Feature Server as an “external dial plan” as an alternative to the internal SIP Server dial plan. Each choice offers distinct advantages to consider when choosing which dial plan to use. (Note that dial plans may not be combined.)

Feature Server Dial Plan Highlights:

- User-based calling preferences for Call Waiting and Call Forwarding (including Find-Me-Follow-Me)
- Flexible rules with pattern matching logic for choosing a trunk for outgoing calls
- Enhanced support for deployments where voicemail mailboxes are assigned to users (but not to DNs)

SIP Server Dial Plan Highlights:

- Many supported parameters for advanced dial-plan rules, such as “onbusy”, “type”, “calltype”, “clir”, and more
- Native support by SIP Server (smaller footprint, less complexity if Feature Server is not required for the deployment)

SIP Server offers additional control over how a dial plan is applied to the destination of TRouteCall and/or to multi-site (ISCC) calls that are routed through an External Routing Point with two configuration options:

- The **rp-use-dial-plan** configuration option changes the default behavior of the dial plan to any one of the following:
  - SIP Server does not apply any dial plan.
  - SIP Server applies only the digit translation to a dial plan target.
  - SIP Server applies the digit translation and forwarding rules to a dial plan target.

    The **rp-use-dial-plan** option applies to both SIP Server and SIP Feature Server dial plans. If the **UseDialPlan** key-value pair is present in AttributeExtensions of TRouteCall, then it takes priority over the **rp-use-dial-plan** option.
- The **enable-iscc-dial-plan** option enables SIP Server to apply the dial plan to the target destination when a call is routed from an External Routing Point to a DN at the destination site.

Feature Configuration

Using Feature Server Dial Plan

1. Administer the SIP Feature Server dial plan as described in the SIP Feature Server Administration Guide.
2. Configure the SIP Server that is associated with the Feature Server by setting the following option in the **[TServer]** section of the SIP Server Application:
   - **dial-plan**—Set this option to `fs-dialplan`, as described in the SIP Feature Server Deployment Guide.
3. Under a SIP Server Switch object that is associated with the SIP Server, create a VOIP Service DN named **fs-dialplan** and configure these options:
   - **service-type**—Set this option to `extended`.
     Important: Ensure that you add the final slash character (/) to the end of each of the following URLs.
   - **url**—Set this option to `http://`*`FS Node:port/`*
     For n+1 High Availability (HA), add the following parameters:
     - **url-1** = `http://`*`FS Node2:port/`*
     - **url-2** = `http://`*`FS Node3:port/`*
     - **url-n** = `http://`*`FS Node_N:port/`*

     Important: A Feature Server's dial plan URL must be configured only on a VOIP Service DN that was created on the Switch controlled by the SIP Server that is connected to that particular Feature Server.
4. If required, configure the following options in the SIP Server Application object, the **[TServer]** section:
   - **rp-use-dial-plan**—Set this option to a value suitable for your environment.
   - **enable-iscc-dial-plan**—Set this option to `true` to enable SIP Server to apply the dial plan to multi-site (ISCC) calls that are routed through an External Routing Point.
5. (Optional) In a routing strategy, set the **UseDialPlan** key extension in TRouteCall. The key extension setting takes priority over configuration options.

Using SIP Server Dial Plan

1. Configure the Dial Plan feature as described in the Framework 8.1 SIP Server Deployment Guide.
2. If required, configure the following options in the SIP Server Application, the **[TServer]** section:
   - **rp-use-dial-plan**—Set this option to a value suitable for your environment.
   - **enable-iscc-dial-plan**—Set this option to `true` to enable SIP Server to apply the dial plan to multi-site (ISCC) calls that are routed through an External Routing Point.
3. (Optional) In a routing strategy, set the **UseDialPlan** key extension in TRouteCall. The key extension setting takes priority over configuration options.

Configuration Options

rp-use-dial-plan

Setting: **TServer** section, Application level
Default Value: `default`
Valid Values: `default, full, partial, false`
Changes Take Effect: Immediately

Specifies how SIP Server applies the dial plan:

- `default`—For a SIP Server dial plan, the same as the `false` value. For a Feature Server dial plan, the same as the `partial` value.
- `full`—The dial plan is applied to the destination of TRouteCall, including the digit translation and forwarding rules.
- `partial`—Only the digit translation is applied to a dial-plan target. Forwarding rules, such as forwarding on no answer (`ontimeout`), forwarding on busy (`onbusy`), forwarding on DND (`ondnd`), forwarding on no response (`onunreach`), and forwarding on not SIP registered (`onnotreg`) are not applied. Valid for both SIP Server and SIP Feature Server dial plans.
- `false`—No dial plan is applied to the destination of TRouteCall.

> Important
>
> If the SIP Server dial plan is used, SIP Server selects the dial plan assigned to the caller. This is the dial plan configured for the DN/Agent Login of the DN for internal calls, or the Trunk DN for inbound calls, or the Application-level option if no DN/Agent-Login-level dial plan is configured.

enable-iscc-dial-plan

Setting: **TServer** section, Application level
Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: At the next call

Specifies whether SIP Server applies the dial plan to the agent destination of multi-site (ISCC) calls that are routed through an External Routing Point (**cast-type**=`route-notoken`), as follows:

- If set to `true`, the dial plan (full, including the digit translation and forwarding rules) is applied.
- If set to `false`, the dial plan is not applied.

This option must be configured on the remote (destination) site. SIP Server applies the dial plan when a call is routed from an External Routing Point to a DN at the destination site.

Important

SIP Server will still apply the dial plan to the External Routing Point destination of multi-site (ISCC) calls, and this will take priority over the agent DN destination dial-plan rule regardless of the setting of **enable-iscc-dial-plan**.

AttributeExtensions

Key: **UseDialPlan**
Type: String
Values: `full, partial, false`
Request: TRouteCall

Specifies how SIP Server applies the dial plan:

- `full`—The dial plan is applied to the destination of TRouteCall, including the digit translation and forwarding rules.
- `partial`—Valid for both SIP Server and SIP Feature Server dial plans. Only the digit translation is applied to a dial-plan target. Forwarding rules, such as forwarding on no answer (`ontimeout`), forwarding on busy (`onbusy`), forwarding on DND (`ondnd`), forwarding on no response (`onunreach`), and forwarding on not SIP registered (`onnotreg`) are not applied.
- `false`—No dial plan is applied to the destination of TRouteCall.

Important

- For ISCC calls, this extension is applied only to calls routed through an External Routing Point (**cast-type**=`route-notoken`).
- This extension is not supported in Business Continuity deployments.

HTTP Monitoring Interface

Starting with version 8.1.102.13, SIP Server provides the ability to monitor various operational statistics for its internal modules and statistics relating to trunks.

Starting with version 8.1.103.25, SIP Server adds the ability to monitor statistics related to SIP Feature Server interactions.

Starting with version 8.1.103.35, SIP Server adds the ability to monitor statistics related to Extended Services (XS) components.

Starting with version 8.1.103.72, SIP Server adds the following statistics in separate tables for Trunk, Softswitch, MSML, and Trunk Group devices: error statistics, the total number of calls created on the device, the number of out-of-service detection instances per device, and location matching instances.

Starting with version 8.1.103.95, SIP Server adds monitoring of state and quantitative statistics for T-Library client connections of the following SIP Server threads: Session Controller (the main T-Server thread), T-Controller, Interaction Proxy, and Smart Proxy. See SIP Server threads statistics for details.

This topic covers the following:

- SIP Server statistics
- Trunk, Softswitch, MSML, and Trunk Group statistics
- How to monitor statistics
- SIP Server statistics details
- SIP Feature Server statistics
- Configuration options
- Feature limitations

SIP Server Statistics

SIP Server collects statistics for the following internal modules:

- sipServer—General SIP Server statistics
- sipStackObjects—SIP stack statistics
- sipCallManager—Call and client statistics
- sipTSCP—ISCC statistics
- sipSessionController—T-Library statistics
- sipServiceChecker—Out-of-service detection statistics
- tLibClientsStatistics—SIP Server threads statistics

Trunk, Softswitch, MSML, and Trunk Group Statistics

SIP Server collects the following statistics (sipTrunkStatistics) for each configured Trunk, Softswitch, MSML, and Trunk Group DNs:

- Current statistics (real-time)
    - Number of calls currently established via this trunk (sum of incoming and outgoing calls)
    - Current call rate (average number of calls per second, both incoming and outgoing). Averaging interval is specified by the Application-level option **operational-stat-timeout** (in seconds, default value is 10 seconds).
    - Capacity (maximum number of calls allowed on a trunk, both incoming and outgoing)
    - Capacity group (to which a trunk belongs)
    - In Service status
    - Error code counters: 4xx, 5xx, 6xx. Error codes that are received by SIP Server for the device since SIP Server started.
    - Number of calls created since SIP Server started. It is a summary of incoming and outgoing calls created at the device. It includes failed to establish calls due to an error response or timeout.
    - Number of out-of-service detection instances since SIP Server started.
    - Statistics for Trunk, MSML, and Softswitch devices only: If location matching is enabled, devices that don't satisfy the location matching configuration are not displayed. Location matching is configured by the **enable-strict-location-match**, **overflow-location-map**, and **find-trunk-by-location** configuration options.
- Summary statistics (over a period of time, such as an hour or a day)
    - Peak number of calls
    - Peak call rate
    - Call attempts (total number of new calls)
    - Total number of released calls
    - Summary period start time
    - Summary period end time

SIP Server collects the following statistics (sipTrunkStatistics) for each configured Capacity Group:

- Current statistics
    - Current number of calls (combined number for all the trunks in a group)
    - Current call rate (combined rate for all the trunks in a group)
    - Capacity
- Summary statistics

- Peak number of calls
- Peak call rate
- Call attempts
- Total number of released calls
- Summary period start time
- Summary period end time
- Number of calls created since SIP Server started

The period of summary statistics calculation is configurable with the Application-level option **summary-stat-timeout**.

How to Monitor Statistics

There are two possible ways to monitor the collected statistics:

- Using the HTTP interface
- Using the dedicated SIP Server log file

Monitoring Statistics via HTTP interface

To enable the HTTP interface, set the **http-port** option to a valid and unoccupied port number in the range of 1024-65535 (values lower than 1024 are system reserved ports). Only the HTTP interface is available on the configured port (HTTPS is not available).

To get the statistics data, the following URL must be retrieved with the HTTP GET request: `http://<SIP Server IP address>:<configured HTTP port>` (for example, http://192.168.0.1:8088)

Depending on the path used in the URL, the statistics page can be provided in the HTML or XML format:

- To get the statistics in the HTML format, use an empty path ("" or "/") or path `/server`.
  For example: `http://192.168.0.1:8088` or `http://192.168.0.1:8088/server`.
  The above URL returns a root statistics page with the list of statistics for SIP Server internal modules, with each list item being a link to the statistic data for that module.
- To get the statistics in the XML format, use path `/serverx`.
  For example: `http://192.168.0.1:8088/serverx`.
  The above URL returns a root statistics page with the full statistics dump, including data for each SIP Server internal module.

To get the statistics page for one module, add the URL parameter with the module name. For example, to get the Trunk Statistics page in the HTML format, the following URL must be used: `http://192.168.0.1:8088/server?sipTrunkStatistics`.

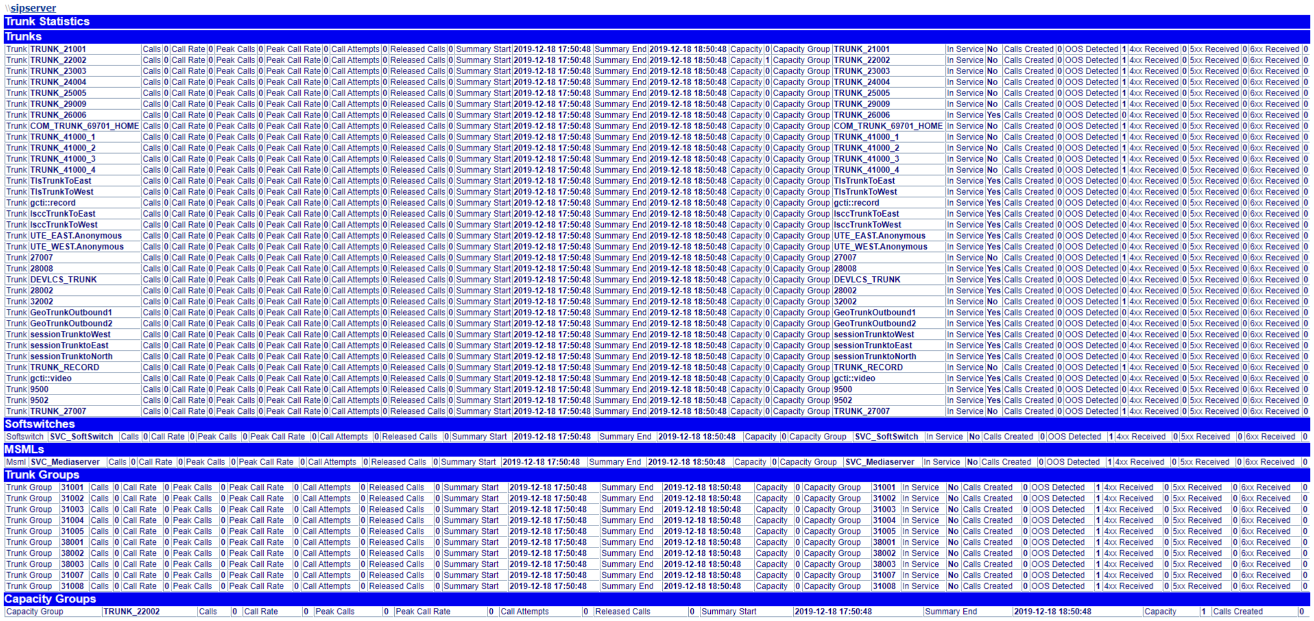
Trunk Statistics Examples

- In the HTML format:

[+] Show HTML source

```
<table>
<caption>Trunk Statistics</caption>
<tr>
<table>
<caption>Trunks</caption>
<tr>
    <td>Trunk</td><td>21001</td>
    <td>Calls</td><td>100</td>
    <td>Call Rate</td><td>15</td>
    <td>Peak Calls</td><td>150</td>
    <td>Peak Call Rate</td><td>20</td>
    <td>Call Attempts</td><td>5000</td>
    <td>Released Calls</td><td>4900</td>
    <td>Summary Start</td><td>2016-01-01T12:00:01</td>
    <td>Summary End</td><td>2016-01-01T13:00:01</td>
    <td>Capacity</td><td>500</td>
    <td>Capacity Group</td><td>MyTrunks</td>
    <td>In Service</td><td>Yes</td>
</tr>
<tr>
    <td>Trunk</td><td>21002</td>
    <td>Calls</td><td>200</td>
    <td>Call Rate</td><td>30</td>
    <td>Peak Calls</td><td>250</td>
    <td>Peak Call Rate</td><td>40</td>
    <td>Call Attempts</td><td>10000</td>
    <td>Released Calls</td><td>9800</td>
    <td>Summary Start</td><td>2016-01-01T12:00:01</td>
    <td>Summary End</td><td>2016-01-01T13:00:01</td>
    <td>Capacity</td><td>500</td>
    <td>Capacity Group</td><td>MyTrunks</td>
    <td>In Service</td><td>Yes</td>
</tr>
</table>
</tr>
<tr>
<table>
<caption>Capacity Groups</caption>
<tr>
    <td>Capacity Group</td><td>MyTrunks</td>
    <td>Calls</td><td>300</td>
    <td>Call Rate</td><td>45</td>
    <td>Peak Calls</td><td>400</td>
    <td>Peak Call Rate</td><td>60</td>
    <td>Call Attempts</td><td>15000</td>
    <td>Released Calls</td><td>14700</td>
    <td>Summary Start</td><td>2016-01-01T12:00:01</td>
    <td>Summary End</td><td>2016-01-01T13:00:01</td>
    <td>Capacity</td><td>500</td>
</tr>
</table>
</tr>
</table>
```

Example of statistics displayed in the HTML format in a browser:



\\sipserver

Trunk Statistics

Trunks

Trunk	TRUNK_21001	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	TRUNK_21001	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	TRUNK_22002	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	1	Capacity Group	TRUNK_22002	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	TRUNK_23003	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	TRUNK_23003	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	TRUNK_24004	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	TRUNK_24004	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	TRUNK_25005	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	TRUNK_25005	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	TRUNK_29009	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	TRUNK_29009	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	TRUNK_26006	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	TRUNK_26006	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	COM_TRUNK_69701_HOME	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	COM_TRUNK_69701_HOME	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	TRUNK_41000_1	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	TRUNK_41000_1	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	TRUNK_41000_2	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	TRUNK_41000_2	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	TRUNK_41000_3	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	TRUNK_41000_3	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	TRUNK_41000_4	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	TRUNK_41000_4	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	TlsTrunkToEast	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	TlsTrunkToEast	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	TlsTrunkToWest	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	TlsTrunkToWest	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	gcti::record	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	gcti::record	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	IsccTrunkToEast	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	IsccTrunkToEast	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	IsccTrunkToWest	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	IsccTrunkToWest	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	UTE_EAST.Anonymous	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	UTE_EAST.Anonymous	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	UTE_WEST.Anonymous	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	UTE_WEST.Anonymous	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	27007	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	27007	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	28008	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	28008	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	DEVLCS_TRUNK	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	DEVLCS_TRUNK	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	28002	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	28002	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	32002	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	32002	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	GeoTrunkOutbound1	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	GeoTrunkOutbound1	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	GeoTrunkOutbound2	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	GeoTrunkOutbound2	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	sessionTrunktoWest	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	sessionTrunktoWest	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	sessionTrunktoEast	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	sessionTrunktoEast	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	sessionTrunktoNorth	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	sessionTrunktoNorth	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	TRUNK_RECORD	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	TRUNK_RECORD	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	gcti::video	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	gcti::video	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	9500	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	9500	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	9502	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	9502	In Service	Yes	Calls Created	0	OOS Detected	0	4xx Received	0	5xx Received	0	6xx Received	0
Trunk	TRUNK_27007	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	TRUNK_27007	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0

Softswitches

Softswitch	SVC_SoftSwitch	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	SVC_SoftSwitch	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0

MSMLs

Msml	SVC_Mediaserver	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	SVC_Mediaserver	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0

Trunk Groups

Trunk Group	31001	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	31001	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk Group	31002	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	31002	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk Group	31003	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	31003	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk Group	31004	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	31004	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk Group	31005	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	31005	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk Group	38001	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	38001	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk Group	38002	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	38002	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk Group	38003	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	38003	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk Group	31007	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	31007	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0
Trunk Group	31008	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	0	Capacity Group	31008	In Service	No	Calls Created	0	OOS Detected	1	4xx Received	0	5xx Received	0	6xx Received	0

Capacity Groups

Capacity Group	TRUNK_22002	Calls	0	Call Rate	0	Peak Calls	0	Peak Call Rate	0	Call Attempts	0	Released Calls	0	Summary Start	2019-12-18 17:50:48	Summary End	2019-12-18 18:50:48	Capacity	1	Calls Created	0

- In the XML format:

[+] Show XML source

```
<sipTrunkStatistics id='sipTrunkTable'>
<sipTrunkData id='sipTrunkData'>
<TRUNK id='00150F72-6A0A-1D4A-B024-3F3C330AAA77' type='7' rem="Trunk">trunk1_sjc</TRUNK>
<CURRENT_CALLS id='00150FA4-6A0A-1D4A-B024-3F3C330AAA77' type='4'
rem="Calls">0</CURRENT_CALLS>
<CURRENT_CALL_RATE id='00150FAE-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="Call
Rate">0</CURRENT_CALL_RATE>
<PEAK_CALLS id='00150FC2-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="Peak Calls">0</PEAK_CALLS>
<PEAK_CALL_RATE id='00150FD6-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="Peak Call
Rate">0</PEAK_CALL_RATE>
<CALL_ATTEMPTS id='00150FE0-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="Call
Attempts">0</CALL_ATTEMPTS>
<RELEASED_CALLS id='00150FF4-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="Released
Calls">0</RELEASED_CALLS>
<SUMMARY_START id='00150FFE-6A0A-1D4A-B024-3F3C330AAA77' type='5' rem="Summary
Start">1568671922665</SUMMARY_START>
<SUMMARY_END id='00151012-6A0A-1D4A-B024-3F3C330AAA77' type='5' rem="Summary
End">1568675522665</SUMMARY_END>
<CAPACITY id='0015101C-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="Capacity">0</CAPACITY>
<CAPACITY_GROUP id='00151030-6A0A-1D4A-B024-3F3C330AAA77' type='7' rem="Capacity
Group">PSTN_Trunk_ash</CAPACITY_GROUP>
<IN_SERVICE id='0015103A-6A0A-1D4A-B024-3F3C330AAA77' type='7' rem="In
Service">Yes</IN_SERVICE>
<NCALLSCREATED id='00150FA4-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="Calls
Created">0</NCALLSCREATED>
<NOOS_DETECTED id='0015103A-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="OOS
Detected">0</NOOS_DETECTED>
<N4xx_RECEIVED id='0015103A-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="4xx
Received">0</N4xx_RECEIVED>
<N5xx_RECEIVED id='0015103A-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="5xx
Received">0</N5xx_RECEIVED>
<N6xx_RECEIVED id='0015103A-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="6xx
```

```
Received">0</N6xx_RECEIVED>
</sipTrunkData>
<sipTrunkData id='sipTrunkData'>
<TRUNK id='001516E8-6A0A-1D4A-B024-3F3C330AAA77' type='7' rem="Trunk">trunk2_sjc</TRUNK>
<CURRENT_CALLS id='00151706-6A0A-1D4A-B024-3F3C330AAA77' type='4'
rem="Calls">0</CURRENT_CALLS>
<CURRENT_CALL_RATE id='00151710-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="Call
Rate">0</CURRENT_CALL_RATE>
<PEAK_CALLS id='00151724-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="Peak Calls">0</PEAK_CALLS>
<PEAK_CALL_RATE id='0015172E-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="Peak Call
Rate">0</PEAK_CALL_RATE>
<CALL_ATTEMPTS id='00151742-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="Call
Attempts">0</CALL_ATTEMPTS>
<RELEASED_CALLS id='0015174C-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="Released
Calls">0</RELEASED_CALLS>
<SUMMARY_START id='00151760-6A0A-1D4A-B024-3F3C330AAA77' type='5' rem="Summary
Start">1568671922665</SUMMARY_START>
<SUMMARY_END id='0015176A-6A0A-1D4A-B024-3F3C330AAA77' type='5' rem="Summary
End">1568675522665</SUMMARY_END>
<CAPACITY id='0015177E-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="Capacity">0</CAPACITY>
<CAPACITY_GROUP id='00151792-6A0A-1D4A-B024-3F3C330AAA77' type='7' rem="Capacity
Group">msml_ash</CAPACITY_GROUP>
<IN_SERVICE id='0015179C-6A0A-1D4A-B024-3F3C330AAA77' type='7' rem="In
Service">No</IN_SERVICE>
<NCALLSCREATED id='00150FA4-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="Calls
Created">0</NCALLSCREATED>
<NOOS_DETECTED id='0015103A-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="OOS
Detected">0</NOOS_DETECTED>
<N4xx_RECEIVED id='0015103A-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="4xx
Received">0</N4xx_RECEIVED>
<N5xx_RECEIVED id='0015103A-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="5xx
Received">0</N5xx_RECEIVED>
<N6xx_RECEIVED id='0015103A-6A0A-1D4A-B024-3F3C330AAA77' type='4' rem="6xx
Received">0</N6xx_RECEIVED>
</sipTrunkData>
</sipTrunkStatistics>
```

Monitoring Statistics via Log File

SIP Server always creates a dedicated log file for a statistics output despite of the values set in the log options. The name of the file contains suffix 1536 (for example, `server1-1536.20160201_195851_685.log`).

Statistics are written to the log file periodically, with a period specified by the Application-level option **operational-stat-timeout** (default value is 10 seconds).

Statistics log format

A statistics log file uses the following line format for statistics output:

```
<Timestamp in ISO 8601 format> <Statistics module name>
<Parameter>=<Value> [<Parameter>=<Value>]
```

For example, for the Call Manager module, the output for the "Number of devices" statistics is as follows:

```
2016-01-01T12:00:01.001 sipCallManager NDEVICES=150
```

For more sophisticated Trunk statistics, the output is as follows:

```
2016-01-01T12:00:01.001 sipTrunkStatistics TRUNK=21001
CURRENT_CALLS=100 CURRENT_CALL_RATE=15 PEAK_CALLS=150
PEAK_CALL_RATE=20 CALL_ATTEMPTS=5000 RELEASED_CALLS=4900
SUMMARY_START=2016-01-01T12:00:01
SUMMARY_END=2016-01-01T13:00:01 CAPACITY=500
CAPACITY_GROUP=MyTrunks IN_SERVICE=Yes
```

SIP Server Statistics Details

Each SIP Server internal module has its own statistics section. Each section has the identifier that is used in the log and XML output to distinguish one section from another. This identifier is also used as a URL parameter if a user wishes to get an HTML/XML page with only one statistics section.

The statistics for each SIP Server module is described below. Each statistic record is described in a table that has the following columns:

- **ID**—The ID that is used in the log and XML output.
- **Description**—The general description of the statistic record. It is also used in the HTML output.
- **Comments**—Additional information if the description of the record is not self-explanatory.

General SIP Server Statistics

Section ID: **sipServer**

ID	Description	Comments
SIPS_PROCESS_ID	Process identifier	Identifier that is assigned by an Operating System to a SIP Server process
SIPS_MEMORY_USAGE	Memory usage	In bytes

SIPS_CPU_USAGE	CPU usage	In percent
PROCESS_INFO_TIME	Process info time	Time range on which statistics are gathered
NAME	Name	Name of the SIP Server Application in the Configuration Layer
BIT	Bit data model	32 or 64 bit
PLATFORM	Platform	
SERVER_VERSION	Server version	
COMPILED_DATE	Compiled date	
XS_CPU_USAGE	CPU usage	Extended Service CPU usage on a singe CPU core. Applies only to SIP Cluster deployments starting with SIP Server release 8.1.103.29.
XS_REQUESTS_RATE	Requests rate	Rate of requests processed by the Extended Services component. Applies only to SIP Cluster deployments starting with SIP Server release 8.1.103.29.

SIP Stack Statistics

Section ID: **sipStackObjects**

ID	Description	Comments
DIALOG_CREATED	SIP Dialogs created	A cumulative metric that is reset to zero on restart.
DIALOG_DELETED	SIP Dialogs deleted	A cumulative metric that is reset to zero on restart.
MESSAGES_CREATED	SIP Messages created	A cumulative metric that is reset to zero on restart.
MESSAGES_DELETED	SIP Messages deleted	A cumulative metric that is reset to zero on restart.
CLIENT_TRANSACTION_CREATED	SIP Client transactions created	A cumulative metric that is reset to zero on restart.
CLIENT_TRANSACTION_DELETED	SIP Client transactions deleted	A cumulative metric that is reset to zero on restart.
SERVER_TRANSACTION_CREATED	SIP Server transactions created	A cumulative metric that is reset

		to zero on restart.
SERVER_TRANSACTION_DELETED	SIP Server transactions deleted	A cumulative metric that is reset to zero on restart.
TRANSPORT_CREATED	Transports created	A cumulative metric that is reset to zero on restart.
TRANSPORT_DELETED	Transports deleted	A cumulative metric that is reset to zero on restart.
DATASENT	Data sent	In bytes. If a value is not increased, it might be used as indication of the backup mode of SIP Server.
DATARECEIVED	Data received	In bytes. If a value is not increased, it might be used as indication of the backup mode of SIP Server.
RESPONSE_TIME_LESS20	Response time less than 20 ms	Response time to SIP messages sent by SIP Server.
RESPONSE_TIME_20TO50	Response time 20 to 50 ms	Response time to SIP messages sent by SIP Server.
RESPONSE_TIME_50TO100	Response time 50 to 100 ms	Response time to SIP messages sent by SIP Server.
RESPONSE_TIME_100TO200	Response time 100 to 200 ms	Response time to SIP messages sent by SIP Server.
RESPONSE_TIME_200TO500	Response time 200 to 500 ms	Response time to SIP messages sent by SIP Server.
RESPONSE_TIME_500TO1SEC	Response time 500 ms to 1 sec	Response time to SIP messages sent by SIP Server.
RESPONSE_TIME_1TO5SEC	Response time 1 to 5 sec	Response time to SIP messages sent by SIP Server.
RESPONSE_TIME_5TO10SEC	Response time 5 to 10 sec	Response time to SIP messages sent by SIP Server.
RESPONSE_TIME_MORE10SEC	Response time more than 10 sec	Significant increase of the value might indicate network problems or a SIP Server overload condition.

Call and Client Statistics

Section ID: **sipCallManager**

ID	**Description**	**Comments**
CM_THREAD_ID	Thread ID	
CM_CPU_USAGE	CPU usage of call manager thread	In percent.

NCALLSCREATED	Number of calls created	A cumulative metric that is reset to zero on restart.
NDIALOGS	Number of dialogs	Current active SIP dialogs.
NCALLS	Number of calls	Current active calls.
NPARTIES	Number of parties	Current active parties. If the number of calls is zero, a nonzero value of the number of parties might indicate a leak of resources which might lead to an abnormal behavior of SIP Server.
HADATASENT	HA sync data sent	In bytes.
HADATARECEIVED	HA sync data received	In bytes.
NROUTINGTIMEOUTS	Number of routing timeouts	Increases when the routing timer expires (specified by the option **router-timeout**). Significant increase of the value might indicate issues in the routing strategies or router overload.
NLOGGEDAGENTS	Number of logged on agents	
NREGISTEREDDNS	Number of registered DNs	Number of registered DNs by using a TRegisterAddress request.
NTLIBCLIENTS	Number of T-Library clients	Number of T-Library clients currently connected to SIP Server.
NCALLSABANDONED	Number of abandoned calls	Number of calls released by a caller before the call was answered.
NCALLRECORDINGFAILED	Number of failed call recording sessions	Significant increase of the value might indicate problems with the recording services.
NSIPREGISTEREDEP	Number of active SIP registrations	
NSIPEXPIREDREGS	Number of expired SIP registrations	Number of SIP registrations that were expired since SIP Server startup. Significant increase of the value might indicate problems with the network.
NCALLSOVRLREJECTED	Number of rejected calls due to overload control	Related to the static overload control feature.

NMSMLLOCATIONFAILED	Number of MSML location resolution failed	Significant increase of the value might indicate problems with configuration of geo-location of MSML services.

ISCC Statistics

Section ID: **sipTSCP**

ID	**Description**	**Comments**
ISCC_ACTIVE_ORIG_TRANSACTIONS	ISCC active orig transactions	
ISCC_ACTIVE_DEST_TRANSACTIONS	ISCC active destination transactions	
ISCC_SUCCEEDED_ORIG_TRANSACTIONS	ISCC succeeded origination transactions	
ISCC_SUCCEEDED_DEST_TRANSACTIONS	ISCC succeeded destination transactions	
ISCC_FAILED_ORIG_TRANSACTIONS	ISCC failed origination transactions	Significant increase of the value might indicate issues in the communication between sites in a multisite environment.
ISCC_FAILED_DEST_TRANSACTIONS	ISCC failed destination transactions	Significant increase of the value might indicate issues in the communication between sites in a multisite environment.

T-Library Statistics

Section ID: **sipSessionController**

ID	**Description**	**Comments**
MAIN_THREAD_ID	Thread ID	
MAIN_CPU_USAGE	CPU usage	In percent.
NAPPLY_TREATMENTS	Number of Apply Treatments	Number of processed TApplyTreatment requests. A cumulative metric that is reset to zero on restart.
URS_RESPONSE_LESS50	URS Response less than 50 ms	URS response time means the time passed between the moment the call is queued on a Routing Point (indicated by

		EventQueued) and the moment the call is routed to a destination, including the default destination in case of URS failure/ timeout (indicated by EventRouteUsed). Long response times might indicate problems with the network or URS overload. Response times corresponding to the **router-timeout** option value most likely indicate a URS failure/timeout (e.g. as a result of incorrect routing strategy).
URS_RESPONSE_50TO100	URS Response 50 to 100 ms	
URS_RESPONSE_100TO200	URS Response 100 to 200 ms	
URS_RESPONSE_200TO500	URS Response 200 to 500 ms	
URS_RESPONSE_500TO1SEC	URS Response 500 ms to 1sec	
URS_RESPONSE_1TO5SEC	URS Response 1 to 5 sec	
URS_RESPONSE_MORE5SEC	URS Response more than 5 sec	
NUSER_DATA_UPDATES	Number of User Data updates	Number of processed requests related to UserData (TAttachUserData, TUpdateUserData, TDeletePair, TDeleteUserData). A cumulative metric that is reset to zero on restart.
NTREQUESTS	Number of T-Requests	Number of all processed T-Library requests (of any type). A cumulative metric that is reset to zero on restart.

Out-Of-Service Detection Statistics

Section ID: **sipServiceChecker**

ID	**Description**	**Comments**
SRVC_THREAD_ID	Thread ID	
SRVC_CPU_USAGE	CPU usage	
SRVC_NOOS_DEVICES	Number Out Of Service devices	Significant increase of the value might indicate problems with the network

SIP Server threads statistics

To enable statistics for T-Library client connections of the SIP Server threads, set the t-library-stats-enabled configuration option to `true`. SIP Server provides state and quantitative statistics for the following SIP Server threads:

- Session Controller
- T-Controller
- Interaction Proxy
- Smart Proxy

Table name: **tLibClientsStatistics**
Table ID: **tLibClientsPage**

For each thread, SIP Server displays the client connection statistics and error statistics. See also a limitation for this feature.

Client connection statistics

Statistics tables for client connections have the following names and IDs per SIP Server thread:

- Session Controller: **SC_clientsStatistics** and **id='SC_clientsDataTable'**
- T-Controller: **TC_clientsStatistics** and **id='TC_clientsDataTable'**
- Interaction Proxy: **IP_clientsStatistics** and **id='IP_clientsDataTable'**
- Smart Proxy: **SP_clientsStatistics** and **id='SP_clientsDataTable'**

ID	Description	Comment
CLIENT	Client	The name of the connected client.
CURRENT_CONN_STATE	Client Connection State	The connection state of the client either 1 (connected) or 0 (disconnected).
NSTATE_DISCONNECTED	Number of client disconnects	The number of client disconnections since SIP Server is started.
NCLIENT_REQUESTS	Accumulated number of requests	The accumulated number of requests sent by the client since SIP Server is started.
NCLIENT_EVENTS	Accumulated number of events (errors including)	The accumulated number of events sent by SIP Server to the client since SIP Server is started.
NCLIENT_ERROR_EVENTS	Accumulated number of errors	The accumulated number of errors among events sent by SIP Server to the client since SIP Server is started.
CLIENT_OUTPUT_QUEUE	Output queue size (bytes)	The output queue size (the

ID	Description	Comment
		connection output buffer), in bytes.
CLIENT_DATA_RX_BYTES	Accumulated incoming bytes	The accumulated incoming bytes received by SIP Server from the client since SIP Server is started.
CLIENT_DATA_TX_BYTES	Accumulated outgoing bytes	The accumulated outgoing bytes sent by SIP Server to the application since SIP Server is started.

Error statistics

Error statistics tables contain embedded tables. There are as many tables as different types of errors that are received by a particular thread since SIP Server is started. The error statistics tables have the following names and IDs per SIP Cluster thread:

- Session Controller: **SC_errorsStatistics** and **id='SC_errorsDataTable'**
- T-Controller: **TC_errorsStatistics** and **id='TC_errorsDataTable'**
- Interaction Proxy: **IP_errorsStatistics** and **id='IP_errorsDataTable'**
- Smart Proxy: **SP_errorsStatistics** and **id='SP_errorsDataTable'**

ID	Description	Comment
ERROR_CODE	ErrorCode	The digital error code.
ERROR_TEXT	Error Meaning	The error description.
N_ERRORS	Accumulated number of errors	The accumulated number of errors of this particular type thread that are received since SIP Server is started.

Example of Session Controller client and error statistics in the XML format:

[+] Show XML source

```
<tLibClientsStatistics id='tLibClientsPage'>
<SC_clientsStatistics id='SC_clientsDataTable'>
<ClientData id='ClientData'>
<CLIENT id='9C9A80A9-9ED3-40C6-8077-A582F30CBD5E' type='7'
rem="Client">interactionProxy</CLIENT>
<CURRENT_CONN_STATE id='2B50E251-E09E-45B1-A79F-5E104889B283' type='4' rem="Client Connection
State">1</CURRENT_CONN_STATE>
<NSTATE_DISCONNECTED id='ECC42C89-729A-49FE-84F4-E496DECFE402' type='4' rem="Number of client
disconnects">0</NSTATE_DISCONNECTED>
<NCLIENT_REQUESTS id='181B9E5D-5184-48BC-A7C2-012F8B302C4F' type='7' rem="Accumulated number
of requests">1</NCLIENT_REQUESTS>
<NCLIENT_EVENTS id='BAEE7D8E-D422-4662-997D-5870DF686409' type='7' rem="Accumulated number of
events (errors including)">83</NCLIENT_EVENTS>
<NCLIENT_ERROR_EVENTS id='4CCD9E8B-52A6-4563-8665-16AF7EC556E6' type='7' rem="Accumulated
number of errors">0</NCLIENT_ERROR_EVENTS>
<CLIENT_OUTPUT_QUEUE_BYTES id='45BD9498-1131-434F-B0D5-34173D3AE5E1' type='7' rem="Output
queue size (bytes)">0</CLIENT_OUTPUT_QUEUE_BYTES>
```

```
<CLIENT_DATA_RX_BYTES id='A3E173C1-E3BB-4E70-B03C-5397DBE1EC8D' type='7' rem="Accumulated
incoming bytes">386</CLIENT_DATA_RX_BYTES>
<CLIENT_DATA_TX_BYTES id='F20B02D0-6F95-4CF2-A955-C97CF2D1AA13' type='7' rem="Accumulated
outgoing bytes">9789</CLIENT_DATA_TX_BYTES>
</ClientData>
</SC_clientsStatistics>
<SC_errorsStatistics id='SC_errorsDataTable'>
<ErrorData id='ErrorData'>
<ERROR_CODE id='E8E72766-52EB-4C95-B8ED-B6D39A437513' type='7' rem="ErrorCode">59</ERROR_CODE>
<ERROR_TEXT id='AEBEAC34-2633-4D23-8E89-E9DBFDF031EF' type='7' rem="Error Meaning">DN is not
configured in CME</ERROR_TEXT>
<N_ERRORS id='4F1EB8D1-1ABA-4BA6-8E99-82F3988C03EC' type='4' rem="Accumulated number of
errors">2</N_ERRORS>
</ErrorData>
</SP_errorsStatistics>
</tLibClientsStatistics>
```

SIP Feature Server Statistics

SIP Server collects the following statistics related to SIP Feature Server (**sipFeatureServer**) interactions:

ID	Description
FS_STATE	The Current service state. Values: 0 is out of service, 1 is in service.
FS_QUEUE_SIZE	Current number of XS requests in a queue (requests that have not been sent to Feature Server).
FS_AVERAGE_QUEUE_TIME	Average request in queue time in milliseconds during the period of statistic's summary.
FS_CONNECTIONS	Number of connections for each URL.
Statistics for each configured URL	
FS_URL	URL of Feature Server.
FS_ACTIVE_CONNECTIONS	Current number of active connections.
FS_REQUEST_RATE	Requests rate during the period of statistic's summary (request/sec).
FS_TIMEOUTS	Number of timeouts for requests (sent or in queue) during the period of statistic's summary.
FS_400_ERRORS	Number of 400 error responses during the period of statistic's summary.
FS_404_ERRORS	Number of 404 error responses during the period of statistic's summary.
FS_4XX_ERRORS	Number of 4xx error responses during the period of statistic's summary.
FS_500_ERRORS	Number of 500 error responses during the period of statistic's summary.
FS_501_ERRORS	Number of 501 error responses during the period of statistic's summary.
FS_5XX_ERRORS	Number of 5xx error responses during the period of

ID	Description
	statistic's summary.
FS_AVERAGE_RESPONSE_LATENCY	Average response latency during the period of statistic's summary (in milliseconds).

SIP Feature Server Statistics Examples

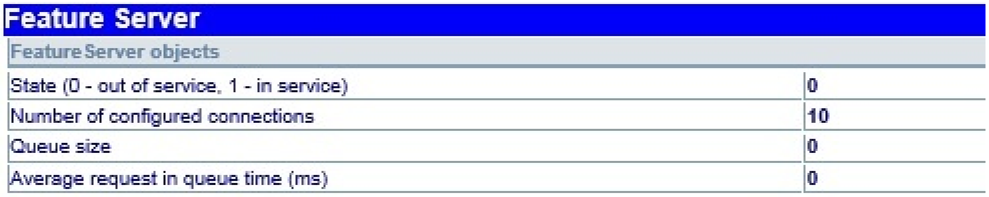
- In the HTML format:

[+] Show HTML source

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- saved from url=(0045)http://localhost:8080/server -->
<HTML><HEAD><META content="IE=5.0000" http-equiv="X-UA-Compatible">

<META http-equiv="Content-Type" content="text/html; charset=utf-8">
<META name="GENERATOR" content="MSHTML 11.00.9600.19104"></HEAD>
<BODY> <LINK href="server_files/style.css" rel="StyleSheet">
<TABLE>
  <TBODY>
  <TR>
    <TD>\\<A
  href="http://localhost:8080/server?core">sipserver</A></TD></TR></TBODY></TABLE>
<TABLE class="object">
  <CAPTION class="Caption">Feature Server</CAPTION>
  <TBODY>
  <TR>
    <TD class="SubTitle" colspan="2">FeatureServer objects</TD></TR>
  <TR>
    <TD class="Name" colspan="0">State</TD>
    <TD class="Value" colspan="0">0</TD></TR>
<TR>
    <TD class="Name" colspan="0">Number of configured connections</TD>
    <TD class="Value" colspan="0">10</TD></TR>
  <TR>
    <TD class="Name" colspan="0">Queue size</TD>
    <TD class="Value" colspan="0">0</TD></TR>
  <TR>
    <TD class="Name" colspan="0">Average request in queue time (ms)</TD>
    <TD class="Value" colspan="0">0</TD></TR>
  <TR></TR></TBODY></TABLE></BODY></HTML>
```

The code above displays these results in a browser:



Feature Server

FeatureServer objects	
State (0 - out of service, 1 - in service)	0
Number of configured connections	10
Queue size	0
Average request in queue time (ms)	0

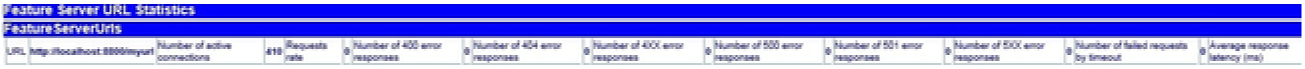
- The URL-related statistic in the HTML format:

[+] Show HTML source

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- saved from url=(0058)http://localhost:8080/server -->
<HTML><HEAD><META content="IE=5.0000" http-equiv="X-UA-Compatible">

<META http-equiv="Content-Type" content="text/html; charset=utf-8">
<META name="GENERATOR" content="MSHTML 11.00.9600.19104"></HEAD>
<BODY> <LINK href="server1_files/style.css" rel="StyleSheet">
<TABLE>
  <TBODY>
  <TR>
    <TD>\\<A
  href="http://localhost:8080/server?core">sipserver</A></TD></TR></TBODY></TABLE>
<TABLE class="object">
  <CAPTION class="Caption">Feature Server URL Statistics</CAPTION>
  <TBODY>
  <TR></TR></TBODY>
<TABLE class="object">
  <CAPTION class="Caption">FeatureServerUrls</CAPTION>
  <TBODY>
  <TR>
    <TD class="Name" colspan="0">URL</TD>
    <TD class="Value" colspan="0">http://localhost:8800/myurl</TD>
    <TD class="Name" colspan="0">Number of active connections</TD>
    <TD class="Value" colspan="0">410</TD>
    <TD class="Name" colspan="0">Requests rate</TD>
    <TD class="Value" colspan="0">0</TD>
    <TD class="Name" colspan="0">Number of error responses</TD>
    <TD class="Value" colspan="0">0</TD>
    <TD class="Name" colspan="0">Number of failed requests by timeout</TD>
    <TD class="Value" colspan="0">0</TD>
    <TD class="Name" colspan="0">Average response latency (ms)</TD>
    <TD class="Value" colspan="0">0</TD></TR></TBODY></TABLE></BODY></HTML>
```

The code above displays these results in a browser:



Feature Server URL Statistics

FeatureServerUrls

URL	http://localhost:8800/myurl	Number of active connections	410	Requests rate	0	Number of 400 error responses	0	Number of 404 error responses	0	Number of 4XX error responses	0	Number of 500 error responses	0	Number of 501 error responses	0	Number of 5XX error responses	0	Number of failed requests by timeout	0	Average response latency (ms)

- The URL-related statistic in the XML format:

[+] Show XML source

```
<sipFeatureServer id='sipFeatureServer'>
<FS_STATE id='14984276-E823-47B9-9094-93FBF41F8249' type='4' rem="State">0</FS_STATE>
<FS_QUEUE_SIZE id='0AC61903-88B8-423E-8ADE-617AA1FA6D0F' type='4' rem="Queue
size">0</FS_QUEUE_SIZE>
<FS_AVERAGE_QUEUE_TIME id='9CB891D1-31E6-4681-829C-CC690F960D00' type='4' rem="Average
request in queue time (ms)">0</FS_AVERAGE_QUEUE_TIME>
</sipFeatureServer>
<sipFeatureServerUrlStatistics id='sipFeatureServerUrlStatistics'>
<sipFeatureServerUrlStatistics id='sipFeatureServerUrlTable'>
<sipFeatureServerUrlData id='sipFeatureServerUrlData'>
<FS_URL id='2C66B966-BF01-4F05-9364-D310304376A7' type='7' rem="URL">http://localhost:8800/
myurl</FS_URL>
<FS_CONNECTIONS id='FB5DEE8B-2C65-46C7-BC25-96C8DC6A620E' type='4' rem="Number of active
connections">9</FS_CONNECTIONS>
<FS_REQUESTS_RATE id='B263BAB0-BA71-490E-A502-185239BEAE15' type='4' rem="Requests
rate">0</FS_REQUESTS_RATE>
<FS_ERRORS id='948FCA63-A0E7-4064-B26A-C5E32D572674' type='4' rem="Number of error
responses">0</FS_ERRORS>
```

```
<FS_TIMEOUTS id='E6406AB8-E600-4968-86A0-18ED85BFC516' type='4' rem="Number of failed
requests by timeout">0</FS_TIMEOUTS>
<FS_AVERAGE_RESPONSE_LATENCY id='8DBE5B2F-6DF7-4C5D-B329-94AB2026F89E' type='4' rem="Average
response latency (ms)">0</FS_AVERAGE_RESPONSE_LATENCY>
</sipFeatureServerUrlData>
</sipFeatureServerUrlStatistics>
</sipFeatureServerUrlStatistics>
```

Configuration Options

http-port

Section: **[TServer]**
Default Value: `0`
Valid Values: `0`, `1024-65535`
Changes Take Effect: After SIP Server restart

Specifies the HTTP interface port number. When set to `0`, the HTTP server is disabled. The port numbers in the range of 1 through 1023 are the system ports and must not be used.

operational-stat-timeout

Section: **[TServer]**
Default Value: `10`
Valid Values: `3-65535`
Changes Take Effect: After SIP Server restart

Specifies how often, in seconds, a local LCA is queried for system information such as CPU and memory usage. This information is then written into the SIP Server Operational Information log as defined in the SIP Server configuration.

summary-stat-timeout

Section: **[TServer]**
Default Value: `60`
Valid Values: Integer value `1-65535`
Changes Take Effect: After SIP Server restart

Specifies how often, in minutes, the summary statistics are calculated.

t-library-stats-enabled

Section: **[TServer]**
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After SIP Server restart

When set to `true`, SIP Server collects T-Library client statistics and embeds them inside HTTP monitoring statistics. When set to `false` (the default, this feature is disabled.

Feature Limitations

- The CSS style for the HTML statistics page is hardcoded.
- For the SIP Server threads statistics feature, each client connected to SIP Server must have a unique name. Two or more clients with the same name must *not* connect to SIP Server simultaneously.

Improved presentation of multiple routing attempts in historical reporting

With this enhancement, SIP Server provides additional information for historical reporting for the environment where **divert-on-ringing** is set to `false` and for call flows when a call is not answered by the first agent and must be distributed from a Routing Point multiple times.

In single-site routing scenarios, SIP Server generates EventReleased with AttributeCallState=7 (NoAnswer) for an unanswered party when the **after-routing-timeout** expires. Previously, SIP Server generated EventReleased with AttributeCallState=22 (Redirected).

To enable this feature in multi-site deployments, set the **sip-server-inter-trunk** configuration option to `true` on the Trunk DNs that are used to connect SIP Servers on different sites.

To retain the previous SIP Server behavior (prior to version 8.1.102.13), set the Application-level **enable-legacy-reporting** configuration option to `true`.

enable-legacy-reporting

Setting: **TServer** section, Application level
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: On the next call

Enables backward compatibility for reporting AttributeCallState that SIP Server distributes in EventReleased for an unanswered routing target party in single-site routing scenarios.

- If set to `true`, SIP Server distributes EventReleased with AttributeCallState=22 (Redirected).
- If set to `false`, SIP Server distributes EventReleased with AttributeCallState=7 (NoAnswer).

Feature Limitations

In multi-site deployments, this feature works only with the **route** ISCC transaction type.

Enhanced Reporting for Early ISCC Transaction Completion

Starting with release 8.1.102.09, SIP Server supports the new **transaction-state** configuration option in the **[extrouter]** section. This option enables improved historical reporting of data for multisite scenarios where a call is successfully delivered to the destination site but is not answered by the target agent. See the TSCP Release Note version 8.1.010.77 for details.

Modifying the From Header in SIP INVITE

SIP Server provides the ability to modify the From header in outgoing SIP INVITE messages. Use the following configuration options to enable this functionality, depending on your needs.

DN-Level Configuration Options

These **cpn**-controlling options are configured on an Extension DN or VoIP Service DN with **service-type**=`softswitch`.

cpn-self

Setting: DN level > **[TServer]** section
Default Value: An empty string
Valid Values: Any string
Changes Take Effect: For the next call

If configured, SIP Server replaces the User-Name in the From header of the INVITE message with the value of this option, when sending the INVITE to the device/DN where this option is specified. This option takes precedence over any other **cpn**-controlling option and the `CPNDigits` key in AttributeExtensions of a T-Library request.

cpn-dnis

Setting: DN level > **[TServer]** section
Default Value: An empty string
Valid Values: Any string containing the name of the VoIP Service DN with **service-type**=`dial-plan`
Changes Take Effect: For the next call

If configured, SIP Server replaces the User-Name in the From header of the INVITE message with the value produced by applying dial-plan rules to the call DNIS, when sending the INVITE to the device/DN where this option is specified. This option applies only to inbound calls.

cpn-digits-to-both-legs

Setting: DN level > **[TServer]** section
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: For the next call

This option applies to a TMakeCall request containing the `CPNDigits` key-value pair in AttributeExtensions. If set to `true`, SIP Server replaces the User-Name in the From header of the INVITE message with the value of the `CPNDigits`, when sending the INVITEs to a call originator and a call destination.

Important

Genesys does not recommend using the **cpn** configuration option and the options described above together on the same device.

Muting/Unmuting a Party in a Conference

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

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

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

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

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

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

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

Attribute	Value
	unmute it.

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

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

Mute State Duration

In SIP Server, TSetMuteOn is applied per-call basis. The Mute state is preserved for the duration of the call or until TSetMuteOff is applied. When a new call is created on the same DN, its Mute state is off. When a muted call is released, EventMuteOff is not needed and is not generated.

Examples:

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

Situations When a Mute Operation is Prohibited

SIP Server prohibits Mute operations in the following scenarios:

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

Feature Configuration

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

msml-mute-type

Default Value: 1
Valid Values: `1, 2`
Changes Take Effect: Immediately

Specifies the type for muting/unmuting a party in a conference. Type 1 is required to support remote mute functionality in SIP Server. Type 2 is for backward compatibility.

> **Warning**
> Use this option only when requested by Genesys Customer Care.

sip-enable-two-party-mute

Setting: **TServer** section, Application level
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

When set to `true`, this option enables muting and unmuting parties in two-way calls via a T-Library request; requires MSML to be enabled.
Note: When set to `true`, two-party conferences are not be converted to direct calls.

Upgrade Notes

If you run 8.1.101.80 or earlier versions of SIP Server, Genesys recommends the following upgrade procedure:

1. Stop the backup SIP Servers.
2. Upgrade the backup SIP Servers.
3. Promote the backup SIP Servers to primary.
4. Repeat steps 1 to 2 on the backup (formerly primary) SIP Servers.
5. After all SIP Servers in the multi-site configuration are upgraded:
   - Set the **sip-enable-call-info** option to `true`.
   - Set the **monitor-party-on-hold** option to `false`.
   - Verify that the **sip-enable-call-info-extended** option is set to `true`.

Feature Limitations

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

No-Answer Supervision: After Routing Timeout Action

Starting with SIP Server release 8.1.102.01, you can define SIP Server's default action for setting the state of an agent who was not able to answer the routed call before the **after-routing-timeout** expired. Enable this feature with the new configuration option **after-routing-timeout-action** or the AFTER_ROUTING_TIMEOUT_ACTION key in AttributeExtensions of TRouteCall.

Use the **agent-no-answer-timeout** option with the corresponding action specified by the **agent-no-answer-action** option to control direct calls to an agent.

> **Important**
>
> Using No-Answer Supervision when the **divert-on-ringing** configuration option is set to `false` does not require the value of no-answer timeout options to be smaller than the value of the **after-routing-timeout** option. The value of no-answer timeout options can be bigger than the value of the **after-routing-timeout** option.

Support in Multi-Site Deployments

Starting with SIP Server release 8.1.102.38, this feature is supported in multi-site deployments. If the original site is configured with the **divert-on-ringing** option set to `false`, but the routing destination resides at another site, this feature is supported only if SIP Server stays in the signalling path (**oosp-transfer-enabled** =`false`).

When configured, the after-routing-timeout action is performed at the SIP Server site of the call routing destination.

If **after-routing-timeout** is in progress and a caller ends the call, neither **agent-no-answer-action** nor **no-answer-action** is performed, and an agent state will not be changed.

The **after-routing-timeout-action** option configured at the site where the TRouteCall request is processed has higher priority than the **agent-no-answer-action** and **no-answer-action** parameters at the destination site.

Feature Configuration

Single-site deployments:

- To enable the feature in single-site deployments, either configure the **after-routing-timeout-action** option at the SIP Server Application level, or set (in a routing strategy) the AFTER_ROUTING_TIMEOUT_ACTION key extension in TRouteCall. The key extension setting takes priority.

Multi-site deployments:

- To enable the feature in multi-site deployments, do one of the actions described for single-site deployments. If configuring **after-routing-timeout-action**, set it at the SIP Server Application that processes TRouteCall requests.

after-routing-timeout-action

Setting: **TServer** section, Application level
Default Value: `none`
Valid Values:

- `none`—SIP Server takes no action.
- `notready`—When an agent is logged in to a routing destination that does not answer the call, SIP Server sets this agent to NotReady state.
- `logout`—When an agent is logged in to a routing destination that does not answer the call, SIP Server logs this agent out.

Changes Take Effect: For the next call

Defines SIP Server's default action if the **after-routing-timeout** expires. If **after-routing-timeout** is disabled (set to `0`), then SIP Server ignores the **after-routing-timeout-action** value.

When you set this option to a valid non-default value, it takes priority over the **agent-no-answer-action** and **no-answer-action** parameters, which are not applied to an agent logged in to a routing destination if the **after-routing-timeout** expires. In addition, none of the following parameters are applied if the **after-routing-timeout** is in progress: **agent-no-answer-overflow**, **no-answer-overflow**, or **extn-no-answer-overflow**.

after-routing-timeout

Setting: **TServer** section, Application level
Default Value: `10`
Valid Values: Any integer
Changes Take Effect: For the next call

Specifies the length of time, in seconds, that SIP Server waits before diverting the call from the Routing Point DN to the destination DN after TRouteCall was processed. If the call is not diverted before the specified number of seconds, the EventError message is issued, containing the Reference ID of the TRouteCall request.
Set the value of the option to `0` (zero) to disable this functionality.

AttributeExtensions

Key: **AFTER_ROUTING_TIMEOUT_ACTION**
Type: String

Values:

- `none`—SIP Server takes no action.
- `notready`—When an agent is logged in to a routing destination that does not answer the call, SIP Server sets this agent to NotReady state.
- `logout`—When an agent is logged in to a routing destination that does not answer the call, SIP Server logs this agent out.

Requests: TRouteCall

If set, the value of this key overrides any value set in the **after-routing-timeout-action** configuration option for the current call.

Feature Limitations

- The after-routing-timeout action is not supported at destinations where there are no agents logged in.
- The after-routing-timeout action is not supported for Shared Call Appearance or Hunt Groups.
- In case of a SIP Server switchover, the **after-routing-timeout** timer is restarted at the new primary SIP Server.

Enable Customer-on-Hold Privacy

Some countries require that a customer who is on hold must be muted to the supervisor and agent(s) who are sharing the call.
Use the option **monitor-party-on-hold** to enable or disable that behavior.

- Set to `false` to mute the customer's line while on hold.
- Set to `true` (the default) to leave the customer's line audible while on hold.

This behavior has subtleties that are explained in the examples below.

Conference Behavior

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

ON THE CALL: Customer, agent, supervisor (in Whisper mode).
ACTION: The agent puts the customer on hold.
RESULT: The customer hears music, and is muted to everyone else.

ON THE CALL: Customer, agent, supervisor (in Open mode).
ACTION: The agent puts the customer on hold.
RESULT: The customer and the supervisor can still converse.

ON THE CALL: Customer, two agents, supervisor (in Whisper mode).
ACTION: The first agent puts the customer on hold.
RESULT: The customer and the second agent can still converse.

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

Feature Configuration

1. In the **TServer** section of the SIP Server Application, configure the following options:
   - **sip-enable-call-info**—Set this option to `true`.
   - **monitor-party-on-hold**—Set this option to `false`.
   - **msml-support**—Set this option to `true`.

2. Verify that the **sip-enable-call-info-extended** is set to `true`.
3. In the **TServer** section of Trunk DNs (for all trunks between SIP Servers participating in the call flow), set the **sip-server-inter-trunk** option to `true`.

monitor-party-on-hold

Setting: **TServer** section, Application level
Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: After SIP Server restart

When this option is set to `true` (the default), the supervisor in the Whisper or in the Silent mode might be able to hear the customer if the agent has put the call on hold and there are no other active participants in the call.
When this option is set to `false`, the supervisor in the Whisper or Silent mode is not be able to hear the customer if the customer is an external party, the agent has put the call on hold, and there are no other active participants in the call.

Upgrade Notes

If you run 8.1.101.80 or earlier versions of SIP Server, Genesys recommends the following upgrade procedure:

1. Stop the backup SIP Servers.
2. Upgrade the backup SIP Servers.
3. Promote the backup SIP Servers to primary.
4. Repeat steps 1 to 2 on the backup (formerly primary) SIP Servers.
5. After all SIP Servers in the multi-site configuration are upgraded:
    - Set the **sip-enable-call-info** option to `true`.
    - Set the **monitor-party-on-hold** option to `false`.
    - Verify that the **sip-enable-call-info-extended** option is set to `true`.

Providing Origination DN Name and Location in EventRinging

SIP Server now reliably provides the origination DN name and location in EventRinging. The agent desktop can use this information to collect extended data about the originating party, such as the agent name, and present it to the destination party while the phone is ringing. In particular, Workspace Desktop Edition displays this information in the “toast” window, which notifies an agent about a new incoming call. This feature applies to all scenarios, including transfers, conferences, and call supervision in both single-site and multi-site deployments.

SIP Server adds two key-value pairs to EventRinging to implement new functionality:

- `OriginationDN`—The name of the origination DN
- `OriginationDN_location`—The name of the SIP Server switch to which the origination DN belongs

Event Examples

The value of `OriginationDN` provided in EventRinging is synchronized with the party name delivered through EventUserEvent of the LCTParty interface.

```
EventRinging
AttributeExtensions
'OriginationDN' '21001'
'OriginationDN_location' 'Home'
AttributeThisDN '7101'
AttributeOtherDN '21001'
```

In the example above, the following LCTParty EventUserEvent will be distributed to DN 7101 when the call is established:

```
EventUserEvent
AttributeExtensions
'LCTParty0' '7001'
'LCTParty0_location' 'Home'
'LCTParty1' '21001'
'LCTParty1_location' 'Home'
'LCTPartiesLength' 2
AttributeThisDN '7101'
```

Origination Party Generation Rules

The following rules apply to the generation of origination party information:

- In calls made through a Routing Point, the Origination party for the TRouteCall destination will be the party that originated the call to the Routing Point.
- In single-step transfer (SST) scenarios, the Origination party for the transfer destination will be the party

that originated the call to the transferrer. If the Origination DN of the transferrer has already been released from the call, then any other party except the transferrer will be added as `OriginationDN`.

- In supervision scenarios, the supervisor desktop will have the same origination DN as distributed for the monitored agent. In addition, if the monitored agent initiates a call, the origination DN for the supervisor will be the party present in the call instead of the monitored agent.

The table below shows the origination information (DN and location) distributed in single-site and multi-site scenarios based on the following information:

- **Home** and **East** sites are connected through ISCC.
- **Home** site has the following configuration:
  - Extensions: DN 7101, DN 7102, DN 7103
  - Routing Point: DN 5000
- **East** site has the following configuration:
  - Extension: DN 7901

Scenarios	**EventRinging Attributes and Extensions**			
AttributeThisDN	**OriginationDN**	**OriginationDN_location**	**AttributeOtherDN**	
7101 makes a call to 7102	7102	7101	Home	7101
1. 7101 makes a call to 5000 2. The call is routed to 7102	7102	7101	Home	7101
1. 7101 makes a call to 7102 2. 7102 issues a single-step transfer to 7103	7103	7101	Home	7101
1. 7101 makes a call to 7102 2. 7102 issues a single-step conference to 7103	7103	7102	Home	Not available
1. 7103 monitors 7102 2. 7101 makes a call to 7102 3. Call supervision starts	7103	7101	Home	7101
1. 7101 makes a call to 5000 2. The call is routed to 7901 with CPNDigits=100100	7901	7101	Home	100100

Scenarios	EventRinging Attributes and Extensions			
1. 7101 makes a call to 7102 2. 7102 issues a single-step conference to 5000 3. The call is routed to 7901	7901	7102	Home	confXXX/msmlXXX

Feature Configuration

Enable the Call Participant Info functionality by setting the **sip-enable-call-info** configuration option to `true` in the `TServer` section of the SIP Server Application.

Sending Outgoing INVITEs with Multipart Body

SIP Server now supports passing geo-location information formed by the routing strategy in the multi-part body of the outgoing INVITE message. The new functionality is triggered from the routing strategy by adding two key-value pairs to the AttributeExtensions: SIP_MIME_HEADERS and Geolocation:

- The **SIP_MIME_HEADERS** extension key consists of the following parameters separated by a colon (see mapping examples):
  - The name of the extension key containing an actual payload to be included in the outgoing INVITE body. The current supported extension key for this feature is `Geolocation`.
  - The content type for this payload, one of the IANA-registered MIME types. The current supported content type for this feature is `application/pidf+xml`.
- The value of the **Geolocation** extension key will be included as the body of the outgoing multipart INVITE message. No format check, no re-encoding and no other modifications to payload are made by SIP Server; the payload is included in the INVITE body as is.

SIP Server generates an outgoing INVITE message using the information provided in the two extensions described above, as specified by RFC 6442.

The feature can be triggered on any calls routed to the external number.

AttributeExtensions

Key: **SIP_MIME_HEADERS**
Type: String
Values: `Geolocation:application/pidf+xml`
Requests: TRouteCall

Passes geolocation content from TRouteCall into an outgoing INVITE message.

Key: **Geolocation**
Type: String
Requests: TRouteCall

Carries geolocation content of the body to be included into an outgoing INVITE message.

Mapping Examples

Example of TRouteCall

```
RequestRouteCall
AttributeThisDN '5002'
```

```
AttributeConnID 22660268d90ab001
AttributeOtherDN    '22002'
AttributeRouteType  1   (RouteTypeDefault)
AttributeReferenceID    10
AttributeExtensions
  ‘SIP_MIME_HEADERS’       ‘Geolocation:application/pidf+xml’
  ‘Geolocation’            ‘<?xml version="1.0" encoding="UTF-8"?>
                            <presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10" xmlns:gml="http://www.opengis.net/gml"
entity="pres:point2d@example.com">
                            <tuple id="22c0e6a14348456597c8f02b5915a29b">
                            <status>
                            <gp:geopriv>
                            <gp:location-info>
                            <gml:Point srsName="urn:ogc:def:crs:EPSG::4326"
xmlns:gml="http://www.opengis.net/gml">
                            <gml:pos>43.6198128 -70.2696997</gml:pos>
                            </gml:Point>
                            </gp:location-info>
                            </gp:geopriv>
                            </status>
                            <timestamp>2015-07-16T13:07:06Z</timestamp>
                            </tuple>
                            </presence>’
```

Example of the corresponding outgoing INVITE

```
INVITE sip:22002@192.168.73.38:63081 SIP/2.0
From: <sip:msml=5593f1ad00000001@UTE_HOME:11001>;tag=CE972381-9AD5-46EA-B8E9-43E45959890D-13
To: <sip:5002@UTE_HOME:11001>
Call-ID: 6FE4A45E-37B2-468B-B618-8A9D41F5B751-8@UTE_HOME
CSeq: 1 INVITE
Via: SIP/2.0/UDP UTE_HOME:11001;branch=z9hG4bKD0725BBB-9A0A-4A89-981C-163DBD1F47A9-16
Contact: <sip:SVC_Mediaserver@UTE_HOME:11001>
X-Genesys-CallInfo: routed
Allow: ACK, BYE, CANCEL, INFO, INVITE, MESSAGE, NOTIFY, OPTIONS, PRACK, REFER, UPDATE
Max-Forwards: 69
X-Genesys-CallUUID: UQS8MJGDDD0KD8IKDCUQC17F20000001
Session-Expires: 1800;refresher=uac
Min-SE: 90
Supported: geolocation,timer
Geolocation: cid:1430852104988
Content-Type: multipart/mixed; boundary=845F3842_73B5_48B3_AC8A_15B65DA517FA
Content-Length: 947

--845F3842_73B5_48B3_AC8A_15B65DA517FA
Content-Type: application/sdp

v=0
o=PhoneSimulator 1 1 IN IP4 192.168.73.29
s=incoming INVITE
c=IN IP4 192.168.73.29
t=0 0
m=audio 63209 RTP/AVP 0
a=rtpmap:0 PCMU/8000/1

--845F3842_73B5_48B3_AC8A_15B65DA517FA
Content-Type: application/pidf+xml
Content-ID: 1430852104988
```

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10" xmlns:gml="http://www.opengis.net/gml"
entity="pres:point2d@example.com">
<tuple id="22c0e6a14348456597c8f02b5915a29b">
<status>
<gp:geopriv>
<gp:location-info>
<gml:Point srsName="urn:ogc:def:crs:EPSG::4326" xmlns:gml="http://www.opengis.net/gml">
<gml:pos>43.6198128 -70.2696997</gml:pos>
</gml:Point>
</gp:location-info>
</gp:geopriv>
</status>
<timestamp>2015-07-16T13:07:06Z</timestamp>
</tuple>
</presence>

--845F3842_73B5_48B3_AC8A_15B65DA517FA--
```

Private Conversations During Conference

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

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

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

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

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

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

Feature Configuration

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

- `sip-enable-call-info`—Set this option to `true`.
- `sip-enable-call-info-extended`—Set this option to `true`.
- `music-listen-disconnect`—Set this option to the path of any valid audio file.

music-listen-disconnect

Default Value: `music/on_hold`
Valid Values: The path to any valid audio file
Changes Take Effect: For the next TListenDisconnect request

This option specifies the path to an audio file to be played to the temporary disconnected party from the conference.

Feature Limitations

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

Alternate Routing for Unresponsive URS/ORS

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

You can now configure multiple destinations for alternate routing.

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

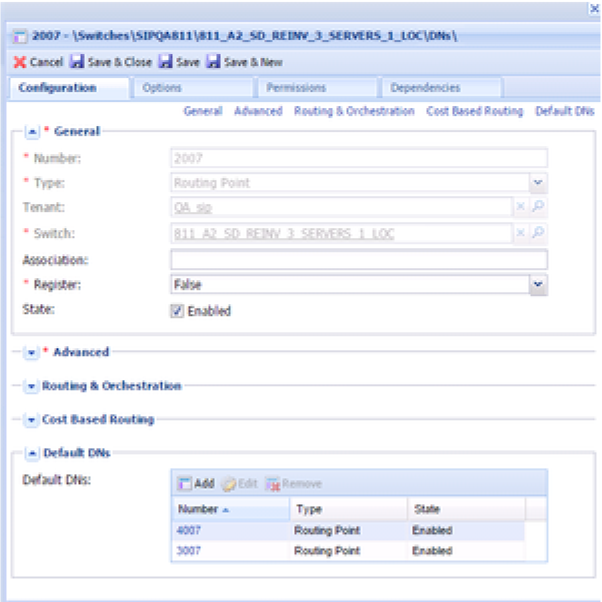
In addition, with this enhancement:

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

Feature Configuration

- Use the Application-level option `alternate-route-profile` to define a valid Routing Point DN that contains a Default DNs list. SIP Server uses that list when it encounters a Routing Point with an empty Default DNs list.
- Set the parameter `alternate-route-cof` to `true` to specify that alternate routing uses the ISCC Call Overflow feature.

> **Important**
>
> Alternate routing with attached data is enabled when alternate destinations are configured in a Default DNs list of the Routing Point DN configuration. However, if you configure the alternate destination using the `default-dn` option (on either the Application or the DN level), the alternate destination will be taken from that `default-dn` option. The alternate destination configured in `alternate-route-profile` will be ignored and not used.



2007 - \Switches\SIPQA811\811_A2_SD_REINV_3_SERVERS_1_LOC\DNs\

* General

* Number: 2007
* Type: Routing Point
Tenant: QA_sip
* Switch: 811_A2_SD_REINV_3_SERVERS_1_LOC
Association:
* Register: False
State: ☑ Enabled

Default DNs

Default DNs:

Number	Type	State
4007	Routing Point	Enabled
3007	Routing Point	Enabled

Configuration example: Default DNs list

alternate-route-profile

Setting: Application level
Section: `TServer`
Default Value: An empty string
Valid Values: A Routing Point DN with non-empty Default DNs list
Changes Take Effect: For the next default routing

Defines a Routing Point DN with a Default DNs list in its configuration. This list is used for alternate routing for all Routing Points with an empty Default DNs list.

alternate-route-cof

Setting: `ISCC Protocol Parameters` field of the Switch Access Code configuration object
Default Value: An empty string
Valid Values: `true, false`
Changes Take Effect: After SIP Server restart

When set to `true`, SIP Server uses the ISCC Call Overflow (COF) feature for alternate routing.

Feature Limitations

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

Find Me Follow Me

These configuration options support the SIP Feature Server Find Me Follow Me functionality for any 1pcc and 3pcc calls where Feature Server dial plans are applied to destinations. The feature is supported for MSML–based environments.

For this feature, SIP Server supports:

- Sequential dialing (SIP Server dials all locations sequentially)
- Parallel dialing (all locations are dialed simultaneously)
- Early media for inbound calls

Configuration Options

fmfm-prompt-file

Setting: Application level
Section: TServer
Default Value: Any empty string
Valid Values: A valid filename
Changes Take Effect: For the next call

(Optional) Specifies the filename of the confirmation prompt. Must match the path and filename in the ***MCP folder*/users** folder on the Media Control Platform server host. For example: for the file `users/fmfm-confirmation-prompt-0.wav`, set **fmfm-prompt-file** to `fmfm-confirmation-prompt`.

fmfm-confirmation-digit

Setting: Application level
Section: TServer
Default Value: `0`
Valid Values: `0-9`
Changes Take Effect: For the next call

(Optional) Specifies the digit that a caller must enter for call confirmation. This digit could be included in the prompt to be used for human recognition. If used, this digit must match the digit in the recorded prompt file. To use a different digit, you must record a new prompt and place the file in the ***MCP folder*/users** folder on the Media Control Platform server host.

fmfm-confirmation-timeout

Setting: Application level
Section: TServer
Default Value: `10`
Valid Values: A positive number
Changes Take Effect: For the next call

(Optional) Specifies the timeout value, in seconds, that SIP Server waits for a confirmation digit to be entered. Enter a number that includes playing time of the confirmation prompt and time for the confirmation digit to be entered.

Note: A call is considered abandoned when: the caller hangs up, the entered digit does not match the value of the **fmfm-confirmation-digit** option, or the call times out with no input at all.

fmfm-trunk-group

Setting: Application level
Section: TServer
Default Value: An empty string
Valid Values: A valid Trunk Group DN name
Changes Take Effect: For the next call

Specifies the Trunk Group DN where events are generated, when each destination leg connects to Media Server. Enter a Trunk Group DN name that represents Media Server. SIP Server uses that DN to play ringback, and for all outbound calls to Find Me Follow Me destinations.

msml-support

Setting: Application level
Section: TServer
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: For the next call

When set to `true`, enables the MSML support required to use Find Me Follow Me.

Feature Limitations

- It is not compatible with agent state monitored by Stat Server. If calls routed to agents have Find Me Follow Me rules applied, then the state of the DN where the agent is logged in might not be changed when the call is delivered to a non-monitored agent phone, and the next call could be delivered to the same agent.
- Early media is not supported for outgoing calls.
- Media service recovery is not supported with Find Me Follow Me. If an MSML dialog for call confirmation fails, SIP Server handles it as successful confirmation.

Informing Agents of Supervisor Presence

SIP Server now reliably informs agents of a supervisor's in-call presence in multi-site and complex single-site scenarios, when the Providing Call Participant Info feature is enabled. This feature applies to all types of call supervision—single-site, multi-site, and remote supervision. SIP Server reports the supervisor information in the EventUserEvent messages distributed to the logged-in agents. This information is primarily used by T-Library clients, such as Workspace Desktop, to display parties participating in the call.

The supervisor-related information is reported in the Extensions attribute of the relevant event using the following key-value pairs:

- `LCTSupervisor<n>`—An integer that represents the supervisor of the call, where `n` is an integer value starting from 0.
- `LCTSupervisor<n>_location`—A name of the switch to which this supervisor belongs.
- `LCTSupervisor<n>_monitoredDN`—An integer that represents the agent monitored by this supervisor.
- `LCTSupervisor<n>_mode`—Supervision mode.

A supervisor can switch between supervision modes and whenever there is change in supervision mode, SIP Server reports the change in EventPrivateInfo.

Using the EventUserEvent and EventPrivateInfo messages, Workspace Desktop could improve the customer experience by providing the accurate status of call supervision scenarios.

Note: Supervision mode is distributed only in the first EventUserEvent message generated immediately after a supervisor answers the call.

Sample Scenario

The following sample scenario describes the enhanced LCTParty interface with the supervision information:

1. Internal DNs 1001 and 1002 are provisioned on Switch A.
2. DN 1002 subscribes to monitor DN 1001 (`mute` mode, `call` scope).
3. Inbound call from DN 21001 on Switch B is routed to DN 1001.
4. Call supervision started.

SIP Server generates EventUserEvent—immediately after a supervisor answers the call—for DNs 1001@A and 1002@A with the following information:

```
EventUserEvent
  AttributeExtensions
    'LCTParty0' '21001'
    'LCTParty0_location' 'B'
    'LCTParty1' '1001'
    'LCTParty1_location' 'A'
    'LCTPartiesLength' 2
    'LCTSupervisor0' '1002'
    'LCTSupervisor0_location' 'A'
    'LCTSupervisor0_mode' 'mute'
    'LCTSupervisor0_monitoredDN' '1001'
    'LCTSupervisorLength' 1
  AttributeConnID 1
```

Feature Configuration

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

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

sip-enable-call-info

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

If set to `true`, SIP Server does the following:

- Distributes the information about call participants except their locations and the supervisor-related information (see the `sip-enable-call-info-extended` option) to logged-in agents by using the SIP NOTIFY method and EventUserEvent messages.
- Distributes an EventPrivateInfo(4024) message, with the `MonitorMode` key in AttributeExtensions, to a supervisor and agent DNs indicating that monitoring mode was changed.

If set to `false`, SIP Server does not distribute an EventPrivateInfo(4024) message when the monitoring mode changes.

sip-enable-call-info-extended

Default Value: `true`
Valid Values: `true, false`
Changes Take Effect: Immediately

This option applies only when `sip-enable-call-info` is enabled. When this option is set to `true`, SIP Server generates the supervisor information (`LCTSupervisor<n>` key-value pairs) and the location of call participants (`LCTParty<n>_location`) in EventUserEvent.

Feature Limitations

- When multi-site supervision is established with the `call` scope and if a monitored agent leaves the call, the requests submitted by the supervisor to switch between supervision modes will be rejected by SIP Server.

- When multi-site supervision is established with the `agent` scope and if consultation call supervision is started, the supervisor will not be aware of the consultation call even though the supervisor will be able to hear audio from the consultation call.

Upgrade Notes

This feature is available starting with SIP Server version 8.1.101.74. If you run SIP Server version 8.1.101.59 and later, Genesys recommends the following upgrading procedure:

1. Configure the Application-level option `sip-enable-call-info-extended` to `false` in all backup instances of SIP Servers.
2. Upgrade the backup SIP Servers.
3. Configure `sip-enable-call-info-extended` to `true` in backup SIP Servers.
4. Promote backup SIP Servers to primary.
5. Repeat steps 1 to 3 on the new backup (formerly primary) SIP Servers.

DTMF Clamping in a Conference

This feature guards a customer's sensitive credit card information from an agent and from call recording. DTMF clamping is supported in single-site and multi-site deployments. Here is how it works when activated and enabled:

1. The customer needs to enter a credit card number.
2. The agent adds IVR to the call, which bridges the customer, agent, and IVR.
3. The customer enters the requested credit card digits, but they are not recorded and the agent hears only silence.
4. The credit card number is passed to the IVR, securely.

This behavior is called *DTMF clamping*, and SIP Server supports it to comply with the Payment Card Industry Data Security Standard (PCI DSS). MCP performs DTMF clamping for selected parties in a conference, for the following DTMF transmission modes:

- RTP packets with a Named Telephone Event (NTE) payload as specified by RFC 2833
- In-band audio tones (encoded using a regular audio codec, such as G.711)
- SIP INFO packets with the content-type `application/dtmf-relay`

SIP Server uses MSML messages to inform MCP about which legs of the conference should reveal DTMF tones and which legs should suppress DTMF tones. Each leg is controlled individually. SIP Server defines the DTMF mode for each leg based on the DN type or DN-level configuration option.

In multi-site deployments, SIP Server uses the same mechanism as for Call Participant Info notifications (NOTIFY requests) to provide information about multi-site call participants. Routing Point parties are now included in these NOTIFY requests when DTMF clamping is enabled.

Activating DTMF Clamping

1. Activate DTMF clamping by setting the Application-level option **clamp-dtmf-allowed** to `true`.
2. When activated, you can enable the feature on a DN object that is configured as IVR. For this purpose, IVR can be configured as DNs of type Voice Treatment Port, Trunk, or Trunk Group:
   - If IVR is configured as a DN of the type Voice Treatment Port is added to the conference, then DTMF tones are clamped for all parties in the conference except for the Voice Treatment Port DN. No DN-level configuration is required.
   - If IVR is configured as a Trunk or Trunk Group DN, then activate DTMF clamping by setting the **clamp-dtmf-enabled** option to `true` on the corresponding Trunk or Trunk Group DN.
3. In multi-site deployments, set the Application-level option **sip-enable-call-info** option to `true`.

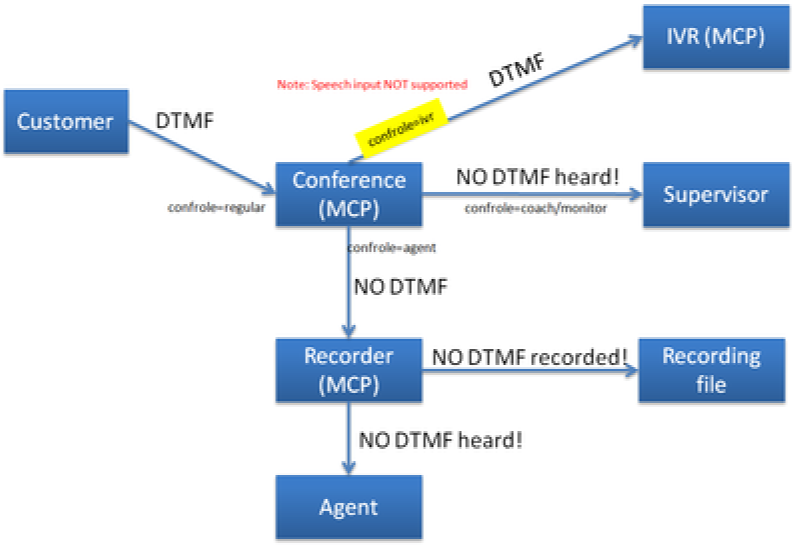
On Routing Points

SIP Server automatically activates DTMF clamping in any conference where a Routing Point is invited. No DN-level configuration is required, and only a party represented by the Routing Point is allowed to receive DTMF digits. DTMF clamping is activated regardless of the type of treatment applied at the Routing Point, and it remains active as long as the Routing Point stays in the conference.

DTMF Clamping in Recordings

PCI compliance requires that DTMF tones are not recorded when clamping is enabled. To satisfy this requirement, recording must be disabled on the caller's leg. Otherwise, DTMF digits dialed by a caller could be recorded.

Genesys recommends that you enable recording on the agent's leg as shown on the diagram below.



Configuration Options

clamp-dtmf-allowed

Setting: Application level
Section: `TServer`
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: For the next call

- When set to `true`, enables the DTMF Clamping feature.
- When set to `false`, disables this feature. This setting also preserves backward compatibility.

clamp-dtmf-enabled

Setting: DN level
Section: `TServer`
Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: For the next call

- When set to `true` on a Trunk or Trunk Group DN that is added to a conference, enables DTMF clamping for all parties except the DN where this option is configured.
- When set to `false`, disables DTMF clamping.

This option applies only to Trunk and Trunk Group DNs.

Feature Limitations

DTMF Clamping requires the Application-level option **ringing-on-route-point** to be set to `true` (the default value) when DTMF digits are collected via a treatment applied at the Routing Point.

Caller Information Delivery Content for AT&T Trunks

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

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

SIP Server communicates with GVP using SIP to pass the CID content that it receives in a multipart INVITE body in the relayed INVITE. See Passing CID content to SIP Destinations (GVP) below.

Notes:

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

Enabling CID Content Retrieval

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

Setting: DN-level
Section: TServer
Option: **`sip-accept-body`**
Default Value: An empty string
Valid Values: `cid` or empty
Changes Take Effect: For the next call

This option specifies the content type that SIP Server retrieves from the incoming INVITE with a multipart body received from an origination DN.

- If set to an empty string (default), SIP Server ignores the multipart body of an INVITE.
- If set to `cid`, SIP Server extracts the CID body from the INVITE and stores it as the caller's property.

This option...

- ...does not affect SDP.

- ...is supported only on Trunk DNs, ignored by all other DN types.

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

SIP Server previously mapped the SDP portion of a SIP message body to a T-Library event attribute; see the section "Mapping SIP Headers and SDP Messages" in Chapter 5 of the *SIP Server Deployment Guide*. Now it can also perform CID mapping to T-Library clients (URS/ORS). SIP Server sends EventRouteRequest with the CID content passed in AttributeExtensions.

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

- `extensions-1 = CID`

By default, CID content is passed to T-Library clients unchanged (UTF-8 encoding). If conversion to a local charset is enabled for SIP-to-TLib mapping (set with the `encoding` option), then this conversion is also applied to CID content.

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

Example

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

```
                    <tf:mailableVerified>true</tf:mailableVerified>
                    <tf:listType>Bus</tf:listType>
                </cl:civicAddress>
            </gp:location-info>
        </gp:geopriv>
    </dm:device>
    </presence>'
```

Passing CID Content to SIP Destinations (GVP)

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

Setting: DN-level
Section: TServer
Option: **`sip-pass-body`**
Default Value: An empty string
Valid Values: `cid`
Changes Take Effect: For the next call

This option specifies the content type that should be passed in the multipart body of the origination INVITE to this device, if that content type is received from the caller.

- If set to an empty string (default), SIP Server does not send any special content types.
- If set to `cid`, SIP Server sends the CID body to the DN.

This option...

- ...does not affect SDP.
- ...is supported on Trunk DNs, Trunk Group DNs, and VoIP Service DNs with `service-type` set to `msml` and `voicemail`.

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

Setting: Application-level
Section: TServer
Option: **`cid-enable-on-vtp`**
Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: For the next call

Use this option to simplify provisioning of the IVR that is configured through the Voice Treatment Port DNs.

- If set to `true`, SIP Server passes the CID content to the VTP DN in the initial INVITE.
- If set to `false`, SIP Server does not pass the CID content to the VTP DN in the initial INVITE.

Note: CID content default encoding is in UTF-8. No content re-encoding and no URL encoding is performed; CID is passed to SIP destinations as it is received.

Geo-location Support by GVP

Genesys software applies geo-location to multiple configuration objects. This enables Resource Manager to select the closest Media Server to the caller or agent, minimizing WAN traffic and telecom costs. SIP Server passes geo-location data to Resource Manager when Genesys Media Server is configured as:

- a Trunk DN
- a Trunk Group DN
- a Voice Treatment Port (VTP) DN
- an MSML Voice over IP Service (VOIP) DN
- a Voicemail VOIP DN

This table matches integration modes with DN types:

SIP Server-Genesys Media Server Integration Modes: Required DN Types

Integration Mode	Configure Genesys Media Server as this DN Type:
GVP Inbound mode	Trunk DN
Outbound Integration mode	Trunk Group DN and VTP DN
Voicemail Integration mode	VOIP Service DN (service-type = voicemail)
Media Server mode	VOIP Service DN (service-type = msml)

Added functionality has not changed the behavior of this feature. SIP Server puts the geo-location value of a call into the X-Genesys-geo-location header of the INVITE that it sends to Resource Manager, but only under these conditions:

- if the call's geo-location is defined as a call property.

OR

- if the call's geo-location is passed as an extension in a T-Library request (such as TApplyTreatment and TRouteCall).

If neither is true, then SIP Server does not pass the geo-location to Resource Manager.
For example: some countries require that an incoming call's geo-location be passed as a call property, and other countries do not require it. Now you can configure Media Server to account for that.

Note: For more information about setting geo-location for a call, see Framework 8.1 SIP Server Deployment Guide.

Deployment Examples

GVP Inbound mode

- Single Media Server with MCP farms located at a different geo-location
- Multiple Media Servers each with MCP farms located in different geo-locations

Outbound Integration mode

- Single Media Server handling multiple geo-location farms
- Multiple Media Servers handling multiple geo-location farms

CTI through IVR Server (IVR-centric)

- Single Media Server handling multiple media farms
- Multiple Media Servers located in multiple locations handling multiple MCP farms

Voicemail Integration mode

- Single Media Server and multiple MCP farms
- Multiple Media Servers located at different locations handling multiple MCP farms

Strict Geo-location matching scenario

See Strict Geo-location matching scenario.

Nailed-Up Connections for Agents

SIP Server supports a persistent “nailed-up connection” for agents, where it maintains an extended telephone call between SIP Server and the agent. During this time, the agent can handle multiple customer interactions without dropping the telephone connection to SIP Server.

Nailed-up connections offer a few key benefits, including:

- Minimal delay between the time an agent is selected and the audio path to the customer is established
- Improved overall reliability—the connection is already established when delivering a customer “call”, and the agent is less likely to take non-contact center calls

One typical use of nailed-up connections is for agents who use a legacy PSTN phone. These agents could be working from their home, or in a branch office that has simple PSTN connectivity. Another typical use of nailed-up connections is for agents behind a third-party PBX, when the PBX is connected to SIP Server through a gateway or simple SIP trunk.

SIP Server supports virtually all agent functionality in conjunction with nailed-up connections. The agent can make calls, receive calls, transfer calls, consult with other agents, use call supervision, and more. In addition, SIP Server’s call recording functionality is fully compatible with nailed-up connections.

Inbound calls to an agent with a nailed-up connection are delivered by default with “auto answer”—meaning the audio connects immediately. If this “auto answer” experience is not desired, then Preview Interactions should be used to provide the agent the opportunity to see call information in their agent desktop and accept or reject the call.

Nailed-up connections can be established or disconnected either by SIP Server or by the agent.

> Important
>
> In Business Continuity deployments, any DN with a statically configured contact must use **dr-forward** set to `no-agent`. In practical terms, such a DN is commonly used for a “remote agent”, often in conjunction with the nailed-up connection.

Establishing the Nailed-Up Connection

Nailed-up connections can be established by three different methods:

- SIP Server establishes the nailed-up connection on agent login or when an agent is in Ready state.
- SIP Server establishes the nailed-up connection on the first customer call.
- Agent establishes the nailed-up connection by calling into a contact center Route Point.

SIP Server Establishes the Nailed-up Connection on Agent Login or Ready state

SIP Server can establish the persistent nailed-up connection with an agent when the agent logs in, depending on the configuration:

- When **connect-nailedup-on-login**=`<Routing Point number>`, SIP Server connects the agent's endpoint with the specified Routing Point and then, after processing the TRouteCall to the predefined **gcti::park** device, SIP Server parks the agent on the **gcti::park** device establishing the persistent nailed-up connection with the agent's endpoint.
- When **connect-nailedup-on-login**=`gcti::park`, SIP Server directly parks the agent on the **gcti::park** device, establishing the persistent nailed-up connection with the agent's endpoint while processing TAgentLogin.

If a nailed-up connection is terminated for any reason, SIP Server places the agent in the NotReady state. If an agent is in the NotReady state and a nailed-up connection is not yet established, SIP Server, while processing the TAgentSetReady request, initiates a SIP call to the agent's phone with further parking on the **gcti:park** device. If the call fails, SIP generates EventError in response to TAgentSetReady; the agent remains in the NotReady state.

If the agent logs out, the nailed-up connection is dropped.

SIP Server Establishes the Nailed-up Connection on First Customer Call

SIP Server calls the agent to start a session—SIP Server sends the call to 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 Media Server, keeping the connection for the call leg to the nailed-up connection.

The basic call flow when SIP Server first calls an agent 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. Media Server applies a silent treatment while the nailed-up connection is maintained.

In Business Continuity deployments, SIP Server applies the following “Call Delivery” logic when establishing the initial call to a DN with a statically configured contact:

1. If the first SIP Server to handle the call determines an agent is locally logged in and using the DN, this SIP Server delivers the call directly to the DN.
2. Otherwise, the first SIP Server forwards the call to the second SIP Server on the alternate site, using the inter-site Trunk DN and ISCC. The second SIP Server delivers the call to the DN, regardless of whether any agent is logged in and using the DN or not.

> **Important**
>
> Carefully consider this behavior. This could result in high telephone connection charges, if, for example, DNs and data centers are distributed across different countries.

Agent Establishes the Nailed-Up Connection by Calling into a Contact Center Route Point

The agent calls the contact center to start a session—The remote TDM agent (configured for the nailed-up feature) initiates a call (1pcc) to the contact center.

The basic call flow when a remote TDM agent who is configured for the nailed-up feature is as follows:

1. A call from the remote agent arrives at the contact center on a Routing Point DN.
2. A short treatment is applied, and URS issues a TRouteCall to the predefined **gcti::park** device (RouteType=Unknown; OtherDN='gcti::park').
3. SIP Server parks the agent on the **gcti::park** device, keeping the call leg to the agent connected. At this point, the agent is considered to be nailed-up. Media Server applies a silent treatment while the nailed-up connection is maintained.

In Business Continuity deployments, each data center should have a unique routing point, which allows an agent to connect to their preferred data center based on which routing point they contact.

Disconnecting the Nailed-Up Connection

Nailed-up connections can be disconnected for several reasons:

- The agent hangs up the phone.
- A network problem between SIP Server and the phone causes the call to be dropped.
- The agent logs out (applies when SIP Server established the connection on login, or if the **drop-nailedup-on-logout** option is set to `true`).
- The agent is inactive (no changes in agent state or incoming/outgoing calls at the DN) for the specified period of time (**disconnect-nailedup-timeout**).

Feature Configuration

1. Configure the gateway.

On the Trunk DN that represents the gateway that is used to contact the remote agent, specify the following option in the **[TServer]** section:

- **refer-enabled**—Set this option to `false`.

2. Configure the agent DN.

On the ACD Position or Extension DN for the agent, specify the following options:

- **contact**—Set this option to the contact URI of the PSTN gateway/SBC or third-party PBX, depending on the agent location.
- **line-type**—Set this option to 1.
- **refer-enabled**—Set this option to `false`.
- **dual-dialog-enabled**—Set this to `false`.
- **reject-call-notready**—Set this option to `true` (recommended, not mandatory).
- **sip-cti-control**—Ensure that this option is not configured.

3. Configure the SIP Server Application.

Connection on Agent Login

To enable the persistent nailed-up connection on agent login, in the **[TServer]** section of the SIP Server Application, configure the following option:

- **connect-nailedup-on-login**

Note: If the agent logs out, the nailed-up connection will be dropped; the same behavior as if the **drop-nailedup-on-logout** is set to `true`.

Disconnection on Inactivity

To terminate the agent's nailed-up connection because of agent's inactivity, in the **[TServer]** section of the SIP Server Application, configure the following option:

- **disconnect-nailedup-timeout**

Disconnection on Agent Logout

To enable automatic disconnection of the agent from the nailed-up connection on agent logout, in the `TServer` section of the SIP Server Application, configure the following option:

- **drop-nailedup-on-logout**—Set this option to `true`.

Note: If enabled, SIP Server can only establish the nailed-up connection if the agent is logged in.

4. (Optional) Configure Business Continuity.

For Business Continuity deployments, set **dr-forward** to `no-agent`. See Basic Deployment for details.

Configuration Options

connect-nailedup-on-login

Setting: Application and DN levels
Default Value: An empty string
Valid Values: Routing Point number, `gcti::park`
Changes Take Effect: At the next agent login session

Specifies SIP Server actions when receiving a TAgentLogin request from a DN with the configured nailed-up connection, as follows:

- When this option is set to a DN of type Routing Point, SIP Server immediately establishes a nailed-up connection between an agent's endpoint and the specified Routing Point. After processing the TRouteCall request to the **gcti:park** device, SIP Server parks the agent on **gcti::park**, establishing the persistent SIP connection with the agent's endpoint.
- When this option is set to `gcti::park`, SIP Server parks the agent on the **gcti::park** device directly, establishing the persistent SIP connection with the agent's endpoint.
- When the value for this option is not specified (the default), SIP Server does not take any action.

At a DN level, this option must be set on agent Extension DN, or, if this DN is located behind the softswitch on the respective softswitch DN.

disconnect-nailedup-timeout

Setting: Application and DN levels
Default Value: `0`
Valid Values: Any positive integer
Changes Take Effect: At the next nailed-up connection

Specifies whether SIP Server terminates an agent's nailed-up connection because of the agent's inactivity. When set to a non-zero value, SIP Server waits this time interval, in seconds, before terminating the agent's nailed-up connection. When set to `0` (the default), SIP Server does not terminate the agent connection.

AttributeExtensions

The **connect-nailedup-on-login** key supports this feature. It overrides the **connect-nailedup-on-login** option setting but only for a current login session.

Key: **connect-nailedup-on-login**
Type: String
Values: Routing Point number, `gcti::park`
Requests: TAgentLogin

Specifies SIP Server actions when receiving a TAgentLogin request from a DN with the configured nailed-up connection, as follows:

- When this key is set to a Routing Point number, SIP Server immediately establishes a nailed-up connection between an agent's endpoint and the specified Routing Point. After processing the TRouteCall request to the **gcti:park** device, SIP Server parks the agent on **gcti::park**, establishing the persistent SIP connection with the agent's endpoint.

- When this key is set to `gcti::park`, SIP Server parks the agent on the **gcti::park** device directly, establishing the persistent SIP connection with the agent's endpoint.
- When this key is set to an empty value, SIP Server disables this feature for a particular agent in a current login session.

Key: **ReasonCode**
Type: String
Values: `NailedUpConnectionTerminated`
Event: EventAgentNotReady

Specifies that the nailed-up connection is terminated.

Feature Limitations

- Consultation calls for nailed-up DNs are supported in single dialog mode only.
- If an agent with the nailed-up connection is participating in the first call before it was ever parked, SIP Server cannot park this agent if the call is released before it is established. For example, if the agent with the nailed-up connection initiates a call and releases it while the call is ringing, or if the agent with the nailed-up connection completes a two-step transfer in ringing state. To avoid this, the agent should call the call center to get parked first.

Shared Call Appearance

SIP Server supports Shared Call Appearance (SCA) that enables a group of SIP phones to receive inbound calls directed to a single destination (shared line); that way, any phone from this group can answer the call, barge-in to the active call, or retrieve the call placed on hold. The shared line has sub-lines called appearances. Each shared line has one or more appearances; each appearance can handle one call at a time. The current status of each call (appearance) is displayed on each phone in the SCA group that includes outbound calls made from any phone in this group, which appear as they are placed from the same origination device.

There are several standards which enable implementation of SCA within the SIP protocol. Genesys SIP Server implemented the BroadWorks SCA standard that supports barge-in and is supported by leading phone manufacturers. Refer to your SIP phone documentation for information about SCA standards supported by your phone.

These are common scenarios where SCA can be used:

- Executive/Assistant—The call appearances on the executive's phone also appear on the assistant's phone. The assistant may answer incoming calls to the executive and then place the call on hold for the executive to pick up. The assistant can always see the state of all calls on the executive's device.
- Key System Emulation—Multiple lines are shared across two or more phones. A call answered on one phone can be put on hold and picked up on another phone. Another phone can be added/joined/bridged to an existing appearance resulting in a conference call.
- Single Line Extension—Several phones are formed in to a group to which incoming calls arrive. When one device answers, the other phone are informed. If another phone in the group goes off hook, it is immediately bridged or joined in with the call.
- Changing devices—A user is on a call on one phone and wishes to change phones and continue the call on another phone. The user places the call on hold, notes the appearance number of the call, then walks to another phone. Users are able to identify the same appearance number on the other phone, pick up the call, and continue the conversation.

Note: This feature may also be referred to as Bridged Line Appearance (BLA) or Shared Line Appearance (SLA).

User Experience

- Incoming calls to a Shared Call Appearance ring on all the associated phones.
- The status of every call is shown on all phones associated with the Shared Call Appearance.
- Calls are always associated with a “line appearance”. Incoming calls will be assigned the lowest numbered idle line appearance. All phones associated with the Shared Call Appearance should have the same number of “line appearances” configured, typically with each line appearance having a dedicated “line key” button.
- A user may seize (go off hook) a particular line appearance if it is idle by pressing the corresponding line key button. For example, pressing the second line key will seize (go off hook) the second line appearance when it is idle.

- Held calls may be retrieved by any phone associated with the Shared Call Appearance.
- An active call on a phone associated with the Shared Call Appearance may be joined at any time by another phone associated with the Shared Call Appearance. This is sometimes referred to as a “barge-in”. The parties are then conferenced together.
- Each phone associated with the Shared Call Appearance might have only one active call at a time, and other calls will be held.
- Outgoing calls from any line appearance of the Shared Call Appearance will present an outgoing caller ID with the identity of the Shared Call Appearance. (A phone could have other lines not associated with the SCA, and these are not impacted, they would present a different caller ID).

Note: According to the BroadWorks SCA standard, one DN cannot be a member of multiple shared lines. If, for example, an executive assistant needs to share lines with two executives, two independent shared lines must be configured on the assistant's phone. All of them are displayed at the screen and operable.

Sample Call Flow

A sample call flow for a Shared Call Appearance scenario is as follows:

1. Two phones are configured with a Shared Call Appearance of 7000 and all are idle. In this example, they are referred to as Phone A and B, and both are configured to show two line appearances.
2. An incoming call to 7000 rings on both phones using the first line appearance.
3. A user at Phone A answers the call. Phone B reflects the call is active on another phone on the first line appearance.
4. A second incoming call to 7000 rings on both phones on the second line appearance
5. The user at Phone A places the first call on hold. Phone B reflects the initial call is held on the first line appearance.
6. The user at Phone A answers the second call. Phone B reflects the second call is active on another phone on the second line appearance
7. The user at Phone B retrieves the held call from the first line appearance. Phone A reflects the call is now active on another phone on the first line appearance.

Shared Call Appearances are configured using two types of DNs:

- Primary shared line DN—The Address of Record (AoR), such as 7000 in the example above.
- Secondary DN—Other DN associated with the Primary shared line DN.

SCA and Other Feature Interaction

- Call Recording can be set for a particular shared line DN, Primary and/or Secondary DN.
- Call Monitoring can be set for a particular shared line DN, Primary and/or Secondary DN. However, neither Primary nor Secondary DN can monitor other DNs. If, during monitoring, a call placed on hold is retrieved by another shared line DN, the monitoring will be dropped.
- Greetings can be set for a particular shared line DN, Primary and/or Secondary DN.

- Greetings and Barge-In—A shared line user can barge-in to an established call with two parties while a greeting is in progress, after which all three parties will be connected.
- Hunt Groups—A shared line DN cannot be a member of a Hunt Group.
- Routing—Only routing to a Primary shared line DN is supported (and all phones will ring). Routing to a Secondary DN directly is not supported. A shared line DN can make a call to a Routing Point using one of the shared line appearances—the same way as for any call.
- Call Pickup—An inbound SCA call cannot be picked up by a DN rather than a shared line DN. However, if an inbound call is ringing on a regular non-shared line DN, it can be picked up by a shared line DN.
- Call Park/Retrieve—Shared line users can park a call, and the call can be retrieved from any phone (shared line or regular phones) using the Primary shared line number. There can be only one parked call per shared line at a time. Shared line users can retrieve calls that were parked by regular phones.
- Dial Plan—For inbound calls, SIP Server applies dial plans to resolve call destinations. If a destination is the Primary shared line DN, a call delivered to the SCA number is treated as a regular SCA call, i.e. is ringing on Primary and Secondary DNs. No more dial plan rules are applied after that. For outbound calls, shared line DNs dial plans are applied—for example, if a Secondary DN makes an outbound call, the dial plan configured for that Secondary DN is applied.

SCA Messaging

SCA related data is transported using the Call-Info and Line-Seize Event Packages. They are used in shared line call-related messages (INVITE, 180 Ringing, SUBSCRIBE, and so on).

SIP Server reports T-Library events separately for each Primary and each Secondary DN. No events are generated for a shared line itself.

Feature Configuration

To enable Shared Call Appearance, complete the following steps.

1. Configure a Primary shared line DN.

In the DN object:

1. Create a DN of type Extension with the number where all incoming calls will be delivered.
2. In the Annex -> `TServer` section, set the following options:
   - `shared-line`—Set this option to `true`.
   - `shared-line-capacity`—(Optional) Set this option to specify a number of shared line appearances, which limits the maximum number of simultaneous calls per shared line.
   - `authenticate-requests`—Set this option to `register` for enabling an authentication procedure on DN registration.

- `password`—Set this option to a valid password to be used for authentication of the Primary shared line DN.

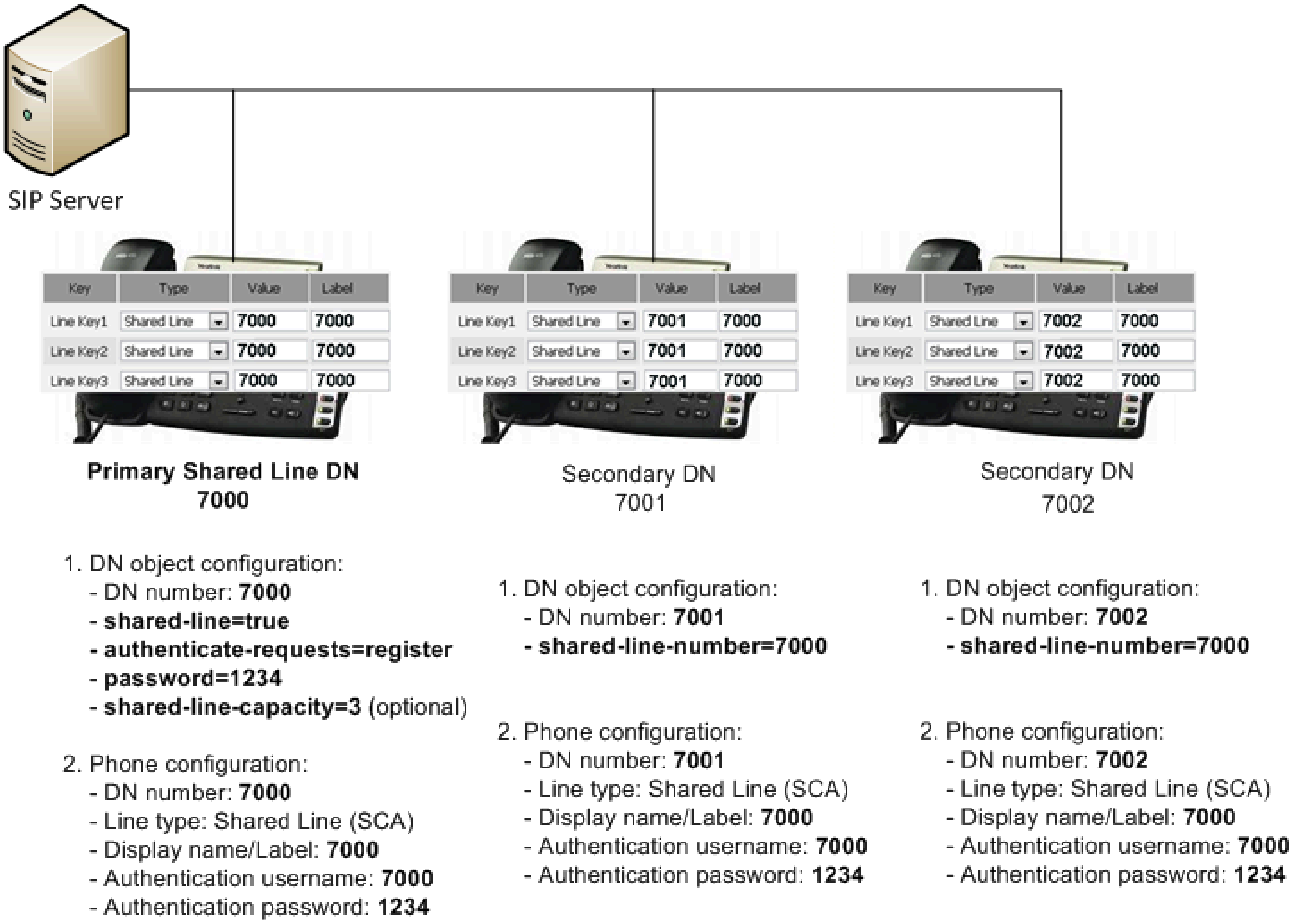
2. Configure Secondary shared line DNs.

1. In the DN object:
    - Create a DN of type Extension with the number to be used as a Secondary DN.
    - In the Annex -> `TServer` section, set the following options:
        - `shared-line-number`—Set this option to the value of the Primary DN.
2. On the SIP phone that supports SCA specify the following properties (the exact property names vary):
    - DN number—Must be set to the same value as the DN number in the DN object for a Secondary DN.
    - Line type—Must be set to Shared Line, BroadSoft SCA, or equivalent.
    - Authentication username—Must be set to the same value as the Primary DN.
    - Authentication password—Must be set to the same value as the `password` option configured for the Primary DN.
3. Repeat the above steps for each Secondary DN to be used as a shared line user.

> **Important**
>
> Starting with SIP Server version 8.1.101.75, Shared Call Appearance is supported in Business Continuity deployments. See the *SIP Server High-Availability Deployment Guide* for details.

Configuration Example

In the configuration example, the Primary shared line DN is 7000. The Secondary DNs are 7001 and 7002.



Key	Type	Value	Label
Line Key1	Shared Line	7000	7000
Line Key2	Shared Line	7000	7000
Line Key3	Shared Line	7000	7000

1. DN object configuration:
   - DN number: **7000**
   - **shared-line=true**
   - **authenticate-requests=register**
   - **password=1234**
   - **shared-line-capacity=3** (optional)

2. Phone configuration:
   - DN number: **7000**
   - Line type: Shared Line (SCA)
   - Display name/Label: **7000**
   - Authentication username: **7000**
   - Authentication password: **1234**

Key	Type	Value	Label
Line Key1	Shared Line	7001	7000
Line Key2	Shared Line	7001	7000
Line Key3	Shared Line	7001	7000

1. DN object configuration:
   - DN number: **7001**
   - **shared-line-number=7000**

2. Phone configuration:
   - DN number: **7001**
   - Line type: Shared Line (SCA)
   - Display name/Label: **7000**
   - Authentication username: **7000**
   - Authentication password: **1234**

Key	Type	Value	Label
Line Key1	Shared Line	7002	7000
Line Key2	Shared Line	7002	7000
Line Key3	Shared Line	7002	7000

1. DN object configuration:
   - DN number: **7002**
   - **shared-line-number=7000**

2. Phone configuration:
   - DN number: **7002**
   - Line type: Shared Line (SCA)
   - Display name/Label: **7000**
   - Authentication username: **7000**
   - Authentication password: **1234**

How Configuration Changes Take Effect

If a regular DN (neither Primary nor Secondary shared line DN) is changed to be a Primary or Secondary DN in the Genesys configuration, SIP Server does the following:

- Continues processing DN's existing calls as non-shared line DN calls.
- Delivers and processes new inbound calls as SCA calls. Outbound calls from this DN can be barged-in or retrieved by other shared line users.
- Does not send NOTIFY messages with appearance statuses to this DN until it subscribes to SCA statuses. To force the DN to subscribe, it must be reconfigured as a BroadWorks SCA DN. Until then, it is not able to barge-in or retrieve calls served by other shared line users.

If a Primary or Secondary DN is changed to be a non-shared line DN, SIP Server does the following:

- Continues processing of existing calls for this DN.
- Processes new inbound/outbound calls as non-shared line calls.

- Stops sending NOTIFY messages with appearances statuses to this DNs.

DN-Level Configuration Options

shared-line

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

Indicates if this DN is used as a Primary shared line number.

shared-line-capacity

Default Value: 2147483646
Valid Values: Integer in the range 1-2147483646
Changes Take Effect: For the next call

Specifies the maximum number of line appearances (or simultaneous calls) for a Primary shared line DN. These calls are distributed among shared line users and one user can handle only one call at a time. This option can be configured only for a Primary shared line DN (`shared-line=true`). The default value means that the number of simultaneous calls is (almost) unlimited.

shared-line-number

Default Value: No default value
Valid Values: Primary shared line DN
Changes Take Effect: For the next call

Specifies the Primary shared line DN to be used by the Secondary shared line DN to receive incoming calls and make outgoing calls.

Feature Limitations

- Only 1pcc operations are supported.
- One DN cannot be a member of multiple shared lines.
- Calls to Secondary DNs are not supported. Customers may choose to disable calls to Secondary DN numbers through a dial plan.
- Private Hold SCA Broadsoft functionality is not supported.
- Agent login to SCA DNs (Primary or Secondary) is not supported.
- Multi-site scenarios with the `direct-notoken` ISCC transaction type to a shared line destination DN is not supported. (No EventRinging reporting if the call is answered by a Secondary DN.)
- TRouteCall to a Secondary DN is not supported. See SCA and Other Feature Interaction.
- ICON version 8.1.400.08 or earlier might not report redirect scenarios for SCA calls correctly.
- In inbound call scenarios, no 3pcc requests can be processed before a call is answered by a shared line

user.

- SCA DNs (Primary or Secondary) cannot be located behind the softswitch.
- Semi-attended transfers and Mute transfers to the shared line are not supported.

Deleting Party From Conference

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

Feature Configuration

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

1. In the `TServer` section, set the following configuration options:
    - `sip-enable-call-info`—Set this option to `true`.
      The Call Participant Info functionality must be activated, enabling SIP Server to maintain an LCTParty list containing DNs and their locations for all parties present in the call. The LCTParty list is distributed to a T-Library client in EventUserEvent.
    - `sip-remote-del-from-conf`—Set this option to `true`.
2. In the `extrouter` section, set the `use-data-from` configuration option to `current`. This enables Party Events propagation.

sip-remote-del-from-conf

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately
Related Option: `sip-enable-call-info`

In multi-site deployments, when this option is set to `true`, SIP Server processes a TDeleteFromConference request to remove a remote party (specified in OtherDN) from a conference. The OtherDN attribute of the TDeleteFromConference request must contain the party ID received in the LCTParty list. When this option is set to `false`, this feature is disabled.

Feature Limitations

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

Agent Login and State Update on SIP Phones

This feature enables an agent to perform agent-related operations from the phone and then synchronize the phone and agent's desktop. A typical scenario involves an agent using the phone exclusively to log in/log out and set the Ready/Not Ready status without using the agent desktop application. Or, if an agent prefers using the agent desktop, then with this feature, the agent login and state will be automatically updated on the phone display. SIP Server fully synchronizes agent actions that are done using the phone or the desktop.

This functionality is implemented using two subscription packages described in the *SIP Access Side Extensions Interface* document by BroadSoft:

- Application Server Feature Event Package
- Hoteling Event Package

SIP phones that support these subscriptions enable agents to perform the following operations without using the desktop:

- Log in and log out
- Change the state to Ready, Not Ready, or AfterCallWork
- Set/synchronize the Reason code for the Not Ready state

SIP Server distinguishes subscription requests by DN (the From field) and subscription type (the Event field).

Agent Login and Authentication

There is a difference between agent desktop and phone authentication. If an agent logs in to the phone first and enters the password, the agent still must enter the password on the desktop. If an agent logs in to the desktop first and enters the password, the agent gets logged in to the phone automatically. The agent does not re-enter the password to change the agent state or to log out.

High Availability and Business Continuity Deployments

SIP Server synchronizes the agent state if a switchover occurs after the agent logs in from the phone or desktop. If the UDP transport is used, SIP Server continues sending agent state notifications to the phone through the existing subscription. If SIP is sent over TCP, it is expected that the phone should re-establish the TCP connection to SIP Server and use this connection to re-subscribe for agent state notifications from SIP Server. If the phone re-establishes the connection but does not re-subscribe, notifications are not sent. See feature limitations.

In Business Continuity deployments, phones must be configured with a single registration using the FQDN resolved in two IP addresses that correspond to SIP Server peer 1 and SIP Server peer 2. See Business Continuity deployments for details.

The Business Continuity recovery steps, if an agent uses both the desktop and phone, are as follows:

- The desktop remains in the logout state until it receives the registration request from a phone.
- The phone registers and subscribes.
- The desktop logs in automatically.

The Business Continuity recovery steps, if an agent uses only a phone, are as follows:

- The phone registers and re-subscribes.
- SIP Server sends a notification about the missing the agent-DN link and logout state.
- The phone can indicate this logout state or automatically re-log in.

Feature Configuration

- Enable the “ACD agent Availability” and “Hoteling Enhancement” features on the phone.
- If the ACD login operation on the phone requires agent authentication, provide the agent password in the Agent Login configuration object. Note that for agent authentication on the agent desktop, the desktop reads agent information from the Person configuration object.

Feature Limitations

- There is no synchronization of subscriptions between primary and backup SIP Servers. The phone must re-subscribe after the switchover.
- When an agent uses both the phone and desktop, the phone will not receive notifications after the switchover until the next SUBSCRIBE request.
- If you use phone-based agent operations, the `agent-emu-login-on-call` option must be set to `true` or not used at all.

Disabling Media Before Greeting

SIP Server provides the ability to prevent establishing a preliminary audio/video connection between a caller and an agent before greetings are applied. This feature can be applied to scenarios where for a very short time a caller and an agent could hear each other before a greeting starts playing. SIP Server is able to disable the media connection between the caller and agent for that period of time before greetings are applied.

Feature Configuration

In the `TServer` section of the SIP Server Application (or in the DN object), set the `disable-media-before-greeting` configuration option to `true`.

disable-media-before-greeting

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

Specifies whether SIP Server establishes a call in hold state if greetings are configured to be played for a caller and an agent. If set to `true`, SIP Server establishes a call in hold state (an SDP to the caller and the agent is placed on hold/inactive state). If the recording is enabled, the SDP to a recorder is also placed on hold before the greeting is played. If set to `false`, SIP Server establishes the call in active state and the media is played before the greeting.

Note: This option can be configured at both Application and DN levels. Setting at the DN level takes precedence over the Application level. If this option is set at an Application level and if a particular DN does not support this functionality, this option must be explicitly set to `false` for that DN. For a DN-level activation of this feature, this option must be set for both origination and destination DNs.

Feature Limitations

- This feature is enabled only when a call is delivered to an agent from a Routing Point.
- This feature does not apply to a greeting after a two-step conference or transfer is completed.
- This feature does not apply when TRedirectCall is used by an agent to whom the call is routed.
- This feature does not work when early media is involved in a call.
- The phones must accept an initial INVITE with the hold SDP.
- In the case of the INVITE timeout from a Media Server, there is a delay in establishing a media path between a caller and an agent.
- This feature is enabled only when MSML is used for playing greetings.
- In the case of a multi-site call, this feature is enabled only for a greeting configured using TRouteCall

extensions.

Geo-location for MSML-Based Services: Strict Matching

SIP Server supports strict geo-location matching for MSML-based services by ensuring that a call with a particular geo-location is served only by an MSML service within the same geo-location or by an MSML service within the alternate location (if configured). If a correctly geo-located MSML service is unavailable, SIP Server does not provide the required service.

SIP Headers

To prevent GVP from using a wrongly located MCP farm, SIP Server adds (in addition to the `X-Genesys-geo-location` header) the `X-Genesys-strict-location` header with a value of `enforce` to the INVITE that it sends to GVP.

Header Name	Header Value	Meaning
`X-Genesys-geo-location`	It contains the call geo-location label if an MSML service matching that label is available. It contains the overflow-location label if the intial MSML service is not available.	Instructs the Resource Manager to choose the MCP that serves a particular location.
`X-Genesys-strict-location`	`enforce`	Informs the Resource Manager that it must reject INVITE if there is no MCP avaiable that serves a particular the geo-location.

Alternate Geo-location

The alternate geo-location defined by the `overflow-location-map` option allows you to pair an alternate MSML-based service with a geo-location label. The alternate (overflow) location will be tried if the primary geo-location is not available or fails. In addition, the `overflow-location` key can be provided in AttributeExtensions of the TRouteCall and TApplyTreatment client requests. The value of the `overflow-location` key in AttributeExtensions takes precedence over the `overflow-location-map` option value. If present in the request, the `overflow-location` key enables a strict MSML-based service location for the call even if it is disabled at at Application level. If the `overflow-location` key is empty, SIP Server removes the previously assigned overflow-location (if set at an Application level) and enables MSML strict geo-location matching.

If SIP Server finds the corresponding service and sends an INVITE to that service but does not receive a positive response, SIP Server might retry an INVITE attempt once more—but only within the service that has the same geo-location or geo-location equal to the value of the `overflow-location` key.

Failure Alarms

SIP Server can generate an MSML geo-location failure alarm whenever an attempt to provide an MSML service fails because of the geo-location strict matching. The option `msml-location-alarm-timeout` specifies how often SIP Server sends that alarm (52052 code). An alarm message contains a list of failed geo-locations along with a number of failures occurred within the timeout. There is no message to reset the alarm. It is supposed to be reset by the Management Layer timeout (should be greater than the timeout defined by the option above) when SIP Server stops detecting new MSML geo-location failures.

Feature Configuration

To enable geo-location with strict matching, complete these steps:

1. Configure the SIP Server Application.

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

- `enable-strict-location-match`—Set this option according to your environment.
- (Optional) `overflow-location-map`—Set this option to *geo-location label=overflow-location label*.
- (Optional) `msml-location-alarm-timeout`
- `msml-support`—Set this option to `true`.

For more information, see the Selection Based on Geo-Location section in the Framework 8.1 SIP Server Deployment Guide.

2. Configure MSML DN(s).

See the Configuring Genesys Media Server section in the Framework 8.1 SIP Server Deployment Guide.

3. Configure a Trunk DN.

1. Create a DN of type `Trunk`.
2. In the `TServer` section, configure the following option:
   - `geo-location`—Set to the same geo-location value as any of the MSML DNs.

For more information, see the Selection Based on Geo-Location section in the Framework 8.1 SIP Server Deployment Guide.

4. (Optional) Configure a treatment block in the routing strategy.

Include the `geo-location` extension key with the same value as any of the MSML DNs.

Note: This method takes precedence over geo-location configured at the DN level.

enable-strict-location-match

Setting: Application level
Section: `[TServer]`
Default Value: No default value (empty string)
Valid Values: `msml` or `true`, `softswitch`, `trunk`, `all`
Changes Take Effect: On the next call

This option controls the SIP Server behavior in cases where an MSML service that matches a call by geo-location or overflow-location is not available, or, if during an attempt to apply a treatment, the matching service responds to the INVITE message with a SIP error, as follows:

- If this option is not present or not configured, SIP Server tries other available services for a call.
- If this option is set to `msml` (or `true`), SIP Server tries other available services that match a call by geo-location or overflow-location. If there is no match, SIP Server does not apply a service to the call with a different geo-location. (A value of `true` is supported for compatibility with previous releases of this feature.)
- If this option is set to `trunk` or `softswitch`, SIP Server tries other available trunks or softswitches that match a call by geo-location. This applies to calls directed to an external destination or DNs located behind the softswitch. If there is no match, SIP Server does not send a call to a device with a different geo-location.
- If this option is set to `all`, SIP Server applies the msml setting for calls to GVP and the trunk/softswitch setting to other cases.

Note: If the `enable-strict-location-match` option is set to `msml` or `true`, it is possible to specify an alternative geo-location using the Application level option `overflow-location-map`, or using the `overflow-location` key in AttributeExtensions of TRouteCall and TApplyTreatment client requests.

overflow-location-map

Setting: Application level
Section: `[TServer]`
Default Value: No default value (empty string)
Valid Values: Any valid string with comma-separated elements
Changes Take Effect: On the next call

This option creates an association between geo-location labels and overflow-location labels, to support strict geo-location matching.
The format of the option is: *geo-location label=overflow-location label*.

For example, in `labelA=labelB,labelC=labelD...`
`labelA` and `labelC` are geo-location labels; `labelB` and `labelD` are overflow-location labels.
If services or resources that match the call by `geo-location=labelA` are not available, SIP Server will

try services or resources that matches the call by `overflow-location=labelB`.

msml-location-alarm-timeout

Setting: Application level
Section: `[TServer]`
Default Value: `0`
Valid Values: An integer `0-65535`
Changes Take Effect: Immediately

This option enables a configurable alarm when a connection that involves an MSML DN and is restricted by geo-location, cannot be established. SIP Server maintains an alarm log of failed attempts and will display a 52052 message that lists those failures. The value of this option is the number of seconds between displays.
When the value is `0` or this option is not configured, no alarms are raised.

AttributeExtensions

Key: **`geo-location`**
Type: String
Values: A geo-location label
Requests: ApplyTreatment, TRouteCall

If TRouteCall or TApplyTreatment contains the geo-location extension, SIP Server makes this geo-location to be the most preferable on the current call. It means that each time when the switch resource is selected based on the geo-location parameter, resources with the preferred geo-location take precedence.

Key: **`overflow-location`**
Type: String
Values: An alternate geo-location label
Requests: ApplyTreatment, TRouteCall

Note: The `overflow-location` extension key applies only if the `geo-location` extension key is defined in the same request.

Feature Limitations

- The feature works only for MSML-based devices (no NETANN support).
- If an MSML service is selected for a device with `contact=::msml` on a corresponding DN (Trunk, Voice Treatment Port, Trunk Group, or Voicemail DN), the feature works properly only in the Active-Active RM deployment.
- If an MSML service is selected for a device with `contact=::msml` on a corresponding DN (Trunk, Voice Treatment Port, Trunk Group, or Voicemail DN), SIP Server does not try the alternate (overflow) location if an initial INVITE to the primary geo-location fails. This limitation does not apply to the initial INVITE selection.

Keep Alive for TCP Connections

SIP Server provides the ability to detect stale TCP connections between SIP Server and a SIP device using the TCP keep-alive mechanism. This functionality is recommended for those environments in which SIP endpoints are located behind a firewall that is configured to drop inactive TCP connections silently and without sending any notification to SIP Server. If SIP Server tries to use a stale connection to initiate a new call or to execute call control, the attempt would fail. As a result, the SIP endpoint is placed to out of service.

When the TCP keep-alive mechanism is enabled, SIP Server sends keep-alive packets for all existing SIP connections. If there is no response for a configured time interval, and if there is an active transaction for this connection, SIP Server attempts to reopen the connection immediately and re-sends the last SIP request. If the connection does not have an active transaction, then it will be reopened only when a new transaction is initiated. If an attempt to open a connection for an active transaction fails, SIP Server releases the call.

For this feature to work with TLS over TCP, the SIP Endpoint must be able to accept the connection when SIP Server attempts to reopen it.

The TCP keep-alive mechanism does not replace the active OOS check, which should be configured as usual even if the TCP keep-alive feature is enabled.

Feature Configuration

1. Configure TCP keep-alive timeouts for your operating system. You can use the following links for your reference:
    - For Windows, see http://technet.microsoft.com/en-us/library/cc957549.aspx
    - For Linux, see http://tldp.org/HOWTO/TCP-Keepalive-HOWTO/usingkeepalive.html
2. In the `TServer` section of the SIP Server Application, configure the `sip-enable-tcp-keep-alive` configuration option to enable the TCP keep-alive functionality.

sip-enable-tcp-keep-alive

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: After restart

When set to `true`, enables the TCP keep-alive mechanism for all SIP-related connections. Keep-alive timeouts are configured on the OS level.

Feature Limitation

For Voice over IP Service DNs, SIP Server will not attempt to reopen the connection within an active transaction.

Switching Between Supervision Modes

Call Supervision functionality has been enhanced to enable a supervisor to change between any supervision modes—silent monitoring, whisper coaching, or open supervisor presence—in MSML-based call monitoring, as follows:

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

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

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

This feature is supported for Assistance Supervision and Multi-site Supervision, and for both monitoring scopes `agent` and `call`.

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

Feature Configuration

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

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

Other call supervision options can be configured as required:

- cancel-monitor-on-disconnect
- default-monitor-mode
- default-monitor-scope
- intrusion-enabled

- monitor-internal-calls
- monitor-consult-calls

Configuration Options

The `sip-enable-call-info` configuration option has been modified to support this feature.

sip-enable-call-info

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

If set to `true`, SIP Server does the following:

- Distributes the information about call participants to logged-in agents by using the SIP NOTIFY method and EventUserEvent messages.
- Distributes an EventPrivateInfo(4024) message, with the `MonitorMode` key in AttributeExtensions, to a supervisor and agent DNs indicating that monitoring mode was changed.

If set to `false`, SIP Server does not distribute an EventPrivateInfo(4024) message when the monitoring mode changes.

AttributeExtensions

The `MonitorMode` key has been modified to support this feature.

Key: `MonitorMode`
Type: String
Valid Values: `mute, normal, connect, coach`
Requests: TMonitorNextCall, TRouteCall, TSetMuteOn, TSetMuteOff
Events: EventPrivateInfo

Specifies the monitoring mode as follows:

- `mute, normal`—A mute connection.
- `connect`—A three-party conference call (open supervision).
- `coach`—Only the agent can hear the supervisor (whisper coaching).

If `MonitorMode` is set to `coach` in the TSetMuteOff or TSetMuteOn request, the monitoring mode is changed to whisper coaching for the current supervision session.

Note: TSetMuteOn and TSetMuteOff support only the `coach` value.

Feature Limitation

Supervision modes cannot be changed during the remote supervision session.

VXML Support for Agent Greeting

VoiceXML (VXML) support for agent greeting functionality allows an agent to accept, reject, transfer the call (arrived from a Routing Point), or redirect the call (using TRedirectCall) to a new destination.

When the agent answers the call, SIP Server informs GVP about the VXML file and Genesys Media Server starts its processing. VXML does the following:

1. Might play the details about the call collected by URS to the agent.
2. Prompts the agent to take action for the call—to accept, reject, or transfer the call to a new destination.
3. Collects the result provided by the agent and passes it as user data to SIP Server. The VXML file can collect the result from the agent in the following ways:
    - By asking the agent to press the DTMF keys.
    - By asking the agent to say some words.

Media Server sends the user data `acceptcall` to SIP Server in the SIP INFO message which terminates VXML file processing. SIP Server receives the user data and based on that does the following:

- When the agent accepts the call, SIP Server adds the user data `acceptcall=true` to the call and connects the agent and the caller.
- When the agent rejects the call, SIP Server adds the user data `acceptcall=false` to the call and returns the call to the same Routing Point.
- When the agent transfers the call to other destination, SIP Server adds the user data `acceptcall=false` to the call and returns the call to the same Routing Point from which it is routed by URS to the other destination specified by the agent in user data.

Note: Starting with SIP Server release 8.1.101.57, this feature is supported for multi-site and Business Continuity deployments.

Message Example

This is an example of the msml `dialog.exit` message sent by the MCP at the end of the VXML when an agent rejects the call:

```
INFO sip:7101@172.24.133.150:11000 SIP/2.0
From: sip:SVC_Mediaserver@UTE_HOME:11000;tag=C5EC0EA5-84A9-4611-B864-03E9CBC10EC0-4
To: sip:7101@UTE_HOME:11000;tag=F5D150ED-A603-4532-ADF4-8D8CB1272939-36
Call-ID: B38BDC0E-C1EB-4FB3-8071-D376DAE89C0F-31@172.24.133.150
CSeq: 1 INFO
Content-Length: 255
Content-Type: application/vnd.radisys.msml+xml
Via: SIP/2.0/UDP 172.24.133.150:53329;branch=z9hG4bK1F4DFBF2-472D-4068-9747-12AF5BA6720E-3
Contact: <sip:172.24.133.150:53329>

<?xml version="1.0" encoding="UTF-8"?>
<msml version="1.1">
<event name="msml.dialog.exit" id="conn:__MSML-CONN-ID__/dialog:ivr_application">
<name>acceptcall</name>
```

```
<value>false</value>
</msml>
```

Feature Configuration

To enable VXML functionality, complete these mandatory steps:

1. Configure the SIP Server Application.

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

- `greeting-after-merge`—Set this option to `false`.
- `greeting-delay-events`—Set this option to `true`.
- `greeting-repeat-once-party`—Set this option to `agent`.
- `agent-reject-route-point`—(For multi-site deployments) Set this option to a valid Routing Point.

2. Configure the MSML DN.

1. Create a DN of type `Voice over IP Service`.
2. In the `TServer` section, configure the following options:
   - `contact`—Set to the Resource Manager IP address and port.
   - (Optional) `prefix`—Set to `msml=`. Required for conference and monitoring services only.
   - `service-type`—Set to `msml`.
   - `subscription-id`—Set to the name of tenant (used for reliability).
   - (Optional) `userdata-map-filter`—Specify a prefix (or a list of prefixes) that must match the initial characters of the key in the UserData key-value pair, which SIP Server passes to GVP when agent greeting is played.

3. Configure a routing strategy.

1. Enable personal greetings by specifying `agent-greeting` and `customer-greeting` keys in AttributeExtensions of the TRouteCall request.
2. Enable VXML functionality by setting the `agent-greeting-type` key to `vxml`. Configure the URS strategy to collect some basic details about the call and to route the call to the agent with the agent greeting VXML file. The VXML file can be in a regular file directory (file://) or on a web server (http://).

The TRouteCall request must have this VXML file along with agent greeting and customer greeting music files.

3. For multi-site deployments: Configure the URS strategy for the Routing Point to route a call to the origination Routing Point on the origination SIP Server. URS can find this information from the AttributeLastTransferOrigDN in the EventQueued message.

agent-reject-route-point

Default Value: No default value
Valid Values: Any valid Routing Point
Changes Take Effect: Immediately

Specifies the Routing Point where a call is queued if an agent rejects the call. Use this only in multi-site VXML greeting scenarios with the ISCC transaction type `route` for determining whether agent is willing to accept the call. URS can route the call from this Routing Point to the origination Routing Point at the origination SIP Server.

AttributeExtensions

Key: `agent-greeting-type`
Type: String
Valid values: `vxml`
Request: TRouteCall

Feature Limitations

- This feature is supported for MSML-based integration only.
- Customer greetings are only voice files. VXML files for customer greetings are not supported.
- This feature is not supported for greetings configured in the Agent Login object.
- The `greeting-delay-events` option does not work for the `direct-uui` ISCC transaction type. Delaying EventEstablished until the agent accepts the call is not possible in `direct-uui` multi-site call flows.

Hunt Groups in Standalone Deployments

SIP Server supports the Hunt Groups feature as a type of call coverage to distribute incoming calls to a statically configured group of extensions. The Hunt Group call distribution strategy (sequential or parallel) controls how a call is propagated to one or to all extensions within the group.

Note: Starting with version 8.1.101.49, Hunt Groups with the parallel distribution strategy (simultaneous ringing) are supported in Business Continuity deployments. See the *SIP Server 8.1 High-Availability Deployment Guide* for details.

Hunt Group members are Extension DNs or ACD Position DNs listed in the `hg-members` option. In contrast to the typical Genesys call distribution using a routing point, URS/ORS and Stat Server, the Hunt Group does not rely on or require any login. Hunt Group distribution does take into account the status of each DN, and will distribute calls only to those DNs which meet the following criteria:

- DN must be in-service
- DN must be idle (not in a call)
- DN must not have DND or Call Forwarding set on SIP Server

In a sequential call distribution, SIP Server selects one of the available Hunt Group members as a target for the call distribution. If the Hunt Group member answers the call, the call is diverted from the Hunt Group and the distribution is complete. If the call is rejected by the Hunt Group member, or not answered within a specified period of time (`hg-noanswer-timeout`), SIP Server selects the next available Hunt Group member for a call distribution. Depending on the configuration, SIP Server uses one of the following strategies for Hunt Group member selection:

- Linear hunting. SIP Server always distributes the calls to the first Hunt Group member, then to the second, to the third, and so on. Hunting stops at the last Hunt Group member.
- Circular hunting. SIP Server distributes the calls in a round-robin fashion. If a call was previously delivered to the first Hunt Group member, the next call SIP Server distributes to the second member, and so on. The succession throughout each of the Hunt Group members continues even if one of the previous members becomes available. When a list of Hunt Group members is exhausted, the hunting starts over at the first member. Hunting stops at the Hunt Group member who answered the previous call. That is, SIP Server makes only one circle through the Hunt Group member list.

In a parallel call distribution, when any Hunt Group member answers the call, the call is diverted from the Hunt Group and SIP dialogs with the non-answered Hunt Group members are dropped. This SIP Server behavior is known as divert-on-answer and works differently from the usual queue distribution enabled by the `divert-on-ringing` configuration option. For the Hunt Groups feature, you do not need to set the `divert-on-ringing` option to `false`. The `hg-type` option triggers that by default.

The call distribution is considered unsuccessful if:

- None of the Hunt Group members answers the call.
- There are no available Hunt Group members during the specified period of time (`hg-queue-timeout`).
- The number of queued calls on the Hunt Group exceeds the specified limit (`hg-queue-limit`).

The unanswered call is distributed to the default destination if it is configured; otherwise the call is released.

It is not recommended to use Extension or ACD Position as the `default-dn` destination for the Hunt Group to avoid call overflow at that DN. A Routing Point DN should be used instead.

Call forward redirection from a SIP endpoint or from an agent desktop application like Interaction Workspace will be ignored for calls distributed from a Hunt Group. Calls distributed by the Hunt Group to a member will not be diverted to the member's mailbox if there is no answer.

Feature Configuration

On a DN of type ACD Queue, specify the following configuration options in the TServer section:

- `hg-type`—Specify the type of Hunt Group algorithm that is used to deliver calls to Hunt Group members.
- `hg-members`—Specify members of the Hunt Group by listing DNs separated by a comma.
- `hg-noanswer-timeout`—Set the period of time that a call distributed to the members waits to be answered by a member.
- `hg-queue-limit`—Set a maximum number of calls that can be queued on the Hunt Group at the same time.
- `hg-queue-timeout`—Set the period of time that a call can remain in the Hunt Group (while all Hunt Group members are not reachable) before being sent to Hunt Group members.
- `hg-busy-timeout`—Set the period of time during which SIP Server will not distribute calls to the Hunt Group member’s device after it answers with an error.
- `default-dn`—(Optional) Specifies the default destination where a call is distributed if one of the following conditions occurs:
  - `hg-noanswer-timeout` expires
  - `hg-queue-timeout` expires
  - `hg-queue-limit` is exceeded
  - no correct Hunt Group members are defined
- If the Hunt Group does not have the `default-dn` option defined, SIP Server uses the Application-level `default-dn` instead.

Configuration Options

This section includes only new and modified Hunt Group configuration options. See the Framework 8.1 SIP Server Deployment Guide for a complete list of Hunt Group configuration options.

hg-type

Default Value: No default value
Valid Values: `fork, linear, circular`
Changes Take Effect: For next call distribution

Specifies the type of Hunt Group algorithm that is used to deliver calls to Hunt Group members, as follows:

- `fork`–Parallel distribution strategy (forking)
- `linear`–Sequential distribution strategy, linear hunting
- `circular`–Sequential distribution strategy, circular hunting

hg-noanswer-timeout

Default Value: `0`
Valid Values: `0–600`
Changes Take Effect: For the next call

For a parallel call distribution, this option specifies the period of time, in seconds, that a non-answered call remains in a Hunt Group before SIP Server either redirects the call to the `default-dn` destination (if configured) or rejects it.

For a sequential call distribution, this option specifies the period of time, in seconds, that SIP Server allows for a Hunt Group member to answer a call before SIP Server redirects the call to another available Hunt Group member. If the call is not answered, SIP Server either redirects the call to the `default-dn` destination (if configured) or rejects it.

If set to `0`, the call remains in ringing state until it is answered by the destination or dropped by the caller.

hg-queue-limit

Default Value: `0`
Valid Values: `0–20`
Changes Take Effect: For the next call

Specifies the maximum number of calls that can be queued at the Hunt Group. When the limit is reached, a new call is either redirected to the `default-dn` destination (if configured) or rejected.

If set to `0`, the number of calls in the queue is unlimited.

hg-queue-timeout

Default Value: `30`
Valid Values: `0–6000`
Changes Take Effect: For the next call

Specifies the period of time, in seconds, that a call is queued on the Hunt Group waiting for processing. When the time period is reached, the call is either redirected to the `default-dn` destination (if configured) or rejected. If set to `0`, the call remains in the queue until all previous call processing is finished, or the call is dropped by the caller.

Feature Limitations

- Hunt Groups are not compatible with SIP Server’s Early Media feature. A call to a Hunt Group will be immediately connected, which typically results in the caller being charged before the call is answered by an agent.

- Predictive calls (initiated by the TMakePredictiveCall request) are not supported. If a predictive call arrives at a Hunt Group, it will be rejected by the Hunt Group.
- Hunt group is not supported in deployment with IMS (double-triggering).
- A DN with nailed-up connection (`line-type=3`) must not be a member of a Hunt Group.
- DNs of the Hunt Group members must be located on the same switch as the Hunt Group.
- Calls distributed from a Hunt Group will not invoke the external Feature Server dial plan.
- It is not possible to use the Call Pickup feature to answer ringing calls for members of the Hunt Group. An attempt by a Hunt Group member to answer a call using the Call Pickup feature will be rejected.
- 1pcc semi-attended, 3pcc semi-attended, and mute transfers to a Hunt Group destination are not supported.

IMS Integration: Routed Calls as Originating or Terminating

In IP Multimedia Subsystem (IMS) deployments, SIP Server can route calls parked on a Media Server using a call-originating leg or terminating leg. Call-originating legs, compared to call-terminating legs, contain the `orig` parameter in the `Route` header of an INVITE request that SIP Server sends to the IMS.

Originating legs are subject to HSS interactions and might pass through a chain of application servers serving originating calls. Note that this processing consumes network and CPU resources.

For more information about termination and initiation INVITEs in the IMS, see the *3GPP TS 24.229 V9.0.0 (2009-06)—Technical Specification 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP)*.

Feature Configuration

The Application-level `ims-use-term-legs-for-routing` configuration option enables this feature.

ims-use-term-legs-for-routing

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

For use in IMS environments only. When set to `true`, SIP Server uses a call-terminating leg to route calls on behalf of the Routing Point after a treatment is applied. When set to `false`, SIP Server uses a call-originating leg to route calls after a treatment is applied.

Call Recording: DN Recording Override

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

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

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

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

General Rules for DN Recording Override

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

Multi-Site Call Flow Examples

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

Example 1: record=‘disable_source’

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

Example 2: record=‘disable_destination’

1. An inbound call arrives at a Routing Point at Site 1.
2. TRouteCall containing `record=disable_destination` with ISCC transaction type `route` is issued to Agent 2 with `record=true` at Site 2.

3. Call recording is disabled for Agent 2 at the destination site (Site 2).

Feature Configuration

To override enabled call recording, in the routing strategy, configure the TRouteCall request to include the `record` key with the appropriate value, as follows:

- `disable_source`—to override recording on the origination DN.
- `disable_destination`—to override recording on the destination DN.

This feature applies only if the following configurations are enabled:

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

AttributeExtensions

Key: `record`
Values: `source`, `destination`, `disable_source`, `disable_destination`
Used in: TRouteCall

- When set to `disable_source`, it overrides the `record` configuration option set on the origination DN (the DN from which a call is sent to the Routing Point).
- When set to `disable_destination`, it overrides the `record` configuration option set on the destination DN (the DN specified in AttributeOtherDN of the TRouteCall).

This `record` key continues supporting values `source` and `destination`, as follows:

- 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.
- When set to `destination`, call recording is initiated on the routing destination DN (agent), and will continue until the agent leaves the call.

Dial Plan Support for Overdial

SIP Server provides the ability for internal and inbound calls coming to a Routing Point to remove overdialed digits from DNIS when the `dnis-max-length` dial-plan rule parameter is specified. Overdialed digits are added to the `DNIS_OVER` key of AttributeExtensions in T-Library events EventQueued and EventRouteRequest.

Example 1

`dial-plan-rule: 0800xxxxxxx!=>1000; dnis-max-length=11`
Called number: `080012345670123`

Then EventQueued and EventRouteRequest will contain the following attribute values:

```
AttibuteThisDN: 1000
AttibuteDNIS: 08001234567
AttibuteExtensions 'DNIS_OVER': 0123
```

Example 2

The `dial-plan-rule` parameter does not modify the DNIS, except when the `dnis-max-length` is set.

`dial-plan-rule: 5566=>1111`
Called number: 5566

Then attributes `ThisDN` and `DNIS` in T-Library events will contain the following values:

```
AttributeThisDN: 1111
AttributeDNIS: 5566
```

EventQueued and EventRouteRequest will not contain the `DNIS_OVER` in AttributeExtensions.

Example 3

`dial-plan-rule: 5566=>1111;dnis-max-length=2`
Called number: 5566

Then attributes `ThisDN` and `DNIS` in T-Library events will contain the following values:

```
AttributeThisDN: 1111
AttributeDNIS: 55
```

EventQueued and EventRouteRequest will contain the following attribute value:

```
AttibuteExtensions 'DNIS_OVER': 66
```

Feature Limitations

- This feature applies to the Dial Plan feature configuration. Configuration with the SIP Feature Server Dial Plan is not supported.
- This feature is not supported for Outbound Predictive calls.

Feature Configuration

To enable this feature, specify the `dnis-max-length` parameter when you configure the dial-plan rule for your environment.

Parameter: `dnis-max-length`
Type: Integer
Valid values: 1-22

Description: Defines the maximum length of DNIS in the dial-plan rule. The digits that are in position past the specified length are considered overdialed and removed from DNIS. Overdialed digits are included in the `DNIS_OVER` key of AttributeExtensions in EventQueued and EventRouteRequest. Any invalid value disables this feature.

Trunk Capacity Control

SIP Server enables control of the number of outgoing and incoming calls to be handled by a specific trunk or a group of trunks in single-site deployments. SIP Server rejects the calls when trunk capacity is reached. Only traffic to and from a single SIP Server HA pair is controlled. In Business Continuity deployments, capacity control must be configured at each site.

Capacity Control of Outgoing Calls

When capacity control is enabled on a trunk, SIP Server keeps a count of every incoming and outgoing call, including every SIP or T-Library request, that it receives. When this count equals the value specified by the `capacity` configuration option, SIP Server starts rejecting only outgoing calls, generating accompanying messages depending on the call control type, as follows:

- 1pcc calls are rejected with a SIP error code specified in the `capacity-sip-error-code` configuration option.
- 3pcc calls are rejected with an EventError containing ErrorCode specified in the `capacity-tlib-error-code` configuration option.

Example

```
[TServer]
capacity=100
```

With this setting:

- If there are 50 incoming and 50 outgoing calls established through the trunk, SIP Server rejects an attempt to make an outgoing call through this trunk, but accepts incoming calls arriving to this trunk.
- If total calls are less than 100, both incoming and outgoing calls are allowed.

Capacity Control on a Group of Trunks

Trunks can be defined as one capacity group by using the `capacity-group` configuration option. When capacity control is enabled on a group of trunks, SIP Server keeps a count of every incoming and outgoing call, including every SIP or T-Library request, that it receives. When this count equals the value specified by the `capacity` configuration option, SIP Server starts rejecting only outgoing calls, generating accompanying messages depending on the call control type, as follows:

- 1pcc calls are rejected with a SIP error code specified in the `capacity-sip-error-code` configuration option.
- 3pcc calls are rejected with an EventError containing ErrorCode specified in the `capacity-tlib-error-code` configuration option.

Example

- DN of type `Trunk` with the name `Trunk1` and the following options:
  ```
  [TServer]
  ```

capacity=200

capacity-group=TrunkGroup1

prefix=8340

- DN of type `Trunk` with the name `Trunk2` and the following options:

  ```
  [TServer]
  ```

  capacity-group=TrunkGroup1

  prefix=8341

With these settings, the number of calls to the `Trunk1` and `Trunk2` will be limited to 200. When the limit is reached, SIP Server rejects attempts to make an outgoing call through these trunks, but accepts incoming calls arriving to these trunks.

Capacity Control of Incoming and Outgoing Calls

To control both incoming and outgoing calls, configure the `capacity-limit-inbound` configuration option on the same Trunk DN where the `capacity` option is defined. This capacity control mode is applicable only to DNs of type `Trunk`.

Example 1

```
[TServer]
capacity=100
capacity-limit-inbound=true
```

With these settings:

- If total calls are less than 100, incoming and outgoing calls are allowed.
- When the limit is reached (for example, 60 incoming and 40 outgoing calls), incoming and outgoing calls are rejected.

Example 2

- DN of type `Trunk` with the name `Trunk1` and the following options:

  ```
  [TServer]
  ```

  capacity=200

  capacity-limit-inbound=true

  capacity-group=TrunkGroup1

  prefix=8340

- DN of type `Trunk` with the name `Trunk2` and the following options:

  ```
  [TServer]
  ```

  capacity-group=TrunkGroup1

  prefix=8341

With these settings, the number of calls to Trunk1 and Trunk2 will be limited to 200. When the limit is reached, incoming and outgoing calls are rejected.

Feature Configuration

Select how you want to configure Trunk Capacity Control in SIP Server.

Configure capacity control of outgoing calls on a trunk.

DN Level

On a DN (type Trunk, or type Voice over IP Service with service-type=softswitch), specify the following configuration option in the TServer section:

- capacity

Application Level

Specify these options in the TServer section, as required:

- (Optional) capacity-sip-error-code
- (Optional) capacity-tlib-error-code

Configure capacity control of incoming and outgoing calls on a trunk.

DN Level

On a DN of type Trunk, specify the following configuration options in the TServer section:

- capacity
- capacity-limit-inbound

Application Level

Specify these options in the TServer section, as required:

- (Optional) capacity-sip-error-code

- (Optional) capacity-tlib-error-code

Configure capacity control of outgoing calls on a group of trunks.

DN Level

1. On a DN (type Trunk, or type Voice over IP Service with service-type=softswitch) that defines capacity and a trunk group to which capacity is applied, specify the following configuration options in the TServer section:
   - capacity
   - capacity-group
2. For all other trunks included in the same group to which capacity is applied, specify the following configuration option in the TServer section:
   - capacity-group

Application Level

Specify these options in the TServer section, as required:

- (Optional) capacity-sip-error-code
- (Optional) capacity-tlib-error-code

Configure capacity control of incoming and outgoing calls on a group of trunks.

DN Level

1. On a DN of type Trunk, specify the following configuration options in the TServer section:
   - capacity
   - capacity-limit-inbound
   - capacity-group
2. For all other trunks included in the same group to which capacity is applied, specify the following configuration option in the TServer section:
   - capacity-group

Application Level

Specify these options in the TServer section, as required:

- (Optional) capacity-sip-error-code
- (Optional) capacity-tlib-error-code

Configuration Options

DN Level

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 a group of DN objects, type Trunk or type Voice over IP Service (with service-type=softswitch) with shared capacity. All DNs configured with the same capacity-group value share the device capacity defined in the capacity option.

Note: The value of the capacity option must be defined in only one Trunk or Voice over IP Service DN in any particular capacity-group.

capacity-limit-inbound

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

When set to true, enables rejection of incoming calls if a limit on the total number of calls for a trunk (or trunks) specified by the capacity option is reached. This option must be specified on the same Trunk DN where the capacity option is defined.

Application Level

capacity-sip-error-code

Default Value: 603
Valid Values: 400–699
Changes Take Effect: Immediately

Specifies the SIP error code that SIP Server distributes in response to a rejected SIP request (incoming or outgoing) when trunk capacity is reached.

capacity-tlib-error-code

Default Value: No default value
Valid Values: Any positive integer
Changes Take Effect: Immediately

Specifies the error code that SIP Server distributes in the AttributeErrorCode of the T-Library EventError message in response to a rejected T-Library request when trunk capacity is reached. Recommended value is 282, which corresponds to the error message `No Voice Channel Available`. If the value of this option is not specified, SIP Server uses different error codes for different T-Library requests to indicate a capacity problem.

Video Blocking

SIP Server provides the ability to block video streams from SDP offers during the call negotiation/ establishment process, so video will not be played when a call is established.

With this feature enabled:

- If an SDP offer contains both audio and video media types, only the audio stream is available for the call.
- If an SDP offer contains only a video media type and no other media types are available for negotiation, the call is rejected.

The following is an example of the SDP body message containing both audio and video media types (highlighted):

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.dalycity.example.com
s=
c=IN IP4 host.dalycity.example.com
t=0 0
m=audio 49170 RTP/AVP 0 8 97
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
m=video 51372 RTP/AVP 31 32
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
```

When video blocking is enabled, SIP Server blocks (removes) the video media stream, as indicated in the following:

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.atlanta.example.com
s=
c=IN IP4 host.atlanta.example.com
t=0 0
m=audio 49170 RTP/AVP 0 8 97
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
```

Feature Limitation

SDP media stream type filtering is not performed when SIP Server is placed out of the signaling path (OOSP).

Feature Configuration

The `sip-filter-media` configuration option enables this feature. The option can be set at both Application and DN levels. The option setting at the DN level takes precedence over the Application-level setting.

Application level: sip-filter-media

Default Value: No default value
Valid Values: `video`
Changes Take Effect: Immediately

When set to `video`, SIP Server blocks video media streams in calls.

DN level: sip-filter-media

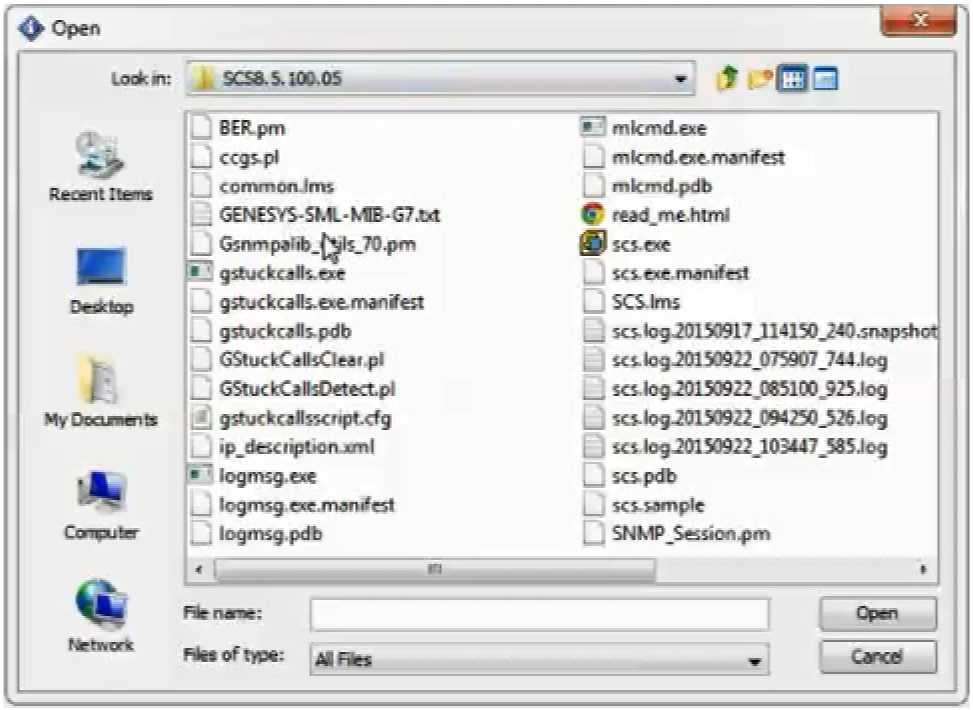
Default Value: No default value
Valid Values: `none, video`
Changes Take Effect: Immediately

When set to `video`, SIP Server blocks video media streams in calls coming to or originating from this DN. When set to none, SIP Server does not block video media streams, even if the `sip-filter-media` option is enabled at the Application level. The option can be configured on DNs of type `Extension`, `Trunk`, `Trunk Group`, or `Voice over IP Service`.

Configure SNMP Monitoring

This task, *Configuring SNMP Monitoring for T-Servers and SIP Server*, requires the skills and knowledge of a system administrator, and knowledge of the Genesys installation that is running on your site. Ideally, you or a current colleague installed the Genesys software, so you have—or can easily find—information that you need, such as the location of certain files. You also need an MIB Editor; ideally, one that you have used before.

Before you configure



Look in: SCS8.5.100.05

Files of type: All Files

Verify or perform these requirements:

- Your Genesys software installation must include a current Media Layer SNMP license. Verify with your Genesys marketing representative.
- The applications SNMP Master Agent and Solution Control Server must be installed, configured and running.

[+] SHOW HOW

The Framework 8.5 Deployment Guide describes how to deploy and configure the Solution Control Server (SCS) and the SNMP Master Agent. See links 5 and 6 in this Deployment

Summary.

- The SIP Server(s)/ TServer(s) that you wish to monitor must be provisioned with *<TServer>*`\management-port` and restarted.
  For each SIP Server and T-Server, configure `<TServer>\management-port`. See the option settings in the SIP Server 8.1 Deployment Guide.

[+] SHOW HOW

You can change T-Server common configuration options in the Options section of the Application object (for example: SNMP-MA, for the master SNMP).

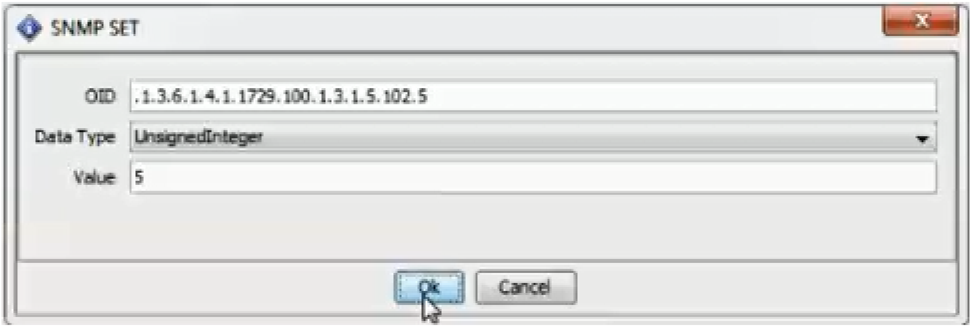
Find the Application object in this directory:

- **If you use Genesys Administrator:** Application object > Options tab > Advanced View (Options)
- **If you use Configuration Manager:** Application object > Properties dialog box > Options tab

Option: management-port
Default Value: `0`
Valid Values: `0` or any valid TCP/IP port
Changes Take Effect: after T-Server is restarted
This option specifies the TCP/IP port that management agents use to communicate with T-Server. If set to `0` (zero), this port is not used.

- The Network Management System (NMS) must be connected to the SNMP Master Agent.
  See the section *How to Activate SNMP Support* in "Chapter 7: SNMP Interface" of the Framework 8.5 Management Layer User's Guide.
- Import the file **GENESYS-SML-MIB-G7.txt** to NMS.
  Find this file in the working directory of the Solution Control Server application; the person(s) who installed Genesys for your company should know the location. This file contains Genesys-specific SML, MIB, and G71 definitions.

Summary



OID: .1.3.6.1.4.1.1729.100.1.3.1.5.102.5
Data Type: UnsignedInteger
Value: 5

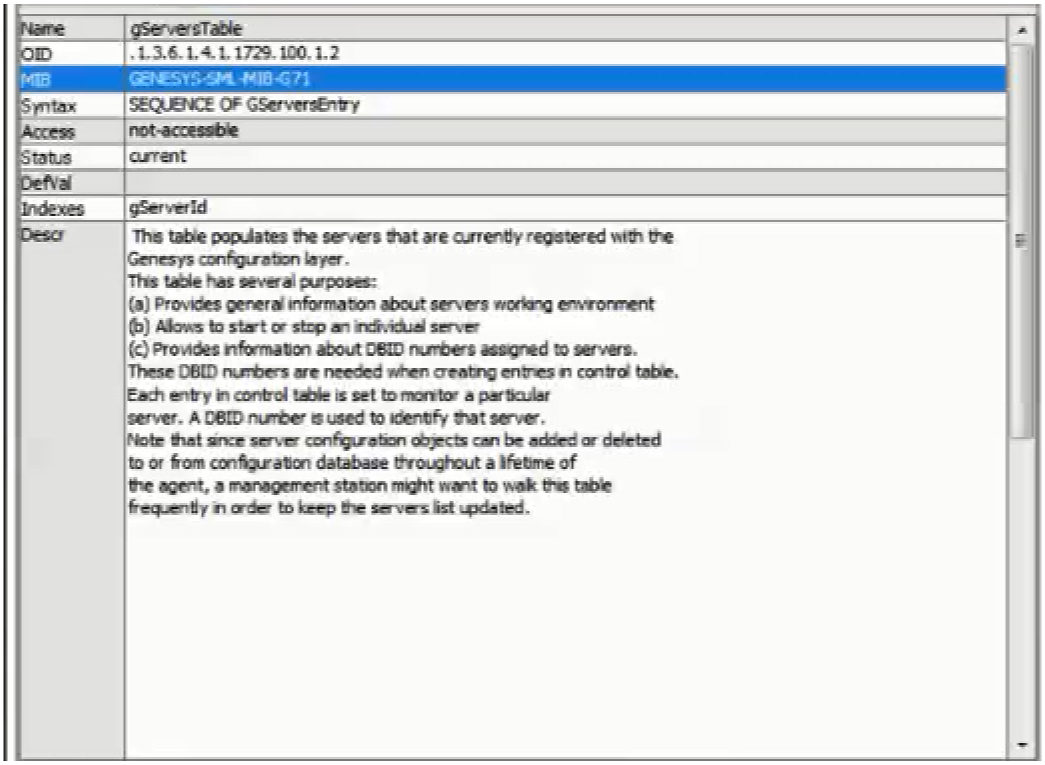
For each SIP Server / TServer...

1. Query the gServersTable and record the value of:
   gServerId (<*index*>).
2. Set the value 4 (createAndGo) for:
   Name/OID: `gsCtrlRowStatus.<`*`index`*`>.5;`
3. Set the refresh timeout. For example, 5 seconds for:
   Name/OID: `gsCtrlAutomaticRefresh.<index>.5 for tsInfoTable`

Consider the three steps above as general directions which you can execute in the most efficient way for you.
The Summary is intentionally generic because every MIB editor is different.

Consider the five steps below as an example of how one system administrator might configure SNMP Monitoring for T-Servers and SIP Server.
The real steps that you use to access SNMP data will depend on your site's NMS implementation and on your MIB editor.

Example steps to configure SNMP monitoring



Name	gServersTable
OID	.1.3.6.1.4.1.1729.100.1.2
MIB	GENESYS-SML-MIB-G71
Syntax	SEQUENCE OF GServersEntry
Access	not-accessible
Status	current
DefVal	
Indexes	gServerId
Descr	This table populates the servers that are currently registered with the Genesys configuration layer. This table has several purposes: (a) Provides general information about servers working environment (b) Allows to start or stop an individual server (c) Provides information about DBID numbers assigned to servers. These DBID numbers are needed when creating entries in control table. Each entry in control table is set to monitor a particular server. A DBID number is used to identify that server. Note that since server configuration objects can be added or deleted to or from configuration database throughout a lifetime of the agent, a management station might want to walk this table frequently in order to keep the servers list updated.

Your actual steps will be similar:

1. With the MIB Editor, **Load** the imported file GENESYS-SML-MIB-G7.txt.
2. Inside GENESYS-SML-MIB-G7.txt, find and read the DESCRIPTION for gServerControlTable and gsCtrlRowStatus.
   The description contains information about a table's contents, purpose, and behavior—and how to

request data from NMS.

[+] SHOW TABLE DESCRIPTION

```
gServerControlTable OBJECT-TYPE
 . . .
 DESCRIPTION
```

"Control table containing a set of parameters to set up and control data collection from Genesys server(s). This table facilitates the monitoring of multiple Genesys servers.

The following data tables defined in this MIB module are controlled by the gsControlTable:

- gsLogTable
- gsInfoTable
- gsClientTable
- gsPollingTable
- tsInfoTable
- tsCallTable
- tsDtaTable
- tsLinkTable
- tsCallFilterTable
- tsCallInfoTable
- tsLinkStatsTable

Entries in the above tables are created on behalf of an entry in the gServerControlTable. If a control row is destroyed, then corresponding row in the respective data table is destroyed too.

Some tables defined in this MIB module may just be used to perform some management function on a server (e.g.,gsPollingTable). When a management station creates a row in the control table for such a table, the agent will create a row in the corresponding table thus making it ready to be used to perform a management function for a selected server. This way, we are allowed to create multiple instances of a management object that can be used to perform a management function for multiple servers.

For example, in case of gsPollingTable a manager will be able to perform reconfig command for multiple Stat Servers."

3. ~~Get the value of gServerId(s) of SIP Server(s) / TServer(s) from the gServersTable.~~

[+] SHOW HOW

Scan the Table view for T-Servers or SIP Servers (in the column gServerType) that have "start" in the gServerCommandLine column. These should appear in the topmost rows of the Table View.

In the partial screen shot above, Line 2 reads "T-Server" in the gServerType column and "start" in the gServerCommandLine column. So that line contains the information that you need.

The T-Server in this row of data has an identifying number that appears in both the gServerId column and the Index column. You will need that value later.

In the example, that value is: 102.

4. Then set the value 4 (createAndGo) for Name/OID: `gsCtrlRowStatus.<`*`index`*`>.5`;
   ...where <index> is the value of `gServerId` from the step above, to initiate the data collection for `tsInfoTable`.

   Set other tables in the same way. For example: gsCtrlRowStatus.<index>.6 for tsCallTable(6).

   ### [+] SHOW HOW

   Go to this directory: `MIB Tree > iso.org.dod.internet > private > enterprises > genesys > servers > genericServer > gsCleanupTimeout > gServerControlTable > gServersEntry > gsCtrlRowStatus`

   Select `Set` from the Operations drop-down. In the SNMP Set dialog box, find the OID field and append the number in it with this: ".102.5" (102 is the gServerId for the T-Server that you found in the previous step.

   In the value field, enter 4 (for and ".4" (for createAndGo).

5. Set the refresh timeout to 5 seconds for Name/OID: `gsCtrlAutomaticRefresh.<`*`index`*`>.5` for `tsInfoTable`

   Use a similar method to set other tables, for example: `gsCtrlAutomaticRefresh.<`*`index`*`>.6` for tsCallTable(6).

New or Updated Configuration Options

The following configuration options have been either newly added or updated after the SIP Server Deployment Guide was last published.

> ### Important
> The SIP Server Deployment Guide PDF is no longer updated with information pertaining to recent changes. Information on recently released features, other modifications, and new or updated configuration options are available in this supplement. If you are looking for specific information, please refer to both the documents, first the SIP Server Deployment Guide and then this Supplement to SIP Server Deployment Guide.

sip-schedule-record-on-hold

Setting: **[TServer]** section, Application level or DN level
Default Value: `true`
Valid Values: `true`, `false`
Changes Take Effect: On the next call

Introduced in version 8.1.104.73, this option is used to avoid scheduling a recording for the call hold party if there is any other record-enabled party on the call. If no other record-enabled party is on the call, the recording will be scheduled once the call hold party retrieves the call. Setting the option to `false` will prevent automatic retrieving of the party on hold when creating a recording leg. The default value is true and retains the existing behaviour.

enable-rp-to-rp-dial-plan

Setting: **[TServer]** section, Application level
Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: On the next call

Introduced in version 8.1.104.69, this option is used to apply dial-plan rules to a call moved from one Routing Point (RP) or external RP to another RP. If set to `true`, SIP Server applies dial-plan rules to a call moved from one routing point to another. If set to `false`, dial-plan rules are not applied and the call is just sent to another routing point. For multi-site ISCC calls, additionally, the **enable-iscc-dial-plan** option must be set to `true`. For regular calls, additionally, the **rp-use-dial-plan** option must be set to `full` or `partial`.

disable-tconf-agent-logins-loading

Setting: **[TServer]** section, Application level
Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: After SIP server restart

Introduced in version 8.1.104.67, this option controls processing of agent login objects by the **tconf** library. If set to `true`, SIP Server, while deployed in SIP Cluster mode, will eliminate processing of agent login objects by the **tconf** library. This allows to speed up initialization of the SIP Cluster node even if the number of configured agent logins exceeds 100,000.

3pcc-requires-agent-session

Setting: **[TServer]** section, Application level
Default Value: `false`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

Introduced in version 8.1.104.66, use this option to enable enhanced restriction of 3PCC requests. When set to `true`, SIP Server only accepts 3PCC requests from a client associated with an agent login session. Requests from other clients are rejected with `EventError error code: 118 (Requested service unavailable) 'Access restricted'`. When set to `false`, SIP Server does not restrict 3PCC requests based on an agent login association. This option is only applicable for requests with **AttributeThisDN** of types **Extension** and **ACD Position**.

The enhanced restriction is applicable only to the following requests:

- RequestAgentLogout
- RequestAgentReady
- RequestAgentNotReady
- RequestSetDNDOn
- RequestSetDNDOff
- RequestMakeCall
- RequestAnswerCall
- RequestHoldCall
- RequestRetrieveCall
- RequestInitiateConference
- RequestCompleteConference
- RequestDeleteFromConference
- RequestInitiateTransfer

- RequestMuteTransfer
- RequestSingleStepTransfer
- RequestCompleteTransfer
- RequestAlternateCall
- RequestReconnectCall
- RequestCallForwardSet
- RequestCallForwardCancel
- RequestReleaseCall
- RequestSendDTMF
- RequestRedirectCall
- RequestListenDisconnect
- RequestListenReconnect
- RequestClearCall
- RequestSingleStepConference
- RequestMonitorNextCall
- RequestCancelMonitoring
- RequestSetMuteOn
- RequestSetMuteOff

dr-back-in-service-on-invite

Setting: **[TServer]** section, Application level
Default Value: `true`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

Introduced in version 8.1.104.61, use this option to specify if a non-emergency DN without an active registration, in DR mode, is set to `Back-In-Service` on the INVITE initiating a first-party call-control (1pcc) call. If the option is set to `false`, a non-emergency DN without an active registration is not set to `Back-In-Service` on the INVITE initiating a first-party call-control. Previously, setting such devices to `Back-In-Service` allowed agents to login and set their DN as ready without having their device ready for voice calls resulting in `EventError (Ivalid Called Dn)` responses to any attempt to route calls to such DNs.

partyadded-def-callstate-conf

Setting: **[TServer]** section, Application level

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

Introduced in version 8.1.104.40, use this option to ensure that SIP Server distributes the **EventPartyAdded** event with **AttributeCallState** set to 2 in a multi-site single step conference scenario. If the option is not found or set to `false`, SIP Server distributes the **EventPartyAdded** event with **AttributeCallState** set to `0` instead of 2.

reuse-tls-conn

Setting: **[TServer]** section, Application level and DN level
Default Value: `true`
Valid Values: `true`, `false`
Changes Take Effect: On the next call

Introduced in version 8.1.104.27, use this option is used to specify whether SIP Server reuses the existing TLS transport for sending SIP requests. If set to `false`, SIP Server opens a new TLS connection to the SIP request destination. If set to `true`, SIP Server reuses the existing TLS transport for sending SIP requests.

Note: The option is supported at the DN level too, starting with version 8.1.104.47.