SIP Endpoint Simulator Deployment Guide

SIP Server 8.5.0

12/31/2021

Table of Contents

Genesys SIP Endpoint Simulator Deployment Guide

The Genesys SIP Endpoint Simulator is a combination tool designed for testing in any SIP Server environment, including any SIP application and any Genesys T-Server. It provides the following major functionality:

- **Simulated SIP endpoint.** The tool supports both SIP signaling and RTP streams. The outgoing RTP stream is generated according to the configuration and can be analyzed on the receiving end.
- **Configurable SIP behavior.** The tool can be configured to control both the SIP message flow and content of particular messages (custom SIP scripts and custom SIP headers).
- **Automatic verification of voice path.** The Tone Label Detector (TLDR) functionality can be used to verify that the RTP audio stream works properly. Twenty six "tone labels" from A to Z (or with custom names assigned to them) can be generated at any simulated endpoint (or simulated VoIP service) and detected at the receiving endpoint, including any combination of tones as a result of conferences.
- **Genesys T-Library GUI client.** The tool provides convenient visualization, even for most complex call scenarios with multiple consultation calls, conferences, and so on. The most used third-party call control (3pcc) is also included via context-sensitive menus (although to make the interface more user-friendly, only a limited number of T-Library requests are supported).
- **Integration with QAART**. This is done with a telnet-based interface and includes the ability to control first-party call control functionality from a QAART (Genesys automated testing tool) script, to query a call and TLDR status, and to receive full SIP history.

Find the information you need from the topics below.

Deploying SIP Endpoint Simulator	Using SIP Endpoint Simulator
Installation and configuration	User interface elements
Configuration file and options	Starting and stopping

Deploying SIP Endpoint Simulator

Complete these steps to deploy SIP Endpoint Simulator:

1. Install SIP Endpoint Simulator:
    - On Windows: In the directory to which the installation package was copied, locate and double-click **Setup.exe**.
    - On Linux: In the directory to which the SIP Endpoint Simulator installation package was copied, locate a shell script called **install.sh**. See Installation package for Linux below to learn about executable files.
2. Configure SIP Endpoint Simulator: In the directory to which the installation package was copied, locate and open the sample configuration file (**esttt.conf**) and set the configuration options depending on how SIP Endpoint Simulator will be used:
    - If Simulator is used as a T-Library client for 3pcc control on some DNs, the configuration must include at least one **[site:]** section containing:
        - The **server** option referring to the particular SIP Server
        - A set of DNs to be monitored/controlled via T-Library
    - If Simulator is used for simulating SIP endpoints, the configuration must include:
        - **sip-port** and **rtp-ports** options in section **[TcCM]**
        - At least one **[site:]** section containing:
            - **sip-proxy** option referring to the particular SIP proxy (either SIP Server or a SIP trunk)
            - A set of SIP devices to be simulated
    - If Simulator is used with SIP Server, one section can be combined for both T-Library and SIP functionality and the configuration must include:
        - **sip-port** and **rtp-ports** options in section **[TcCM]**
        - SIP Server site section must contain:
            - **server** option describing the T-Library connection to a particular SIP Server
            - **sip-proxy** option describing the SIP connection to that SIP Server
            - A set of DNs that are both monitored/controlled via T-Library and simulated as SIP devices
    - In addition, whenever a SIP simulated device is present, the configuration must include at least one SIP configuration section that describes DN-level SIP options.

Installation package for Linux

The SIP Endpoint Simulator installation package for Linux includes two binary executable files:

- The regular **esttt_64** file with GUI support is dynamically linked against GTK+ 2.x runtime and requires that package (and all its dependencies) are installed.

- The restricted **estng_64** file can only work in no-GUI mode (so it can be used only for 3pcc operations, or when controlled by other application or script via telnet-based control connection), but has only dependency on basic X-Server shared libraries. For the deployment on a Linux server that does not have X-Server components installed, a copy of these shared libraries is included in the **lib** folder, and the script **run-estng.sh** is provided for updating LD_LIBRARY_PATH to load libraries from that folder.

Except for GUI support, the configuration and functionality of these two binaries are identical.

Starting and Stopping SIP Endpoint Simulator

To start SIP Endpoint Simulator, go into the installation directory, locate and then double-click **esttt.exe**. If the configuration file is renamed from its default name, specify the file name as an argument of the command. For example: **esttt.exe filename.conf**.

To stop SIP Endpoint Simulator, close the UI window or select **Exit** from the File menu.

Using SIP Endpoint Simulator

To use SIP Endpoint Simulator:

1. Right-click the mouse button to select an object (site, DN, and so on) and open the menu with commands available for the selected object.
2. Select the command from the menu and enter the parameters required for the selected command.
3. Click OK to run the command.

User Interface Elements

The main window of the SIP Endpoint Simulator consists of two panes—device view (left) and call view (right)—that represent the same current state information.

The *device view* pane contains the following elements:

- Sites—listed in order of appearance in the configuration file.
- DNs—listed in order of appearance in the configuration file, with SIP sections (and some in-line options) shown in parentheses.
- (optional) SIP dialog groups for trunks and VoIP services.
- T-Library calls (phone-based icons) and SIP dialogs (colored dots) that are active on the DN (in order of creation):
    - T-Library calls are named by the Connection ID.
    - The name of the SIP dialog consists of the internal dialog ID, TLDR result (if applicable), selected audio codec, abbreviated SIP Call-ID, and a reference to the peer's dialog ID (if applicable).

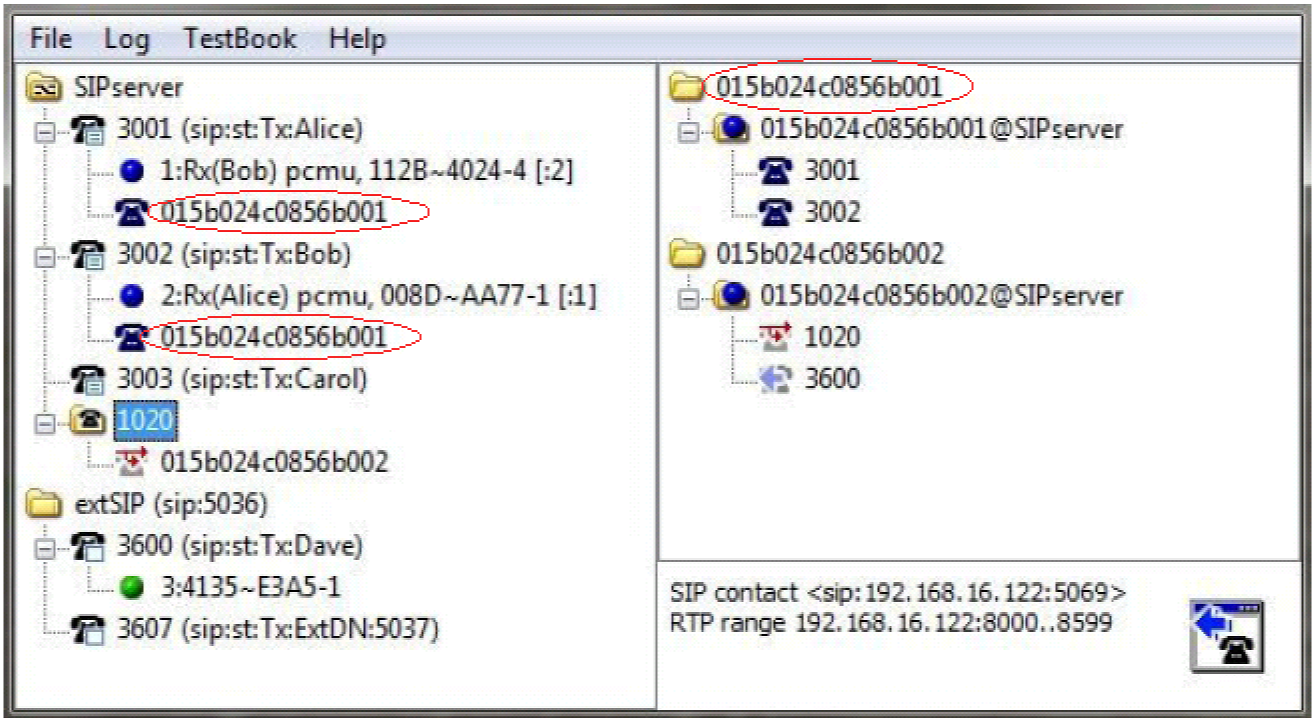
Generally, it is impossible to match a SIP dialog with a corresponding T-Library call (especially when the **dual-dialog-enabled=false** option is used in SIP Server), so SIP Endpoint Simulator does not make an attempt.

The *call view* pane contains the following elements:

- A call thread that is the collection of related calls (main/consultation for a single site, linked calls in multi-site deployments).
- A local call with a label consisting of the Connection ID and the site name.
- A party on that call with the DN name label. Except for the label, this object is essentially the same as "T-Library call" in the device view.

For all elements, the icon usually reflects the current state of that element (although for multiple independent sub-states not all info is visible). You can access the Icons Help from the Help menu.

In the screenshot below, you can see two call threads, each with a single call. The first call—ConnectionID ends with b001—has two parties (3001 and 3002) in the Connected state. The device view pane displays corresponding SIP dialogs connected to each other and the voice path verified (3001 generates the “Alice” tone and hears “Bob” and vice versa). The second call is an inbound call from external DN 3600 (which is not registered on SIP Server, so no T-Library party there) to Routing Point 1020 (no SIP dialog present), in the Routing state.



Manual Controls and Commands

For most elements in both panes, left-clicking the mouse selects the object and right-clicking opens the context-sensitive menu. Operations included in those menus are briefly described in the next sections. As a general rule:

- Operations for SIP dialogs (colored-dot icons) include SIP commands (1pcc).
- Operations for T-Library parties (phone-based icons) include T-Library commands (3pcc).
- Operations for a DN (second-level elements) include both 1pcc and 3pcc commands.

Per standard convention, menu items with the name ending in ellipsis ('...') open a dialog box requesting operation parameters and perform the action only after a value is entered and the Ok button or Enter key are pressed (Cancel button or Esc key aborts the operation). The dialog box prompt shows a very brief description of the value format for easy reference. See below for more complete description of parameters for each operation.

For most operation parameters entered in GUI dialog boxes, SIP Endpoint Simulator keeps a handful of last values for each dialog (but no more than 1000 total entries). That history is persistent, saved between sessions in the `esttt.history` file in the folder where the Simulator executable is located.

Site-level commands

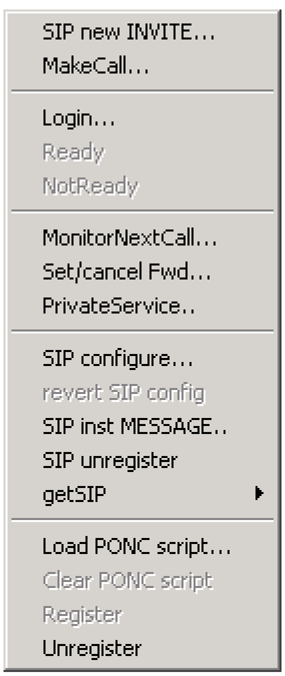
Two sets of operations are defined for the site element (only one of them is usually active at any time), with menu items worded differently depending on the presence of T-Library connection (the **server** configuration option) and SIP registration settings (the **sip-register** configuration option):

- **Connect** (T-Library present)—Opens the T-Library connection to the specified T-Server/SIP Server, registers all SIP-enabled DNs with SIP (sending REGISTER requests to a primary and backup SIP proxy) and T-Library (sending TRegisterAddress to T-Server).
- **Disconnect** (T-Library present)—Closes the T-Library connection (which automatically cancels all T-Library registrations) and unregisters all SIP-enabled DNs (sending REGISTER with zero expiration time).
- **(re-)register SIP** (no T-Library, **sip-register=true**)—Registers all SIP-enabled DNs with SIP.
- **SIP unregister** (no T-Library, **sip-register=false**)—Unregisters all SIP-enabled DNs.

No menu is shown for sites without the T-Library connection and when **sip-register=false**, since no actions can be logically performed for them.

DN-level commands for telephony and agent operations

A set of operations included in the DN-level menu depends on the T-Library registration status and the DN type (as reported by T-Server), and presence of the SIP configuration, as follows:



- **SIP new INVITE...**(regular DN, SIP)—Sends a new INVITE request from the DN (that is, makes a 1pcc call). Use the following parameters:
  destination-DN[@*SIP-proxy-host:port*][,sip=(*on-invite-script*)]

where:

- *destination-DN*—The destination of the call (user part of the request-URI in the INVITE).
- *SIP-proxy-host:port*—(Optional) The location of the SIP proxy/gateway (usually it will be SIP Server). If it is not specified, the value of the **sip-proxy** option is used (which is the most common case).
- *on-invite-script*—The custom SIP script for this particular call, in the same format as the **on-mkcall** configuration option, or the parameters of INVITE message.

  For example:

  `3002` - Sends an INVITE to 3002 through the configured sip-proxy option (SIP Server)

  `3605@pixie:5055` - Sends an INVITE directly to the specified host/port (without SIP Server)

  `3002,sip=(INVITE=(SDP=no))` - Uses the custom SIP command (do not include SDP into INVITE)

  `3002,sip=(SDP=no)` - Uses the custom SIP command, short format

- **MakeCall...** (regular DN, T-Library)—Makes a new 3pcc call (T-Library function TMakeCall). Use the following parameters:
  destination-DN[@*location*][,ty=*type*][,uu=(*user-data*)][,ext=(*extensions*)]

  where:

  - *destination-DN*—The destination of the call (AttributeOtherDN in TMakeCall).
  - *location*—(Optional) The destination location (AttributeLocation).
  - *ty*—(Optional) The MakeCallType (never used with SIP Server).
  - *uu* and *ext*—(Optional) UserData and AttributeExtensions in the request, using standard text representation of TKVList (comma-separated list of "key"="value").

    For example:

    `3002` - Makes a 3pcc call to 3002

    `3301@epi-SIP-sw2` - Makes a call to 3301 at location epi-SIP-sw2 (see the multi.conf sample)

    `4401,uu=(GSW_CALL_TYPE='ENGAGING',GSW_SESSION_DBID=101)` - Makes a call to 4401 and attaches specified user data to the call

- **MakePredictiveCall...** (Routing Point/queue, T-Library)—Makes a 3pcc predictive call (TMakePredictiveCall). Use the following parameters:
  destination-DN[@*location*][,tout=*timeout*][,uu=(*user-data*)][,ext=(*extensions*)]

  where:

  - *destination-DN*—The destination of the call (AttributeOtherDN in TMakeCall).
  - *location*—(Optional) The destination location (AttributeLocation).
  - *timeout*—(Optional) The ring_timeout value (default is 0, which tells T-Server to use the default/configured value).
  - *uu* and *ext*—(Optional) UserData and AttributeExtensions in the request.

- **Login...** (regular DN, T-Library)—Logs an agent in (TAgentLogin). Use the following parameters:
  agent-ID[/*queue*]

where:

- *agent-ID*—The agent ID. The parameter string may include $DN that will be replaced with DN digits.
- *queue*—The ACD queue (optional for SIP Server).

- **Logout** (regular DN, T-Library)—Logs an agent out (TAgentLogout).
- **Ready** (regular DN, T-Library)—Sets an agent to Ready state (TAgentReady).
- **NotReady** (regular DN, T-Library)—Sets an agent to NotReady state (TAgentNotReady).
- **MonitorNextCall...** (regular DN, T-Library)—Enables call monitoring (TMonitorNextCall). Use the following parameters:
  agent-DN[/coach|talk|mute][/agent|call][/all] or *agent-DN*/cancel

  where:

  - *agent-DN*—The DN to be monitored.
  - if **/coach** is specified, sets the monitor mode to **coach**
  - if **/talk** is specified, sets the monitor mode to **connect**
  - if **/mute** is specified, sets the monitor mode to **mute**
    When the monitor mode is not specified, Simulator does not add the MonitorMode key to AttributeExtensions, thus using the T-Server default (which is usually silent monitoring, but may be changed by TServer options)
  - if **/agent** is specified, set monitor scope to **agent**
  - if **/call** is specified, set monitor scope to **call**
  - if **/all** is specified, set monitor type to MonitorAllCalls (default is MonitorOneCall)
  - using the /cancel option in the parameter dialog (for example, "3002/cancel") cancels the call monitoring with the TCancelMonitoring function

- **Set/cancelFwd...** (regular DN, T-Library)—Sets or cancels T-Library call forwarding (either TCallSetForward or TCallCancelForward, depending on the DN presence in parameters). Use the following parameters:
  [*DN-to-forward*] [/*mode*]

  where:

  - *DN-to-forward*—The forwarding destination for TCallSetForward, or empty for TCallCancelForward.
  - */mode*—The forwarding mode, optional for Set (default is /1), mandatory for Cancel, as follows:
    - /**1** = ForwardModeUnconditional
    - /**2** = ForwardModeOnBusy
    - /**3** = ForwardModeOnNoAnswer
    - /**4** = ForwardModeOnBusyAndNoAnswer
    - /**5** = ForwardModeSendAllCalls

- **PrivateSvc** (all DNs, T-Library)—Passes information and request services for this DN (TPrivateService). Use the following parameters:

id [/*extensions*]

Refer to the Framework 8.1 SIP Server Deployment Guide for a list of service IDs and their parameters.

DN-level SIP commands

- **SIP configure...** (SIP-enabled DN)—Modifies the SIP configuration of the particular DN. Parameters must be specified in the same format as in the configuration file, for example:
  on-invite=(ring=(SDP)),play=ExtDN

  Notes:

  - New parameters are added to the existing configuration, replacing options with the same name, but not changing other options. If the intention is to discard old values and apply only the new configuration, start the list with replace keyword, for example: replace,sip=st,play=ExtDN.
  - Most options takes effect for the next call. As an exception (and special feature), the **play** option is applied to all currently active SIP dialogs as well (so the new file/tone starts playing immediately).
  - SIP stack parameters cannot be changed with this command
  - In addition to regular options described, two special options might be used in this command—on-invite-once and on-mkcall-once—have the same effect as their regular counterparts, but are applied to the next call only (and automatically reset after the first use).
- **revert SIP config** (SIP-enabled DN)—Reverts the SIP configuration to the value taken from the configuration file.
- **SIP inst MESSAGE...** (SIP-enabled DN)—Sends the SIP instant MESSAGE (page mode only, RFC 3428). Use the following parameters:
  DN, message-text or *DN,content-type* ="*message-text*"[,x=(*extra-headers*)]

  where:

  - *DN*—The destination of the instant message (with the optional SIP proxy, as in SIP new INVITE).
  - *content-type*—Changes the Content-Type in the MESSAGE request (in the short form text/plain is used).
  - *message-text*—The new lines encoded as '|' , for example: "Hello|just testing|".
  - *extra-headers*—(Optional) A list of additional SIP headers, in the "header=value" format.
- **SIP register** (SIP-enabled DN)—Sends the REGISTER request for the DN, which can be either a new SIP registration or renewal of the existing one.
- **SIP unregister** (SIP-enabled DN)—Sends the REGISTER request with zero expiration.
- **getSIP » device** (SIP-enabled DN)—Opens a window with SIP history of all device-level messages (REGISTER, SUBSCRIBE, and so on). It is also activated by double-clicking the DN element.
- **getSIP » id (last inc)**—Opens SIP history of the last incoming SIP dialog.
- **getSIP » id (last out)**—Opens SIP history of the last outgoing SIP dialog.

Note: By design, the content of the SIP history window is not updated in real time. To get an update, close the window and input the corresponding command again.

Other DN level commands

- **Load PONC script...**(all DNs)—Loads the play-on-new-call (PONC) script to the DN. For Routing Points, the parameter has the same format as it does in the Treatment command described below.
  For example:

  `play=music/ring_back` - Applies the ring-back treatment to a new call on this Route DN

- **Clear PONC script** (all DNs)—Clears the PONC (or Routing) script, stopping any further automatic action for new calls arriving to this device.

- **Register** (all DNs, T-Library)—Registers the DN (TRegisterAddress).

- **Unregister** (all DNs, T-Library)—Unregisters the DN (TUnregisterAddress).

T-Library party commands

A set of operations for T-Library call parties depends on the party state (as reported by T-Server in events). All operations listed in this section have two implicit arguments: ThisDN and ConnectionID.

- **party** » **Answer** (regular DN, Alerting state)= TAnswerCall
- **party** » **Hold** (regular DN, Connected state)= THoldCall
- **party** » **Retrieve** (regular DN, Held state)= TRetrieveCall
- **party** » **Release** (regular DN, any state)= TReleaseCall
- **party** » **1stepTransfer...** (regular DN, Connected state)= TSingleStepTransfer
- **party** » **1stepConference...** (regular DN, Connected state)= TSingleStepConference

For the above party commands, use the following parameters:
destination-DN[@*location*][,uu=(*user-data*)][,ext=(*extensions*)]
where:
destination-DN—The destination of the call (AttributeOtherDN in TMakeCall).
location—(Optional) The destination location (AttributeLocation).
user-data and *extensions*—(Optional) UserData and AttributeExtensions in the request, using standard text representation of TKVList (comma-separated list of "key"="value").

- **party** » **InitTransfer...** (regular DN, Connected state)= TInitiateTransfer

- **party** » **InitConference...** (regular DN, Connected state)= TinitiateConference
  For SIP Server (and many other T-Servers) these InitTransfer and InitConference commands do the same thing: the currently active call is placed on hold and a new consultation call is initiated (some T-Servers require Initiate/Complete operations to match each other). Parameters for both are in the same format as for the **1stepTransfer/Conference...** above.

- **party** » **Alternate** (regular DN, Connected/Held states)= TAlternateCall

- **party** » **Reconnect** (regular DN, Connected/Held states)= TReconnectCall

- **party** » **TransferComplete**(regular DN, Connected/Held states)= TCompleteTransfer
- **party** » **ConferenceComplete** (regular DN, Connected/Held states)= TCompleteConference. All four operations—Alternate, Reconnect, TransferComplete, and ConferenceComplete—are only available when two parties are present on the DN, one in Connected and the other in Held state. They might be performed on either party with exactly the same result (SIP Endpoint Simulaator reorders parties in correct order when generating corresponding T-Library request).
- **party** » **Route...** (Routing Point)—Routes the call to the specified destination. Use the following parameters:
  destination-DN[@*location*][,ty=*type*][,ext=(*extensions*)]

  where:
  - *destination-DN*—The destination of the call (AttributeOtherDN in RequestRouteCall).
  - *location*—(Optional) The destination location (AttributeLocation).
  - *type*—(Optional) The RouteType (default is 0=RouteTypeUnknown).
  - *extensions*—(Optional) AttributeExtensions in the request, using standard text representation of TKVList (comma-separated list of "key"="value").
- **party** » **Treatment...** (Routing Point)—Applies the treatment to the call on a Routing Point. Use one of the following parameters:
  - play=*filename* [,t=*duration*] - TreatmentMusic (with optional DURATION parameter)
  - annc=(*announcement-prameters*) - TreatmentPlayAnnouncement
  - app=(*play-app-prameters*) - TreatmentPlayApplication
  - collect=(*collect-prameters*) - TreatmentCollectDigits
  - record=*filename* - TreatmentRecordUserAnnouncement
- **party** » **SendDTMF...** (regular DN, Connected state) = TSendDTMF (parameter = digits to send)
- **party** » **UserData...** (regular DN, any state) = TUpdateUserData (parameter = TKVList)

TKVList parameter must be specified in its standard text format as a comma-separated list of key-value pairs (the string value must be enclosed in single or double quotes when it starts with digits or includes a comma or parentheses).

- **party** » **PrivateSvc** (all DNs, any state) = TPrivateService for this party, with parameters: *id* [/*extensions*]

Refer to the Framework 8.1 SIP Server Deployment Guide for a list of service IDs and their parameters.

SIP dialog commands

A set of operations for SIP dialogs depends on the dialog state (according to SIP messages sent or received), as follows:

- **dialog** » **talk(answer)** (Alerting state)—Answers the call, sending a 200 OK SIP message.

- **dialog** » **603 Decline** (Alerting state)—Rejects the call, sending a 603 Decline SIP message.
- **dialog** » **SIP cancel** (Initiated state)—Abandons the calling attempt, sending a CANCEL message.
- **dialog** » **SIP hold** (Connected state)—Places the call on hold, sending re-INVITE with the hold SDP (using either RFC 2543 or RFC 3264, depending on the value of **sip-hold-rfc3264**).
- **dialog** » **SIP retrieve** (Held state)—Retrieves the call from hold, sending re-INVITE with the "talk" SDP.
- **dialog** » **BYE** (release) (any state)—Releases the call, sending a BYE message.
- **dialog** » **blindXfer...** (Connected or Held state)—Performs a 1pcc blind transfer to a given destination.
- **dialog** » **startXfer...** (Connected or Held state)—Initiates a two-step transfer to a given destination. Use the following parameters:

  destination-DN[@*SIP-proxy-host:port*]

  where:

  - *destination-DN*—The destination of the call (user part of the request-URI in the INVITE).
  - *SIP-proxy-host:port*—(Optional) The location of the SIP proxy/gateway (usually it will be SIP Server), if it is not specified, the **sip-proxy** option value is used.
- **dialog** » **completeXfer** (Connected/Held state)—Completes the two-step transfer. The operation is only available when two SIP dialogs are present on the DN, one in the Connected and the other in the Held state. It can be performed on either SIP dialog with the same result.
- **dialog** » **customSIP...** (any state)—Executes the Custom SIP script (CSS).
- **dialog** » **getSIP(call)** (any state), also activated by double-clicking a SIP dialog element—Opens a window with full SIP history of all call-related messages for a particular dialog collected so far. As with device-level SIP history, the content is not updated in real time. To get an update, close the window and invoke the command again.
- **dialog** » **SendDTMF** » **NTE/rfc2833...** (SIP, Connected)—Sends DTMF using RFC 2833 telephony events.
- **dialog** » **SendDTMF** » **SIP msg INFO...** (SIP, Connected)—Sends DTMF using the INFO message.
- **dialog** » **SendDTMF** » **audio tone...** (SIP, Connected)—Sends DTMF as in-band audio tones.

Configuration File and Options

The configuration file includes the following sections located in the file in arbitrary order:

- **Main** section called **[[[SimConfOptions#TcCM|TcCM]]]**—The mandatory section that contains global (application level) options.
- **Site** section(s) with the variable name **[[[SimConfOptions#SITESIPS|site:*anyname*]]]**—The mandatory section describing the "site"—a set of simulated DNs/devices that are served by the same SIP Server, or located behind the same trunk, or logically grouped together by some other trait. The section name contains the fixed prefix **[site:]**. There can be multiple sections in the configuration file. Site sections contain a number of site-level options and a number of DN definitions, at least one DN must be defined.
- **SIP configuration** section(s) with the variable name **[[[SimConfOptions#EXTSIP|*anyname*]]]**—The mandatory section for simulated SIP endpoints, otherwise is optional. It contains DN-level SIP options. There can be multiple sections. In most cases, multiple simulated emulated DNs share the same settings, thus these sections provide a mechanism to list all common settings in one place.
- **Configuration variables** section called **[[[SimConfOptions#VARS|[opt:vars]]]]**—(Optional) If present, this section includes user-defined variables for the entire configuration. Unlike other sections, everything defined in this section is processed at the moment the configuration file is read by using a text substitution. It is not kept afterwards. For example, if that section includes line
  ```
  sipserver-host = sipserver.mydomain.com
  ```
  then every occurrence of `${sipserver-host}` in the entire configuration file is textually replaced with `sipserver.mydomain.com` before further processing.
- **[[[SimConfOptions#Log|Log]]]** section—(Optional) If present, this section includes logging parameters.

The configuration file uses the standard syntax:

- All options belong to a section, starting with section name in square brackets.
- Option names with their values are separated by the equal sign, each listed on its own line.
- Comments starts with the # (pound) sign.

In the installation directory, you can find the sample configuration file **esttt.conf**.

Section [TcCM]

The main **[TcCM]** section contains the configuration options that are used to define application-wide settings.

connect-on-startup

Section: **[TcCM]**
Default Value: `true`
Valid Values: `true, false`

Specifies whether SIP Endpoint Simulator connects to all configured SIP Servers/T-Servers at startup. Otherwise, it would connect later, by double-clicking on the site in the GUI, using the context menu command, or control command from a QAART script.

open-log-on-startup

Section: **[TcCM]**
Default Value: `false`
Valid Values: `true, false`

Specifies whether the Log window opens when Simulator starts.

log-to-file

Section: **[TcCM]**
Default Value: No default value
Valid Values: Any valid file name

DEPRECATED. Use the **[[SimConfOptions#Log|log]]** section instead.

sip-port

Section: **[TcCM]**
Default Value: No default value

Specifies the port for incoming SIP messages. A single SIP port can be used for all DNs, or a specific port can be specified for each site or a particular DN (see the **sip-port** in the **site:*extSIP*** section below).

The SIP port specified on a DN overrides the port specified on the "site" level, which is in turn takes precedence over the global SIP port in the **[TcCM]** section.

SIP Endpoint Simulator supports multiple SIP stacks with a separate SIP port for each "site", or even for each DN. The same **sip-port** option is used for port specification; depending on where it is located, this option creates additional SIP stacks, as follows:

- when **sip-port** is specified in a DN parameter list, the DN-specific SIP stack is created serving this DN only
- when **sip-port** is specified in the SIP Server or external SIP section, the site-specific SIP stack is created serving all DNs configured on this site (except for those having their own stack)
- **sip-port** still can be specified in the main **[TcCM]** section with the main SIP stack serving all other DNs.

sip-addr

Section: **[TcCM]**
Default Value: `$auto`
Valid Values: `$auto, $ipv4, $ipv6,` FQDN, or any valid IP address

Specifies the IP address for a SIP contact header.

rtp-addr

Section: **[TcCM]**
Default Value: `$auto`
Valid Values: `$auto, $ipv4, $ipv6`, FQDN, or any valid IP address

Specifies the IP address for SDP and binding RTP ports.

rtp-ports

Section: **[TcCM]**
Default Value: `8000..8599`

Specifies the port range for RTP.

telnet-ctrl-port

Section: **[TcCM]**
Default Value: No default value
Valid Values: Any valid port number

Specifies the port for the telnet-based control connection. If not specified, no port is opened, so the QAART integration will not work. Only manual operations will work.

tl:n

Section: **[TcCM]**
Default Value: None
Valid Values: A comma-separated list of key-value pairs with the key being an upper-case letter from A to Z and its value being any string.

Specifies the aliases for the tones (tone label), where *n* is any digit from 0 to 9. You can define up to 10 **tl:*n*** options.

Example: `tl:0 = (A=Alice,B=Bob,C=Carol,D=Dave,E=ExtDN,F=ExtTrunk,G=Gcti,H=MoH)`

Section [site:*SIPserver*]

This section defines simulated SIP endpoints and T-Library-controlled DNs, each might have a single reference to SIP Server/T-Server and/or the location of a SIP proxy. The section name starts with the **site:** prefix, you can replace variable *SIPserver* according to your needs.

Different format of the host and port specification for T-Library and SIP connections result from different methods of opening these connections:

- For a T-Library connection (the **server** option), the value is formatted as an XKVList suited for the TOpenServerX function, and might include additional parameters understood by Genesys Common Library. As for all XKVLists used in the configuration, if the text value starts with a digit (as in case of an explicit IP address instead of the host name), it must be enclosed in quotes.
- For a SIP connection, the SIP location (the **sip-proxy** option) is formatted per standard URL rules (the **sip:** prefix is not required, but accepted if specified for clarity) and might include additional SIP

parameters. Only one such parameter currently supported: transport may be specified as udp (default), tcp, or tls.

server

Section: **[site:*SIPserver*]**
Value Format: `(host=${sipserver-host},port=${sipserver-tlib-port})`
Example: `server = (host="192.168.3.241", port=28001)`

Specifies the SIP Server host and T-Library port for this site.

sip-proxy

Section: **[site:*SIPserver*]**
Value Format: `${sipserver-host}:${sipserver-sip-port}`
Example: `sip-proxy = 192.168.3.241:5060;transport=udp`

Specifies the SIP proxy host and port for this site. Usually, it would be the SIP Server host and SIP port.

sip-register

Section: **[site:*SIPserver*]**
Default Value: `true`
Valid Values: `true, false`

Specifies whether a DN is registered (SIP registration) with SIP Server.

dn

Section: **[site:*SIPserver*]**
Default Value: No default value
Valid Values: A comma-separated list of parameters

Specifies the DN configured in the Configuration Layer with the following parameters:

- For SIP-enabled DNs (supports SIP and RTP) (with 'contact' only required when **sip-register** is set to `false`, so DNs are not registered on SIP Server), specify the DN name/number and include a reference to a SIP configuration section (for example, **st** for standard). It can also list the configuration options in-line.
  Example:

  ```
  dn = 3001,sip=st,play=Alice
  ```

- For Routing Points and internal DNs that are only monitored/controlled via T-Library, specify the DN name/number. Routing Points might have a treatment configured, which will be applied automatically to every call arriving at that Routing Point.
  Example:

  ```
  dn = 1020,script="play=music/tl:G"
  ```

> **Important**
> There can be multiple **dn** options in the configuration file—each one adds a single DN to the configuration.

Section [site:*extSIP*]

This section specifies simulated "external" SIP endpoints. The section name starts with the `site:` prefix, you can replace variable *extSIP* according to your needs.

sip-proxy

Section: **[site:*extSIP*]**
Valid Format: `${sipserver-host}:${sipserver-sip-port}`
Example: `sip-proxy = 192.168.3.241:5060;transport=udp`

Specifies the SIP proxy host and port for this site. Usually, it would be the SIP Server host and SIP port.

sip-port

Section: **[site:*extSIP*]**
Default Value: No default value

Specifies the SIP port. This DN setting overrides the port specified on the "site" level and takes precedence over the global SIP port in the **[TcCM]** section.

SIP Endpoint Simulator supports multiple SIP stacks with a separate SIP port for each "site", or even for each DN. The same **sip-port** option is used for the port specification; depending on where it is located, this option creates additional SIP stacks, as follows:

- When **sip-port** is specified in the DN parameter list, a DN-specific SIP stack is created, serving this DN only.
- When **sip-port** is specified in the SIP Server or external SIP section, the site-specific SIP stack created, serving all DNs configured on this site (except for those having their own stack).
- **sip-port** can be specified in the main **[TcCM]** section with the main SIP stack serving all other DNs.

sip-register

Section: **[site:*extSIP*]**
Default Value: `false`

Specifies whether a DN is registered with a SIP proxy.

dn

Section: **[site:*extSIP*]**

This section contains DNs that can simulate non-CTI–enabled endpoints (configured in the Configuration Layer) or external devices behind a gateway (not configured). In the latter case, a trunk DN is required for making calls to these DNs, with the **prefix** matching the digits and **contact** option pointing to the host and sip-port of the SIP Endpoint Simulator.

```
dn = 3600,sip=st,play=Dave
dn = 3607,sip=st,play=ExtDN,sip-port=5037
```

SIP configuration section

For each **sip=*name*** option used in any DN definition, the configuration file must include a section **[*name*]** that defines a set of SIP parameters. All these parameters can also be specified in-line in the DN definition.

codecs

Section: **[*name*]**
Default Value: No default value
Value format: `(pcmu,pcma)`

Specifies the list of codecs to be used by an endpoint.

play

Section: **[*name*]**
Default Value: No default value
Value can contain one of the following:

- tone name alias (as defined by the **tl:n** option)
- reference to a telephony tone defined in the music/QTMF file (for example, music/ring_back)
- any valid file name (with either relative or absolute path) to play pre-recorded media.

Specifies media to be played when a connection is established. If the option is not specified, no media is played. You can specify the **play** option in the SIP section, in which case media is played for all DNs (unless a codec-related scenario is tested), or you can specify the **play** option for a specific DN.

on-mkcall

Section: **[*name*]**
Default Value: `(INVITE=(SDP=talk))`

Specifies the custom sequence for 1pcc MakeCall (a.k.a. **SIP new INVITE**).

Custom SIP scripts (CSS) describe the sequence of SIP messages to be sent in a particular case (for example, when performing 1pcc MakeCall, on a receiving INVITE message, or by an explicit command from a QAART script). As soon as that sequence is completed, the Simulator returns to its regular semi-automatic operation. Particularly, it automatically sends an ACK on a receiving final response to an outgoing INVITE (but does not auto-answer the incoming call without the instruction to do so).

CSS for MakeCall (option **on-mkcall**) must start with an INVITE message, and usually this is the only message specified. Parameters for this message include specification of SDP content. By default, a regular "talk" SDP is included, so all these lines are equivalent:

```
on-mkcall = (INVITE=(SDP=talk))
on-mkcall = (INVITE)
on-mkcall = (INVITE=(SDP))
```

To send an initial INVITE without the SDP, use the following setting:

```
on-mkcall = (INVITE=(SDP=no))
```

on-invite

Section: **[*name*]**
Default Value: `(ring)`

Specifies the custom response to incoming INVITE requests.

Custom SIP scripts (CSS) describe the sequence of SIP messages to be sent in a particular case (for example, when performing 1pcc MakeCall, on a receiving INVITE message, or by an explicit command from a QAART script). As soon as that sequence is completed, the Simulator returns to its regular semi-automatic operation. Particularly, it automatically sends an ACK on a receiving final response to an outgoing INVITE (but does not auto-answer the incoming call without the instruction to do so).

CSS for incoming call (option on-invite) is a comma-separated list of message names listed below, along with the pause specification. Each message can have a SDP content parameter added, the same way as for the INVITE message described above.

Notation	Sends this SIP message
100	100 Trying (any other numeric SIP response code can be used as well)
180 or ring	180 Ringing (By default, the regular "talk" SDP is added to 200 and reliable 1xx responses only, use `=(SDP)` to add the SDP to non-reliable response)
183 or early	183 Session Progress (By default, the regular "talk" SDP is added to 200 and reliable 1xx responses only, use `=(SDP)` to add the SDP to non-reliable response)
200 or talk	200 OK (By default, the regular "talk" SDP is added to 200 and reliable 1xx responses only, use `=(SDP)` to add the SDP to non-reliable response)
486 or busy	486 Busy Here
p=*number*	the pause for a given number of seconds (fractional values are accepted, so to pause for half a second use "p=0.5")

Examples of auto-answer and early media specification:

```
[AA] on-invite = (100,ring,p=0.5,talk)
[EM] on-invite = (100,early=(SDP))
```

> **Important**
>
> While a custom SIP script is active, it must account for all sent and received SIP messages. In particular, the auto-answer example does not work with reliable provisional responses enabled (because of a PRACK message not being processed in the CSS correctly).

auto-hold-on-talk

Section: ***[name]***
Default Value: `true`

If set to `true`, automatically places the active SIP dialog on hold when answering/retrieving another dialog (like real phones do) (1pcc Hold operation).

sip-hold-rfc3264

Section: ***[name]***
Default Value: `false`

If set to `true`, 1pcc Hold uses RFC 3264 (`sendonly` in SDP) otherwise RFC 2543 used (Hold SDP with 0.0.0.0 connection address).

support-100rel

Section: ***[name]***
Default Value: `true`

Enables support of reliable provisional responses (100rel) by adding the `Supported: 100rel` header to INVITE and generate PRACK messages.

require-100rel

Section: ***[name]***
Default Value: `false`

When set to `true` and 100rel is advertised by the remote end, SIP Endpoint Simulator requests the PRACK in a SIP provisional response; that is, adds a `Require: 100rel` header to the response.

Note: Per RFC 3262, the 100rel feature applies to all 1xx responses except for 100.

Section [opt:vars]

The **[opt:vars]** section contains the user-defined variables that apply to the entire configuration.

For example:

```
[opt:vars]
app_sip_port = 13242
app_telnet_port = 13243
app_host_ip = CPRS_FWK
app_log_name = ./logs/EpiPhone.log
```

Section [log]

SIP Endpoint Simulator provides two types of logging, which can work in parallel or independently (displaying the same content):

- A short-term log is directed to a special log window opened from the menu (or automatically at startup when configured). The output going to the log window can be "paused" by pressing the corresponding button, which affects only the output into that log. SIP Endpoint Simulator continues to operate as usual, writing the file log (if configured).

- A long-term log is written to the specified file, via either a built-in logger (deprecated) or Genesys Log Library. In the former case, use of the file name results in that file being overwritten on each run.

For using Log Library, the **log-to-file** option must not be specified. Instead, the **[log]** section must be added to the configuration file with the values as described below.

all

Section: **[log]**
Default Value: No default value
Valid Values (log output types):

- `stdout`: Log events are sent to the Standard output (`stdout`).
- `stderr`: Log events are sent to the Standard error output (`stderr`).
- `[filename]`: Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Specifies the outputs to which an application sends all log events. The log output types must be separated by a comma when more than one output is configured. For example: **all** = `stdout, logfile`

segment

Section: **[log]**
Default Value: `10 MB`
Valid Values:

- `false`: No segmentation is allowed.
- `<number> KB` or `<number>`: Sets the maximum segment size, in kilobytes. The minimum segment size is `100 KB`.
- `<number> MB`: Sets the maximum segment size, in megabytes.
- `<number> hr`: Sets the number of hours for the segment to stay open. The minimum number is 1 hour.

Specifies whether there is a segmentation limit for a log file. If there is, sets the mode of measurement, along with the maximum size. If the current log segment exceeds the size set by this option, the file is closed and a new one is created. This option is ignored if log output is not configured to be sent to a log file.

expire

Section: **[log]**
Default Value: `10`
Valid Values:

- `false`: No expiration; all generated segments are stored.
- `<number> file` or `<number>`: Sets the maximum number of log files to store. Specify a number from `1–1000`.
- `<number> day`: Sets the maximum number of days before log files are deleted. Specify a number from `1–100`.

Determines whether log files expire. If they do, sets the measurement for determining when they expire, along with the maximum number of files (segments) or days before the files are removed. This option is ignored if log output is not configured to be sent to a log file.