SIP Endpoint Simulator Deployment Guide

Configuration File and Options

12/24/2025

Contents

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.