This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

SIP Endpoint SDK Developer's Guide

SIP Endpoint SDK 8.5.2OSX

2/11/2022

Table of Contents

Developer's Guide

Important

The following features are not supported:

- Video
- IPv6
- NAT, ICE, TURN, STUN
- SIP Cluster
- SIP Proxy

This Developer's Guide contains information that will help you understand how to:

- Configure SIP Endpoint SDK for OS X
- Run the QuickStart application

Before reading this material you may want to:

- Install the SIP Endpoint SDK for OS X.
- Ensure you have access to the latest version of the API Reference.

Configuring SIP Endpoint SDK for OS X

> **Important**
>
> The following features are not supported:
>
> - Video
> - IPv6
> - NAT, ICE, TURN, STUN
> - SIP Cluster
> - SIP Proxy

Using the Default Configuration File

You can find the default configuration file in the following location:

```
<installation folder>/Bin/SipEndpoint.config
```

This file contains XML configuration details that affect how your SIP Endpoint SDK application behaves. The initial settings are the same as those specified for use with the QuickStart application included with your SIP Endpoint SDK release.

Configuration settings are separated into two containers: the Basic Container holds the connectivity details that are required to connect to your SIP Server, while the Genesys Container holds a variety of configuration settings.

Basic Container

The first Container ("Basic") holds the basic connectivity details that are required to connect to your SIP Server. This container has at least one connection (Connectivity) element with the following attributes:

```
<Connectivity user="DN" server="SERVER:PORT" protocol="TRANSPORT"/>
```

Make the following changes and save the updated configuration file before using the SIP Endpoint SDK:

- **user="DN"** — Supply a valid DN for the user attribute.
- **server="SERVER:PORT"** — Replace SERVER with the host name where your SIP Server is deployed, and PORT with the SIP port of the SIP Server host. (The default SIP port value is 5060.)

- **protocol="TRANSPORT"** — Set the protocol attribute to reflect the protocol being used to communicate with SIP Server. Possible values are UDP, TCP, or TLS.

Genesys Container

The second Container ("Genesys") holds a number of configurable settings that are organized into domains and sections. These settings do not have to be changed, but can be customized to take full control over your SIP Endpoint SDK applications.

An overview of the settings in this container and the valid values for these settings is provided here:

Domain	Section	Setting
policy		
	endpoint	
		audio_qos
		include_os_version_in_user_agent_header
		include_sdk_version_in_user_agent_header
		ip_versions
		public_address
		rtp_inactivity_timeout
		rtp_port_min
		rtp_port_max
		signaling_qos
		sip_port_min
		sip_port_max
		sip_transaction_timeout
		vq_report_collector
		vq_report_publish
		answer_sdp_priority
		sip_port_binding
	session	
		agc_mode
		auto_answer
		auto_answer_delay
		dtmf_method
		echo_control
		noise_suppression
		dtx_mode
		reject_session_when_headset_na
		sip_code_when_headset_na
		vad_level

Domain	Section	Setting
		ringing_enabled
		ringing_timeout
		ringing_file
	device	
		audio_in_device For more information, see Audio Device Settings
		audio_out_device
		capture_device
		ringer_device
		headset_name
		use_headset
codecs—See Working with Codec Priorities		
proxies		
	proxy<*n*>	
		display_name
		domain
		password
		reg_interval
		reg_match_received_rport
		reg_timeout
		mailbox
		password
		server
		timeout
		transport
		user
system		
	diagnostics	
		enable_logging
		log_file
		log_level
		log_options_provider
		logger_type
		security
		cert_file
		tls_enabled

Domain	Section	Setting
		use_srtp
		priv_key_file
		ca_list_file

policy Domain

endpoint Section

audio_qos

Valid Values: Number

The integer value representing the DSCP bits to set for RTP audio packets. The value should be 4 * Preferred DSCP value.

include_os_version_in_user_agent_header

Valid Values: 0 or 1
Default Value: 1

If set to 1, the user agent field includes the OS version the client is currently running on.

include_sdk_version_in_user_agent_header

Valid Values: 0 or 1
Default Value: 1

If set to 1, the user agent field includes the SDK version the client is currently running on.

ip_versions

Valid Values: IPv4, empty Default Value: IPv4.

A value of IPv4 or an empty value means that the application selects an available local IPv4 address.

NOTE: This parameter has no effect if the public_address option specifies an explicit IP address.

public_address

Valid Values: See description below.

Local IP address of the machine. This setting can be an explicit setting or a special value that the SDK uses to automatically obtain the public address.

Valid Values:
This setting may have one of the following explicit values:

- An IP address. For example, `192.168.16.123` for IPv4.
- A bare host name, for example, `epsipmac2`.

This setting may have one of the following special values:

- `$auto`—The SDK selects the first valid IP address on the first network adapter that is active (status=up) and has the default gateway configured. IP family preference is specified by the **policy.endpoint.ip_versions** setting.
- `$ipv4`—Same behavior as the `$auto` setting but the SDK restricts the address to a particular IP family.
- `$host`—The SDK retrieves the standard host name for the local computer using the `gethostname` system function.
- An adapter name or part of an adapter name prefixed with `$`. For example, `en0` or `en1`. The specified name must be different from the special values `$auto`, `$ipv4`, and `$host`.

Default Value: Empty string which is fully equivalent to the `$auto` value.

If the value is specified as an explicit host name, the `Contact` header includes the host name for the recipient of SIP messages (SIP Server or SIP proxy) to resolve on their own. For all other cases, including `$host`, the resolved IP address is used for `Contact`. The value in SDP is always the IP address.

rtp_inactivity_timeout

Valid Values: Integer between `5` and `150`
Default Value: `150`
Suggested Value: `30`

Timeout interval in seconds for RTP inactivity.

rtp_port_min

Valid Values: Number
Default Value: `8000`

The integer value representing the minimum value for an RTP port range. Must be within the valid port range of 8000 to 65535. If the minimum value is not specified or set to an invalid value, the default value of 8000 is used for `rtp_port_min`. Setting the minimum to a value that is larger than the maximum is considered an error and will result in a failure to initialize the endpoint.

rtp_port_max

Valid Values: Number
Default Value: 9000

The integer value representing the maximum value for an RTP port range. Must be within the valid port range of 8000 to 65535. If the maximum value is not specified or set to an invalid value, the default maximum 9000 is used for `rtp_port_max`. Setting the maximum to a value that is less than the minimum is considered an error and results in a failure to initialize the endpoint.

signaling_qos

Valid Values: Number

The integer value representing the DSCP bits to set for SIP packets. Integer value should be configured to `4 * Preferred DSCP value`.

sip_port_min

Valid Values: Number between 1 and 65535
Default Value: 5060

The integer value representing the minimum value for a SIP port range. Must be within the valid port range of 1 to 65535. If the minimum value is not specified or set to an invalid value, the default value of 5060 is used for `sip_port_min`. Setting the minimum to a value that is larger than the maximum is considered an error and will result in a failure to initialize the endpoint.

sip_port_max

Valid Values: Number between 1 and 65535
Default Value: 5080

The integer value representing the maximum value for a SIP port range. Must be within the valid port range of 1 to 65535. If the maximum value is not specified or set to an invalid value, the default value of 5080 is used for `sip_port_max`. Setting the maximum to a value that is less than the minimum is considered an error and will result in a failure to initialize the endpoint.

sip_transaction_timeout

Valid Values: Number between 1 and `32000`
Default Value: `4000`

SIP transaction timeout value in milliseconds. Valid values are 1 through `32000`, with a default value of `4000`. The recommended value is `4000`.

vq_report_collector

See Producing RTCP Extended Reports

vq_report_publish

See Producing RTCP Extended Reports

answer_sdp_priority

Valid Values: `config` or `offer`
Default value: `config`

- `config`—The endpoint selects the first codec from the codec configuration listed in both the codec configuration and the SDP offer.
- `offer`—the endpoint selects the first codec in the SDP offer listed in both the codec configuration and the SDP offer.

sip_port_binding

Valid Values: `0` or 1
Default Value: `0`

- `0`—opens the SIP port to listen on any interface
- 1—the SIP port binds to the interface specified by the `public_address` setting and listens only on this IP address.

session Section

agc_mode

Valid Values: `0` or `1`

If set to 0, AGC (Automatic Gain Control) is disabled; if set to 1, it is enabled. Default: 1. Other values are reserved for future extensions. This configuration is applied at startup, after which time the `agc_mode` setting can be changed to `1` or `0` from the main sample application. NOTE: It is not possible to apply different AGC settings for different channels in multi-channel scenarios.

auto_answer

Valid Values: `0` or `1`

If set to 1, all incoming calls should be answered automatically.

auto_answer_delay

Valid Values: Number in milliseconds

Default Value: `0`

Time in milliseconds to wait before auto-answering (only applicable when `auto_answer=1` and `auto-answer` is not blocked by missing headset).

dtmf_method

Valid Values: `Rfc2833`, `Info`, or `InbandRtp`

Method to send DTMF

echo_control

Valid Values: `0` or `1`

If set to 1, echo control is enabled.

noise_suppression

Valid Values: `0` or `1`

If set to 1, noise suppresion is enabled.

dtx_mode

Valid Values: `0` or `1`

If set to 1, DTX is activated.

reject_session_when_headset_na

Valid Values:`0` or `1`

If set to 1, the SDK should reject the incoming session if a USB headset is not available.

sip_code_when_headset_na

Valid Values: Number

Defaul Value: `480`
If a valid SIP error code is supplied, the SDK rejects the incoming session with the specified SIP error code if a USB headset is not available.

vad_level

Valid values: 0–3,from 0 (conventional VAD) to 3 (aggressive high)

Sets the degree of bandwidth reduction.

ringing_enabled

Valid values: 0, 1, 2, 3, 4, or 5.
Default Value: 3

- 0—event Ringing disabled
- 1—event Ringing enabled
- 2—play ringtone internally (event Ringing disabled)

- 3—play ringtone internally and enable event Ringing.
- 4—play ringtone through separate "ringer" device
- 5—play ringtone through "ringer" device and enable event Ringing

ringing_timeout

Valid Values: Empty, 0, or a positive number
Default Value: 0

Specifies the duration, in seconds, of the ringing tone. If set to `0` or if the value is empty, the ringing time is unlimited.

ringing_file

Valid Values: Empty string or string to the path to the ringing sound file. The path may be a file name in the current directory or the full path to the sound file.
Default Value: `ringing.wav`

Specifies the audio file that is played when the ringing tone is enabled with the `ringing_enabled` option.
Note that WebRTC does not support MP3 playback. The ringtone file for built-in ringing should be a RIFF (little-endian) WAVE file using one of the following formats:

kWavFormatPcm = 1, PCM, each sample of size bytes_per_sample
kWavFormatALaw = 6, 8-bit ITU-T G.711 A-law
kWavFormatMuLaw = 7, 8-bit ITU-T G.711 mu-law

Uncompressed PCM audio must be 16 bit mono or stereo and have a frequency of 8, 16, or 32 KHZ.

`device` Section

audio_in_device

Valid Values: String

Microphone device name. For more information, see Audio Device Settings

audio_out_device

Valid Values: String

Speaker device name

capture_device

Valid Values: String

Capture device name.

ringer_device

Valid Values: String

Name of the device dedicated to playing ringing tone (used when `ringing_enabled` set to 4 or 5)

headset_name

Valid Values: String

The name of the headset model.

use_headset

Valid values: `0` or `1`

If set to `0`, the audio devices specified in `audio_in_device` and `audio_out_device` are used by the SDK. If set to `1`, the SDK uses a headset as the preferred audio input and output device and the audio devices specified in `audio_in_device` and `audio_out_device` are ignored.

`codecs` Section

See Working with Codec Priorities.

`proxies` Domain

`proxy <n>` Section

display_name

Valid Values: String

Proxy display name

domain

Valid Values: String containing any valid SIP domain
Default Value: Empty string

A SIP domain is an application layer configuration defining the management domain of a SIP proxy. The configured value should include `hostport` and may include `uri-parameters` as defined by RFC 3261. The scheme, `userinfo`, and `transport` URI parameters are included automatically.

If set to an empty string, SIP Endpoint SDK uses the parameters from the `Connectivity` section to construct the SIP domain value as it did in previous versions.

password

Valid Values: String

Proxy password. Password configured for DN object.

reg_interval

Valid Values: Integers greater than or equal to `0`
Default Value: `0`

The period, in seconds, after which the endpoint starts a new registration cycle when a SIP proxy is down. Valid values are integers greater than or equal to `0`. If the setting is empty or negative, the default value is `0`, which means no new registration cycle is allowed. If the setting is greater than `0`, a new registration cycle is allowed and will start after the period specified by `regInterval`.

reg_match_received_rport

Valid Values: `0` or `1`
Default Value: `0`

This setting controls whether or not SIP Endpoint SDK should re-register itself when receiving a mismatched IP address in the `received` parameter of a `REGISTER` response. This helps resolve the case where SIP Endpoint SDK for OS X has multiple network interfaces and obtains the wrong local IP address. A value of 0 (default) disables this feature and a value of 1 enables re-registration.

reg_timeout

Valid Values: Number

The period, in seconds, after which registration should expire. A new `REGISTER` request will be sent before expiration. Valid values are integers greater than or equal to `0`. If the setting is `0`, then registration is disabled, putting the endpoint in standalone mode. If the setting is `empty/null`, the default value of `1800` seconds is used.

`mailbox` Section

> **Important**
> The **mailbox** section is a sub-section of the **proxy<n>** section.

password

Valid Values: String

Mailbox password

server

Valid Values: String

Proxy server address and port for this mailbox

timeout

Valid Values: Any positive integer
Default Value: `1800` seconds (30 minutes)

Subscription expiration timeout in seconds. If the setting is missing or set to `0`, the SDK uses a default timeout of 1800 seconds (30 minutes).

transport

Valid Values: `udp`, `tcp`, or `tls`

Transport protocol to use when communicating with server

user

Valid Values: String

Mailbox number this line subscribes to.

`system` Domain

`diagnostics` Section

enable_logging

Valid Values: `0` or `1`

Valid values of `0` or `1` disable or enable logging.

log_file

Valid values: String containing full path or relative path to the log file.

Log file name, for example, `SipEndpoint.log`

log_level

Valid Values: Number, `0–4`

Valid values correspond to log levels:

- 0 = "Fatal"

- 1 = "Error"
- 2 = "Warning"
- 3 = "Info"
- 4 = "Debug".

log_options_provider

Valid Values: String

Valid values for webrtc are (`warning, state, api, debug, info, error, critical`). For example: `gsip=2, webrtc=(error,critical)`

logger_type

Valid Values: `external`

If set to `external`, an external logger is used.

`security` Section

cert_file

Valid Values: String

Full path pointing to the certificate file used as a client-side certificate for outgoing TLS connections and as a server-side certificate for incoming TLS connections. For example, `/users/Desktop/certificate/hostname_cer.pem`

tls_enabled

Valid Values: Number
Default Value: `0`

If set to 1, connection with TLS transport will be registered.

use_srtp

Valid Values: `optional`, `allowed`, `disabled`, `off`, `elective`, `both`, `enabled`, `force`, `mandatory`

Indicates whether to use SRTP:

- `optional` or `allowed`—do not send secure offers, but accept them
- `disabled` or `off`—do not send secure offers and reject incoming secure offers
- `elective` or `both`—send both secure and non-secure offers and accept either
- `enabled`—send secure offers, accept both secure and non-secure offers
- `force` or `mandatory`—send secure offers, reject incoming non-secure offers

priv_key_file

Valid Values: String

Ful path to the **endpoint private key** file used for client-side authentication for outgoing TLS connections. For example, `/users/Desktop/certificate/hostname_priv_key.pem`

ca_list_file

Valid Values: String

Full path to the **Certificat Authority's list file**. For example, `/users/Desktop/certificate/ca/ca_cert.pem`

Producing RTCP Extended Reports

You can use SIP Endpoint SDK to produce RTCP Extended Reports (RFC 3611) and publish them according to RFC 6035 at the end of each call, using a collector address of your choice.

> **Important**
>
> The publish message is sent to the specified collector address only if the `vq_report_collector` parameter is configured with the `user@server:port;transport=udp` format. For example, `collector@127.0.0.1:5160;transport=udp`.

Settings:

```
<domain name="policy">
<section name="endpoint">
...
<!--
Valid values for Voice Quality (VQ) report publish setting (vq_report_publish):
```

```
0--VQ report is not published
1--VQ report is published to the collector at the end of the call--
  see the vq_report_collector setting information below
-->
<setting name="vq_report_publish" value="0"/>
<!--
Valid values for Voice Quality (VQ) report collector setting (vq_report_collector):
NULL or Empty--The VQ report is published to the proxy described in the
Connectivity section
FQDN or IP address along with port and transport--
 collector@SipServer.genesyslab.com:5060;transport=udp
-->
<setting name="vq_report_collector"
value="collector@SipServer.genesyslab.com:5060;transport=udp"/>
</section>
```

Endpoint:

The `vq_report_publish` and `vq_report_collector` settings can be read from the Endpoint Policy by using the following methods:

```
GetEndpointPolicy(EndpointPolicyQuery.VqReportCollector);
GetEndpointPolicy(EndpointPolicyQuery.VqReportPublish);
```

Working with Codec Priorities

Codecs are listed by name in the `codecs` domain of the configuration file. Codecs are listed by priority with codecs closer to the top of the list having higher priority. To disable a codec, comment out or remove the codec from the configuration file.

The value of setting `payload_type is an integer. For codecs with dynamic payload type such as iSAC, iLBC, OPUS valid values are between 96 and 127.`

Example

```
<domain name="codecs">
        <section name="PCMU/8000">
          <setting name="payload_type" value="0"/>
        </section>
        <section name="PCMA/8000">
          <setting name="payload_type" value="8"/>
        </section>
        <section name="G722/16000">
          <setting name="payload_type" value="9"/>
        </section>
        <section name="iLBC/8000">
          <setting name="payload_type" value="102"/>
        </section>
        <section name="iSAC/32000">
          <setting name="payload_type" value="104"/>
        </section>
        <section name="iSAC/16000">
          <setting name="payload_type" value="103"/>
        </section>
        <section name="opus/48000/2">
          <setting name="payload_type" value="120"/>
        </section>
```

```
</domain>
```

Audio Device Settings

You can configure a headset and other audio input devices using the following parameters in the configuration file:

- `use_headset`
- `headset_name`
- `audio_in_device`
- `audio_out_device`

If the SDK cannot access any of the configured audio devices, the SDK uses the default system audio devices.

If `use_headset` is set to `1`, the SDK searches for a headset with configured `headset_name`. If the SDK does not find the headset, it uses the system defaults.

If `use_headset` is set to `0`, the SDK skips the search for a headset and searches for audio in and audio out devices with names configured by `audio_in_device` and `audio_out_device`. If the devices are not found, the SDK uses the system defaults.

The procedure for audio device selection is applied on startup and every time any changes are made to device presence (such as when a new device is plugged in or an existing device is removed)

Auto-answer and Call Rejection

The auto-answer and call rejection features depend on the `use_headset`, `auto_answer`, and `reject_session_when_headset_na` configuration settings and whether or not audio devices are available. The following tables describe auto-answer and call rejection behavior:

Auto-answer and Call Rejection when use_headset=1

Headset is Available The SDK considers a headset available if the headset is found by name. Outgoing calls can be initiated.	`auto_answer=1`	Incoming calls are answered automatically: • As audio if `auto_accept_video=0` • As audio with video if the call has video and `auto_accept_video=1`
	`auto_answer=0`	Incoming calls are answered

		manually and the user explicitly selects whether or not video streams should be accepted (using the `has_video` parameter supplied in the `gs_session_info` argument)
Headset is Not Available The SDK decides that no headset is available if a headset was not found by name. An audio device is still assigned, if possible (that is, if any supported devices are present in the system), using the first available audio input and output devices or the system defaults.	No auto-answer is possible in this sub-case, so the `auto_answer` setting is not used	`reject_session_when_headset_na=1` • Incoming calls are automatically rejected • Outgoing calls are blocked
		`reject_session_when_headset_na=0` • Incoming calls can be answered manually—it is assumed that the agent will plug the headset in (or use an available non-headset device, if applicable) before answering the call • Outgoing calls can be initiated—it is the agent's responsibility to ensure that the appropriate audio devices are available before the call is answered by the remote side

Auto-answer and Call Rejection when use_headset=0

Audio devices are configured using the `audio_in_device` and `audio_out_device` settings. If no valid `audio_in_device` and `audio_out_device` devices are found, the SDK selects the system defaults. Outgoing calls can be initiated.

Both microphone and speaker are available	`auto_answer=1`	Incoming calls are answered automatically: • As audio if `auto_accept_video=0` • As audio with video if the call has video and `auto_accept_video=1`
	`auto_answer=0`	Incoming calls are answered manually and the user explicitly selects whether or not video streams should be accepted (using the `has_video` parameter supplied in the `gs_session_info` argument)
Either microphone or speaker is not available	No auto-answer is possible in this sub-case, so the `auto_answer`	Auto-rejection is not applicable, so the

• Incoming calls can be answered manually—it is assumed that the agent will plug in the headset (or use an available non-headset device, if applicable) before answering the call • Outgoing calls can be initiated—it is the agent's responsibility to ensure that the appropriate audio devices are available before the call is answered by the remote side	setting is not used	`reject_session_when_headset_na` setting is not used

SIP Endpoint SDK for OS X QuickStart Application

> **Important**
>
> The following features are not supported:
>
> - Video
> - IPv6
> - NAT, ICE, TURN, STUN
> - SIP Cluster
> - SIP Proxy

The easiest way to start using the SIP Endpoint SDK for OS X is with the bundled QuickStart application. This application ships in the same folder as the SDK and is supplied as both a double-clickable application in the `/Bin` folder and as source code in the form of an Xcode project located in the `QuickStart/Src` folder.

Running the QuickStart Application

You can try out the QuickStart application by running `/Bin/QuickStart.app`:

1. Set your configuration settings by editing `/Bin/SipEndpoint.config`.
2. Open and run `/Bin/QuickStart.app`.

Rebuilding the QuickStart Application from the Xcode Project

You can modify the QuickStart application by opening the QuickStart Xcode project, `/QuickStart/Src/QuickStart.xcodeproj`.

Before running your modified QuickStart application, you need to rebuild it in Xcode.

At that point, you can either run the application from within Xcode or you can double-click the application that is contained in the `/Bin` folder. When you rebuild the QuickStart project, the `/Bin/Quickstart.app` file is replaced with your modified QuickStart application.