SIP Endpoint SDK Developer's Guide

SIP Endpoint SDK 8.1.2OSX

2/11/2022

Table of Contents

Developer's Guide

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

- The architecture of the SIP Endpoint SDK for Apple OS
- Its configuration options
- How to design applications that make use of the powerful features of this SDK

Before reading this material you may want to:

- Install the SIP Endpoint SDK on your Macintosh.
- Ensure you have access to the latest version of the API Reference.
- Download the latest version of the Release Note (using links on the SIP Endpoint SDK for Apple OS Product Page) to see the most recent news and updates about this product.

Configuring SIP Endpoint SDK for Apple OS

The sample application that comes with the SIP Endpoint SDK for Apple OS distribution includes a property list (plist) that is used for configuring the application. This file is located at `<SIP Endpoint SDK Installation Folder>/Sample/Src/Sip EP Sample.plist`.

Note: Genesys recommends that you use the sample application as the starting point for your development efforts.

If you are developing applications from scratch, you should:

1. Create a copy of the plist from the sample application
2. Name it appropriately
3. Place it in your app's `Src` folder.

SIP Endpoint Configuration Settings

You can customize the following settings in your SIP Endpoint SDK applications.

Section	Setting	Values	Description
GSDefaultConnectionPolicy			
	networkInterface	String	Name of the network interface
GSDefaultDevicePolicy			
	audio_in_device	String	Microphone device
	audio_out_device	String	Speaker device
	use_headset	Boolean	If set to YES, the SDK uses a headset as the preferred audio input and output device.
GSDefaultEndpointPolicy			
	audioQos	Number	The integer value representing the DSCP bits to set for RTP audio packets.
	includeOSVersionInUserAgentHeader	Boolean	If set to YES, the user agent field includes the OS version the client is currently running on. Default: `NO`.
	ip_version	IPv4 IPv6	A value of `IPv4` means that the application selects an available

Section	Setting	Values	Description
		IPv4,IPv6 IPv6,IPv4 empty	local IPv4 address; IPv6 addresses are ignored. A value of IPv6 means that the application selects an available local IPv6 address; IPv4 addresses are ignored. A value of IPv4,IPv6 or an empty value means that the application selects an IPv4 address if one exists. If not, an available IPv6 address are selected. A value of IPv6,IPv4 means that the application selects an IPv6 address if one exists. If not, an available IPv4 address are selected. Default: IPv4,IPv6. NOTE: This parameter has no effect if the public_address option specifies an explicit IP address. NOTE: Although the IPv6 configuration options are available in the plist file for this product, they are not supported in the 8.1.2 Release of SIP Endpoint SDK for Apple OS.
	public_address	String	Local IP address or Fully Qualified Domain Name (FQDN) of the machine. NOTE: Although the IPv6 configuration options are available in the plist file for this product, they are not supported in the 8.1.2 Release of SIP Endpoint SDK for Apple OS.
	rtpInactivityTimeout	Number	Timeout interval for RTP inactivity. Valid values are integers from 0 to 150. A value of 0 or values greater than 150 mean that this feature is not activated. A value in the range of 1 to 150 indicates the inactivity timeout interval in seconds. Default: 0.
	rtpPortMin	Number	The integer value representing the minimum value for an RTP port range. Must be within the valid port range of 9000 to 65535. If the minimum and maximum values are

Section	Setting	Values	Description
			not specified or are set to an invalid value, the default minimum (9000) and maximum (minimum value + 999) are used. 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.
	rtpPortMax	Number	The integer value representing the maximum value for an RTP port range. Must be within the valid port range of 9000 to 65535. If the minimum and maximum values are not specified or are set to an invalid value, the default minimum (9000) and maximum (minimum value + 999) are used. 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.
	secureSignalingQos	Number	The integer value representing the DSCP bits to set for TCP packets.
	signalingQos	Number	The integer value representing the DSCP bits to set for SIP packets.
	sipPortMin	Number	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 and maximum values are not specified or are set to an invalid value, the default minimum (5060) and maximum (minimum value + 6) are used. Setting the

Section	Setting	Values	Description
			minimum to a value that is larger than the maximum is considered an error and will result in a failure to initialize the endpoint.
	sipPortMax	Number	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 minimum and maximum values are not specified or are set to an invalid value, the default minimum (5060) and maximum (minimum value + 6) are used. 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.
	videoQos	Number	The integer value representing the DSCP bits to set for RTP Video packets.
GSDefaultSessionPolicy			
	AGC_mode	0 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_accept_video	Boolean	If set to YES, all incoming video should be accepted automatically.

Section	Setting	Values	Description
			NOTE: The video mode window will not be opened if `auto_accept_video` and `auto_answer` are both set to `0`.
	auto_answer	Boolean	If set to YES, all incoming calls should be answered automatically. NOTE: The video mode window will not be opened if `auto_accept_video` and `auto_answer` are both set to `0`.
	dtmf_method	Rfc2833 Info InbandRtp	Method to send DTMF
	reject_session_when_headset_na	Boolean	If set to YES, the SDK should reject the incoming session if a USB headset is not available.
	sip_code_when_headset_na	Number	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.
basic: account connection details			
	regInterval	Number	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`.
	registrationTimeout	Number	The period, in seconds, after which registration should expire. A new

Section	Setting	Values	Description
			REGISTER request will be sent before expiration. Valid values are integers greater than or equal to 0. If the setting is empty or negative, the default value is 1800 seconds. If the setting is 0, registration is disabled, putting the endpoint in standalone mode.
	transport	udp tcp tls	The transport protocol to use when communicating with server
	server	String	The server address and port
	stun_server	String	STUN server address (with optional port). An empty or null value indicates this feature is not being used.
	turn_password	Number	Password for TURN authentication
	turn_server	String	TURN server address (with optional port). An empty or null value indicates this feature is not being used.
	turn_userName	String	User ID for TURN authorization
	user	String	User ID for this connection
basic: account mailbox details			
	server	String	Proxy server address and port for this mailbox
	timeout	Number	Registration timeout interval
	transport	udp tcp tls	Transport protocol to use when communicating with server
	user	String	User ID for this mailbox
codecs: priority			
	<Codec Name>	Number	Codec priority in SDP. A higher number means the codec has preference in codec

Section	Setting	Values	Description
			negotiation.
diagnostics			
	enable_logging	Boolean	Enable or disable logging
	log_file	String	Log file name, for example, `SipEndpoint.log`
	log_level	debug info warn error fatal	Log levels
	Log_option_provider	#/Empty gsip=3,webrtc=(state,warning)	If set to `#/Empty`, log messages will not include webrtc messages. If set to `gsip=3,webrtc=(state,warning)` and the level is set to `info`, log messages will include webrtc messages.
	logger_type	file default	If set to `file`, the log data will be printed to the file specified by the `log_file` parameter. If set to `default`, the log data will be printed to the console.
security			
	ca_list_file	String	Certificate of Authority (CA) list file
	cert_file	String	Public endpoint certificate file, which is used as client-side certificate for outgoing TLS connection and server-side certificate for incoming TLS connection
	method	unspecified tlsv1 sslv2 sslv3 sslv23	Security method
	password	String	Password to open private key
	privkey_file	String	Path to the optional private key file of the

Section	Setting	Values	Description
			endpoint certificate to be used. Example: `/usr/local/ssl/certs/example_priv_key.pem`.
	require_client_cert	Boolean	Indicates whether a client certificate is required. Default: `NO`.
	server_name	String	Server name. Default: empty.
	srtp_secure_signaling	no yes sips	Indicates whether SRTP secure signaling is to be used
	timeout	Number	Timeout interval. Default: `0`.
	tls_enabled	Boolean	If set to YES, connection with TLS transport will be registered. Default: `NO`.
	use_srtp	disabled optional mandatory	Indicates whether to use SRTP
	verify_client	Boolean	Indicates whether clients must be verified. Default: `NO`.
	verify_server	Boolean	Indicates whether servers must be verified. Default: `NO`.

Specifying Behavior When A USB headset Is Not Available

The following behaviors can now be specified when a SIP Endpoint user does not have a working USB headset:

- Whether SIP Endpoint should automatically reject an incoming call
- The SIP error code to be sent to the inviting party

Support for this feature involves several configuration settings:

- endpoint:GSDefaultDevicePolicy:use_headset
- endpoint:GSDefaultSessionPolicy:reject_session_when_headset_na
- endpoint:GSDefaultSessionPolicy:sip_code_when_headset_na

Information about these settings is available in the table of SIP Endpoint Configuration Settings that appears elsewhere on this page.

You can tell whether SIP Endpoint has been instructed to use a USB headset by using the following method of `GSDevicePolicyDelegate`:

```
- (BOOL) useHeadset;
```

To determine whether SIP Endpoint will reject an incoming session when a USB headset is not available or to determine which SIP error code is sent if a USB headset is not available, use the following methods of `GSSessionPolicyDelegate`:

```
- (BOOL) rejectWhenHeadsetNa:(id<GSSession>) session;
- (NSString*) sipCodeWhenHeadsetNa:(id<GSSession>) session;
```

Configuring Message Waiting Indicator (MWI) Support

A Message Waiting Indicator (MWI) is usually an audio or visual signal that a voicemail or other type of message is waiting. SIP Endpoint SDK's MWI support involves several configuration settings:

- endpoint:basic:mailbox:user
- endpoint:basic:mailbox:server
- endpoint:basic:mailbox:transport
- endpoint:basic:mailbox:timeout

Information about these settings is available in the table of SIP Endpoint Configuration Settings that appears elsewhere on this page. You can use these settings to have SIP Server notify your application when new messages have been received by the subscribing mailbox. `GSMessageWaitingIndicationService` provides the following methods to control mailbox notification subscriptions:

```
-(GSResult) subscribeForMailbox:(GSMessageWaitingIndicationSubscription*) subscription;
-(GSResult) unsubscribeForMailbox:(GSMessageWaitingIndicationSubscription*) subscription;
```

Notifications are provided by `GSMessageWaitingIndicationNotificationDelegate`. Access to the MWI summary is provided by the following method:

```
- (void) state:(GSMessageWaitingIndicationState*)
         state forSubscription:(GSMessageWaitingIndicationSubscription*) subscription
```

These notifications encapsulate the following information:

```
subscription = theSubscription;
messagesWaiting = theMessagesWaiting;
messageSummary = theMessageSummary;
```

SIP Endpoint SDK OS X Sample Application

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

Running the Sample Application

Before running the sample application, you need to rebuild it in Xcode. To do that:

1. Open the `SipEndpoint Sample.xcodeproj` project file contained in the sample's `Src` folder.
2. Rebuild the project.

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.