SIP Endpoint SDK Deployment Guide

SIP Endpoint SDK 8.5.2NET

12/30/2021

Table of Contents

Deployment Guide

Installation

This deployment guide can be used to install SIP Endpoint SDK on your Windows system and verify the installation. It includes the following information:

- Deployment Information—Details related to the SIP Endpoint SDK installation, including prerequisites and links to related information. Genesys recommends reading this page before beginning your installation, to ensure that your system meets the minimum requirements for the SIP Endpoint SDK.
- Installation procedures
- Verification procedures—This includes tasks for walking through the installation process and verifying that components were installed correctly.

Next Steps

After you have successfully installed the SIP Endpoint SDK, you might want to do the following:

- Download the latest version of the Release Note (using links on the SIP Endpoint SDK Product Page) to see the most recent news and updates about this product.
- Review the pages on using the included .NET QuickStart Application.
- Find out more about how to configure SIP Endpoint SDK for .NET.
- Read the SIP Endpoint SDK API Reference for detailed information about the SIP Endpoint SDK.

New In This Release

Check out the new features that have been added in the latest releases of SIP Endpoint SDK for .NET.

The following new features were added in the 8.5.2 release:

Release 8.5.200.17

- NAT traversal support
- Support for the SIP REFER method
- Ring tone support
- Support for the VP9 codec
- Updated logging mechanism
- Migration Guide for SIP Endpoint SDK 8.1.1

SIP Endpoint for .NET Deployment Information

Introduction

For the 8.x release, the SIP Endpoint SDK allows you to develop applications by using .NET technology.

To assist you with development, the SIP Endpoint SDK is packaged with a SIP Endpoint SDK API Reference document (`SipEndpointNet.chm`) that allows you to find reference information, coding recommendations, and code snippets in a single location.

For your convenience, the SIP Endpoint SDK also includes a Visual Studio Starter Kit that contains a project template and code snippets. This Starter Kit can help you get up and running during early application development.

Finally, every Genesys product also includes a Release Note that provides any late-breaking product information that could not be included in the manual. This product information can often be important. To view it, open the `read_me.html` file in the application home directory, where you will find a link to the latest Release Note for this product.

What You Should Know

This guide is written for software developers and application architects that have an understanding of the Genesys platform and the basics of SIP telephony before using this SDK. Before working with the SIP Endpoint SDK, you should know how to use the logging functionality of the Genesys Log Library.

In addition, the following document can be useful in understanding the Genesys SIP server environment:

- Framework 8.0 SIP Server Deployment Guide

Environment Prerequisites

Supported Operating Systems

- Windows 8 32-bit and 64-bit
- Windows 7 32-bit and 64-bit

Other Prerequisites

To work with Release 8.5.0 of the SIP Endpoint SDK for .NET, you must ensure that your system meets the software requirements established in the Genesys Supported Operating Environment Reference Manual, as well as meeting the following minimum software requirements:

Prerequisites to develop with SIP Endpoint SDK for .NET

- Genesys SIP Server 7.6.x, 8.0.2, or higher
- Genesys Voice Platform (GVP) 8.0 or higher is required for video conference support, which makes use of the GVP Resource Manager and Media Control Platform
- Microsoft .NET Framework 4.0
- Microsoft Visual Studio® .NET 2013 or higher

Prerequisites to run an application built with SIP Endpoint SDK for .NET

- Microsoft .NET Framework 4.0
- Microsoft Visual C++ 2005 Service Pack 1 Redistributable Package MFC Security Update
- Microsoft Visual C++ Redistributable for Visual Studio 2013

Related Resources

- SIP Endpoint SDK Developer's Guide

Installing SIP Endpoint SDK for .NET

Prerequisites

- Check the list of environment prerequisites, and confirm that your system meets these standards prior to installing SIP Endpoint SDK.

> **Important**
> You must have administrator authority in order to install SIP Endpoint SDK.

Procedure

Start of procedure

1. Run the installation file named `setup.exe` located in the `\SIPEndpointSDK\DotNet\windows\` directory on your product CD. The `Genesys Installation Wizard` is displayed to guide you through the installation and setup process.
2. Click `Next` at the `Welcome` dialog to display the `Genesys License Agreement` dialog.
3. Check the `I accept Genesys License Agreement` box to accept the conditions of the agreement.
4. Click `Next` at the `Genesys License Agreement` dialog. The `Choose Destination Location` dialog is displayed, showing the default destination, `C:\Program Files\GCTI\SIP Endpoint SDK`.
5. Click `Next` if you want to accept the default destination folder that is specified. If you prefer to install the SIP Endpoint SDK in a different location than the default directory, complete the following steps:
    1. Click `Browse` to open the `Choose Folder` dialog.
    2. Navigate to and select a directory path.
    3. Click `OK` to return to the `Choose Destination Location` dialog.
    4. Click `Next` to accept the destination folder that you have selected.
6. At the `Ready to Install` dialog, click `Install`. The Wizard installs the SIP Endpoint SDK, and all associated files, in the directory you selected. When the installation is finished, the `Installation Complete` dialog appears.
7. Click `Finish`.

End of procedure Next Steps

- To review the installation and confirm the location of your SIP Endpoint SDK files, continue by verifying the installed components.

Verifying Installed SIP Endpoint SDK for .NET Components

Prerequisites

- You must first complete the procedure that is found at Installing SIP Endpoint SDK for .NET.

Procedure

1. Expand the archive containing the SIP Endpoint SDK installation.
2. Examine each folder (including the root installation folder) to confirm its contents. The SIP Endpoint SDK Folder Contents table below gives a description of the expected result.

SIP Endpoint SDK Folder Contents

Folder	Contents
\Bin	This directory contains all of the binaries that are needed when working with SIP Endpoint SDK.
\Configuration	This directory contains the template for the SIP Endpoint SDK configuration file, `SipEndpoint.config`.
\Doc	This directory contains the SIP Endpoint SDK API Reference (`SipEndpointNet.chm`), which has detailed information about the structure and usage of the SIP Endpoint SDK.
\QuickStart	Visual Studio source files for the SIP Endpoint SDK QuickStart application.
\QuickStartExe	A compiled and ready-to-run version of the SIP Endpoint SDK QuickStart application.

Next Steps

- None

SIP Proxy Support

SIP Endpoint SDK can now be used with Genesys SIP Proxy, which provides high availability without requiring a virtual IP address.

Configuration

Domain Names

SRV records are not currently supported by SIP Endpoint SDK, which supports only A (or AAAA) records. Because of this, Genesys SIP Proxies should be configured with a single fully qualified domain name (FQDN) that resolves via DNS into multiple IP addresses via A (or AAAA) records in DNS server.

For information on how to configure SIP Proxy, consult the SIP Proxy Deployment Guide.

SIP Transaction Timeout

Name	Description	Valid Values	Default	Recommended
sip_transaction_timeout	SIP transaction timeout value in milliseconds.	1 through 32000	32000 (32 seconds)	**4000 (4 seconds)**

Because the default SIP transaction timeout value is too long for use by customer-facing applications, **Genesys strongly recommends that you set the `sip_transaction_timeout` value to `4000`, as shown here**:

```
<domain name="policy">
  <section name="endpoint">
       ...
    <setting name="sip_transaction_timeout" value="4000"/>
       ...
  </section>
```

Limitations

- Genesys SIP Proxy does not support scenarios involving switchovers in mid-transaction. Because of this, call answer and `CANCEL` may not work properly in such situations. In particular, the incoming call cannot be answered under these circumstances and must be released. Also, the outgoing call may be stuck on SIP Server for an unpredictable length of time. (Please note that this is a limitation of Genesys SIP Proxy and not of SIP Endpoint SDK.)
- You must configure the `reg_interval` parameter (in the SIP Endpoint configuration file) to a positive value (for example, `30`) if you want your SIP Endpoint to resend REGISTER and SUBSCRIBE messages to a new SIP Proxy when the current SIP Proxy is down. If you are using TCP as your transport protocol,

however, SIP Endpoint does send REGISTER and SUBSCRIBE messages to a new SIP Proxy immediately even if the `reg_interval` parameter is configured to `0`, because TCP supports immediate reconnection or recovery to a new SIP Proxy.

- The SIP Endpoint SDK always retries INVITE once, regardless of the number of proxies configured.

Technical Background

The following background information describes certain features of the SIP Endpoint SDK internals that might be helpful in planning your application.

> **Warning**
> This information is subject to change without notice and is not supported by Genesys.

High Availability

Because the SIP Endpoint SDK already supports DNS queries for each transaction, the load balancing aspect of High Availability is already taken care of, although it may require configuring an IP address rotation in DNS server. The missing parts for High Availability support are:

- **Temporarily blacklisting the IP address on failure (no response), so the next SIP message is sent to a different proxy:** The proposed method for this assigns a specific penalty to each IP address in the list from the DNS, based on how recently that IP address has failed. When a selection must be made, the one with the lowest penalty value is chosen. If there are no penalty-free IP addresses, the algorithm chooses the least-recently blacklisted address, which may now be available. Because of this algorithm, there is not much sense in making the blacklist interval configurable, and it is currently hard-coded to 10 minutes. (After that, the address is fully cleared and can be tried again).

- **You must set the `sip_transaction_timeout` parameter to a value less than 32000 milliseconds**, as described in the Configuration section above, with the recommended setting being `4000`.

> **Important**
> The `sip_transaction_timeout` setting specifies the maximum interval for a transaction to wait for a response. However, a failure may occur before this timeout interval has elapsed. In particular, when a TCP connection has been used for SIP transport and the active SIP Proxy terminates or is shut down, the broken connection may be detected immediately.
> In this case, the endpoint does not have to wait for a timeout expiration in order to switch to another proxy. On the other hand, if the host that is running the SIP Proxy has been shut down, or unplugged from the network, the timeout will be applied, since there has not been any immediate indication from the network that the connection is not operable.

- **Automatic re-transmission of failed (timed out) SIP messages to a new proxy:** Given the current GSIPLIB architecture, this re-transmission must be done separately for each message type. Thus, it must be tested for all possible use cases, notably:
    - REGISTER and SUBSCRIBE renewal—by design, switching to a new SIP proxy obeys the configured `reg_interval` parameter, so if re-registration is disabled by a value of `0`, the endpoint does not resend the REGISTER or SUBSCRIBE message.
    - Initial INVITE to be retried once to a different SIP proxy (and reported as failed in case of a double failure).
      Note: To give application code full visibility to the SIP call ID , an `in-progress` state is reported twice for the same session ID (with the state reported as `disconnected` in between them).
    - Mid-call INVITE for Hold and Retrieve operations to be retried once, transparent to application code.
    - Retrying a 200 OK response to an initial INVITE (answering the call) and call-terminating BYE and CANCEL requests work differently from other requests. These retry operations work only when the `sip_transaction_timeout` parameter is set to a value lower than 32000 milliseconds (as described in the Configuration section above), with the recommended setting being `4000`. These requests are retried continuously for 32 seconds total (cycling throught the list of configured proxies), after which the call is abandoned.

ICMP Messages

It is not currently possible to intercept ICMP messages using GSIPLIB, because exceptions are processed on the transport level, but the reaction must be implemented on the transaction level and there is no easy way to pass control between those two levels. Because of this, these failures may be detected in the current release by timeout only. This is not much of a limitation, however, as ICMP messages are generated only when:

- The server is on Windows
- No firewall is blocking them
- The host is alive

Therefore, these messages are unreliable and their use would add very little to the timeout method.

8.1.1 Migration Guide

This migration guide will help you update your SIP Endpoint SDK 8.1.1 configuration settings for use with SIP Endpoint SDK 8.5.2. SIP Endpoint SDK 8.5.2 is based on Genesys SIP and WebRTC media stacks. It has a different XML structure and different settings from SIP Endpoint SDK for .NET 8.1.1 which was based on CounterPath.

Changes in Configuration File Structure

The table below show a side by side comparison of the SIP Endpoint SDK 8.1.1 and 8.5.2 configuration file XML structures. Notice that 8.5.2 configuration is based on policy settings rather than on environment specifics as in 8.1.1. The XML examples below show detail up to the section level where the actual settings are located.

SIP Endpoint SDK 8.1.1 Configuration	SIP Endpoint SDK 8.5.2 Configuration
<?xml version="1.0" encoding="utf-8" ?> <SipEndpoint xmlns="http://schemas.genesyslab.com/2009/ sipendpoint"> <Container name ="Basic"> <Connectivity user ="DN_0" server="SERVER_0:PORT_0" protocol="udp"/> </Container> <Container name ="Cp"> <domain name="audio"> <section name="headset"> </section> <section name="incoming"> </section> <section name="vad"> </section> </domain> <domain name="system"> <section name="qos"> </section> <section name="dtmf"> </section> <section name="network"> </section> <section name="diagnostics"> </section> <section name="general"> </section> <section name="indialog_notify"> </section> <section name="call_stats"> </section> </domain> <domain name="rtp"> <section name="2833">	<?xml version="1.0" encoding="utf-8" ?> <SipEndpoint xmlns="http://schemas.genesyslab.com/2009/ sipendpoint"> <Container name ="Basic"> <Connectivity user =" DN_0" server=" SERVER_0:PORT_0" protocol="udp"/> </Container> <Container name ="Genesys"> <domain name="policy"> <section name="endpoint"> </section> <section name="session"> </section> <section name="device"> </section> </domain> <domain name="codecs"> <section name="PCMU/8000"> </section> <section name="PCMA/8000"> </section> <section name="G722/16000"> </section> <section name="iLBC/8000"> </section> <section name="iSAC/32000"> </section> <section name="iSAC/16000"> </section> <section name="vp8"> </section> <section name="g729/8000"> </section> <section name="h264"> </section>

SIP Endpoint SDK 8.1.1 Configuration	SIP Endpoint SDK 8.5.2 Configuration
</section> <section name="inactivity"> </section> </domain> <domain name="proxies"> <section name="proxy0"> </section> <section name="proxy1"> </section> </domain> <domain name="genesyslab"> <section name="device"> </section> <section name="system"> </section> <section name="beeptone"> </section> <section name="dtmf"> </section> <section name="control"> </section> </domain> </settings> </Container> </SipEndpoint>	<section name="vp9"> </section> <section name="ulpfec/ 90000"> </section> <section name="opus/48000/ 2"> </section> </domain> <domain name="proxies"> <section name="proxy0"> <section name="mailbox"> </section> <section name="nat"> </section> </section> </domain> <domain name="system"> <section name="diagnostics"> </section> <section name="security"> </section> <section name="media"> </section> </domain> </settings> </Container> </SipEndpoint>

Changes in Configuration Settings

The table in this section shows how 8.1.1 configuration settings map to 8.5.2 configuration settings.

Dot Notation

For convenience, 8.5.2 configuration settings paths are presented in dot notations. For example:

XML Fragment

```
<Container name ="Genesys">
      <domain name="policy">
        <section name="endpoint">
          <setting name="sip_port_min" value=""/>
```

Dot Notation of XML Fragment

```
Genesys.policy.endpoint.sip_port_min
```

Basic Container

The basic container for SIP EndPoint SDK 8.5.2 is as follows:

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"/>
```

Cp Container

The `Cp` container has been replaced by the `Genesys` container. Use the following table to to map `Cp` container settings to `Genesys` container settings.

Domain	Section	Setting	8.5.2 Setting	8.5.2 Valid Values
audio				
	headset	audio_in_agc_enabled	`Genesys.policy.session.agc_mode`	Valid Values: 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.
	incoming	use_agc	Discontinued	
	vad	continue_sending_from_last_activity_in_milliseconds	`Genesys.policy.session.vad_level`	Voice Activity Detection: 0-3 From conventional VAD, lowest reduction To aggressive VAD, highest reduction
system				
	qos	audio	Discontinued	
	dtmf	force_send_in_band	`Genesys.policy.session.dtmf_method`	Dual-tone multi-frequency: 0 – InbandRtp 1 - rfc2833

Domain	Section	Setting	8.5.2 Setting	8.5.2 Valid Values
				2 - Info
		minimum_rfc2833_play_time	Discontinued	
	network	dtx_enabled	`Genesys.policy.session.dtx_mode`	Discontinuous Transmissions: 0 – DTX is OFF 1 – DTX is ON
	diagnostics	enable_logging	`Genesys.system.diagnostics.enable_logging`	Enable logging: 0 - logging is disabled 1 - logging is enabled
		log_level	`Genesys.system.diagnostics.log_level`	Logging Level: 0 - Fatal messages 1 - Error messages 2 - Warning messages 3 - Info messages 4 - Debug messages
	general	add_OS_version_to_user_agent_header	`Genesys.policy.endpoint.include_os_version_in_u`	Include OS version to the User Agent header: 0 – OS version is not included 1 – OS version is included
	indialog_notify	enable_indialognotify	Discontinued	
	call_stats	enabled	Discontinued	
		has_shown_user	Discontinued	
		url	Discontinued	
rtp				
	2833	enabled	Discontinued	
		hold_over_time_in_ms	Discontinued	
		packet_time_in_ms	Discontinued	
		payload_number	Discontinued	
	inactivity	timer_enabled	`Genesys.policy.endpoint.rtp_inactivity_timeout`	RTP Inactivity Timeout in seconds: 0 no inactivity detection the default value 1 - 150sec inactivity timeout interval
proxies				
	proxy0	sip_port_range_enable	Discontinued	

Domain	Section	Setting	8.5.2 Setting	8.5.2 Valid Values
		sip_port_range_min	`Genesys.policy.endpoint.sip_port_min`	SIP port range Minimum limit: valid values are 1 to 65535 default is 5060
		sip_port_range_max	`Genesys.policy.endpoint.sip_port_max`	SIP port range Maximum limit: Valid values: 1 to 65535 Default: Minimum + 6
		port_range_enable	Discontinued	
		port_range_min	`Genesys.policy.endpoint.rtp_port_min`	RTP port range Minimum limit: Valid values: 1 to 65535 Default: 9000
		port_range_max	`Genesys.policy.endpoint.rtp_port_max`	RTP port range Maximum limit: valid values are 1 to 65535 default is 9999
		reregister_in_seconds	`Genesys.proxies.proxy0.reg_timeout`	SIP Registration Timeout: Valid Values: Number greater than 0 in seconds; 0 to disable registration
		auto_answer_audio	`Genesys.policy.session.auto_answer`	Auto Answer: 0 – disabled (default value) 1 – enabled
		auto_answer_video	`Genesys.policy.session.auto_accept_video`	Auto Accept Video: 0 – disabled (default value) 1 – enabled
		media_encrypted	`Genesys.system.security.use_srtp`	Use SRTP for particular session valid values: • disabled - SRTP is disabled (default) • optional or allowed - SRTP is allowed • elective or both

Domain	Section	Setting	8.5.2 Setting	8.5.2 Valid Values
				– SRTP is allowed in both direction • force or mandatory - SRTP is forced • enabled - SRTP is enabled
Genesyslab				
	device	use_headset	Genesys.policy.device.use_headset	Use Headset: 0 – disabled (default value) 1 - enabled
		reject_call_when_headset_na	Genesys.policy.session.reject_session_when_head	Reject Session if headset is not available: 0 – disabled (default value) 1 – enabled
		error_code_when_headset_na	Genesys.policy.session.sip_code_when_headset_na	SIP Code Message: SIP error code (there is no default value) 480 is recommended value
		error_message_when_headset_na	Discontinued	
		headset_name	Genesys.policy.device.headset_name	Headset name: String
		manual_audio_devices_configure	Discontinued	
		audio_in_device	Genesys.policy.device.audio_in_device	Microphone name: String
		audio_out_device	Genesys.policy.device.audio_out_device	Speaker name: String
		ringer_device	Discontinued	
	system	export_settings	Discontinued	
		enable_export_settings	Discontinued	
		log_level_AbstractPhone	Discontinued	
		log_level_Audio		
		log_level_Auto Configuration		

Domain	Section	Setting	8.5.2 Setting	8.5.2 Valid Values
		log_level_CCM		
		log_level_Conferencing		
		log_level_Contacts		
		log_level_DNS		
		log_level_GUI		
		log_level_Jitter		
		log_level_Licensing		
		log_level_Media		
		log_level_Privacy		
		log_level_RTP		
		log_level_STUN		
		log_level_Security		
		log_level_Storage		
		log_level_Transport		
		log_level_USB Devices		
		log_level_Utilities		
		log_level_Video		
		log_level_Voice Quality		
		log_level_XMPP		
		log_level_Endpoint		
	beeptone	play_locally	Discontinued	
		enable_beeptone	Discontinued	
		beeptone_file	Discontinued	
		beeptone_timeout	Discontinued	
	dtmf	play_locally	Discontinued	
		pause_start_stop_dtmf	Discontinued	
	control	auto_answer	`Genesys.policy.session.auto_answer`	Auto Answer: 0 – disabled (default value) 1 – enabled

Code Snippet: Starting SIP Endpoint SDK 8.5.2

In order to run a WebRTC instance and take advantage of the ability to choose audio layers, Core or Wave needs to run the WebRTC component using MTA (MultiThreaded Apartment). The following code snippet demonstrates how to run SIP Endpoint SDK 8.5.2:

```
. . .
this.providerWebRtcWorker = new Thread(new ThreadStart(RunUpEndpoint));
this.providerWebRtcWorker.Name = "ProviderWebRtcWorker";
this.providerWebRtcWorker.SetApartmentState(ApartmentState.MTA);
this.providerWebRtcWorker.Start();
this.providerWebRtcWorker.Join();
. . .

private void RunUpEndpoint() {
this.endpoint.ApplyConfiguration(confDoc.Root);
this.endpoint.BeginActivate();

// At this point, the actual configuration can be updated and
// the Endpoint can be started.
// The next two lines of code are unnecessary if you are not
// going to change configuration settings.

// Get updated configuration settings
//confDoc = XDocument.Load("SipEndpoint.config");
//this.endpoint.ApplyConfiguration(confDoc.Root);

// The actual SIP Endpoint Start
this.Logger.Debug("this.endpoint.Start()");
this.endpoint.Start();
. . .
}
```