SIP Endpoint SDK Developer's Guide

SIP Endpoint SDK Disaster Recovery and Geo-Redundancy

12/14/2025

Contents

SIP Endpoint SDK Disaster Recovery and Geo-Redundancy

This article describes the ways in which SIP Server SDK supports high availability and resilience.

Introduction

The overall architecture for the disaster recovery/geo-redundancy solution is depicted below. Using this architecture, the SIP Endpoint SDK can connect to multiple sites in different geographical locations, providing redundancy and the potential for quick and efficient disaster recovery.

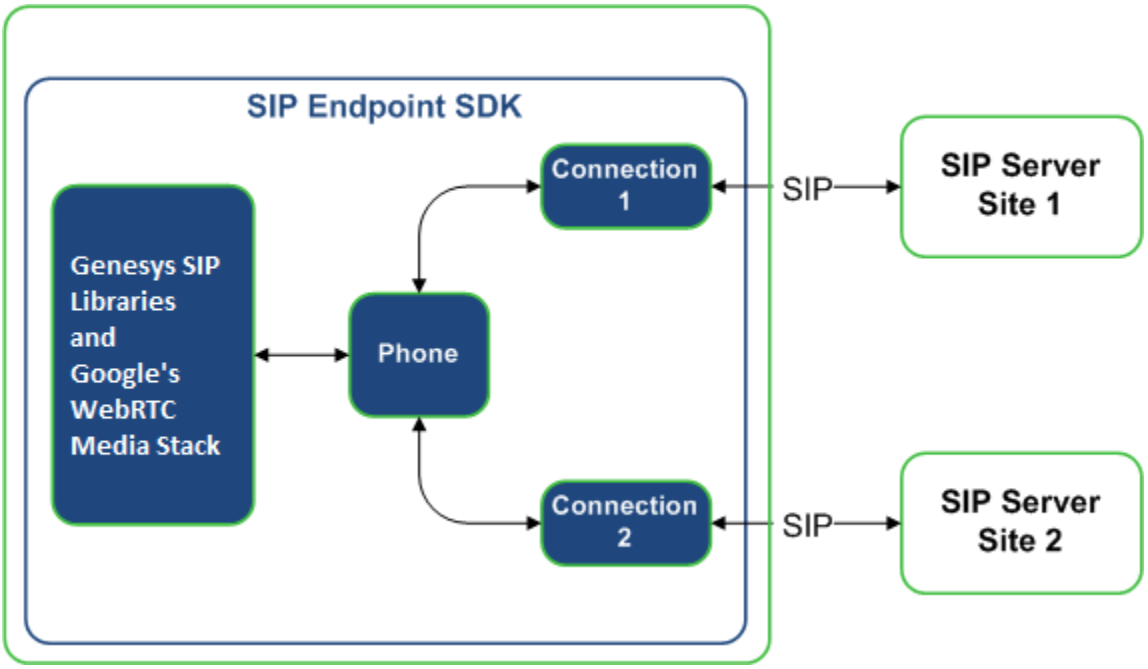


Figure 1: Geographical Redundancy and Disaster Recovery Architecture.

The SIP Endpoint SDK includes a QuickStart application that can be used to run functional testing. The QuickStart application starts SIP Endpoint SDK by creating a Phone instance and then using configuration settings to create two SIP connections and register them on two separate SIP Server sites.

To implement this type of architecture, you must use SIP Server release 8.1.0 or later.

How It Works

To enable Geo-Redundant architecture, the SIP Server SDK has been changed to allow multiple SIP Servers to be registered. Each SIP Server site requires a distinct connection to be made.

Connection Behaviors

Each connection created in the SIP Endpoint SDK has the following behavior:

SIP Server State	Connection Behavior
Available	Registers and works with the SIP Server normally.
Down or Unavailable	Continually attempts to register until the connection is either successful or explicitly removed. Once a specified SIP Server comes up after being down, then the SIP Endpoint SDK registers and works with the SIP Server normally as soon as it becomes available.

SIP Endpoint SDK will register separate connections on both SIP Server sites - even if they are associated with DNs that have the same name. On every registration renewal attempt, the SIP Endpoint SDK checks for successful registration to determine if that SIP Server is down or unavailable. Whenever there is a change in status for a connection, the SIP Server SDK generates a notification event that is sent to the application with information about the connection status. When the SIP Endpoint SDK is using multiple connections, equal priority is given to each connection. Registration on primary and secondary sites occurs simultaneously.

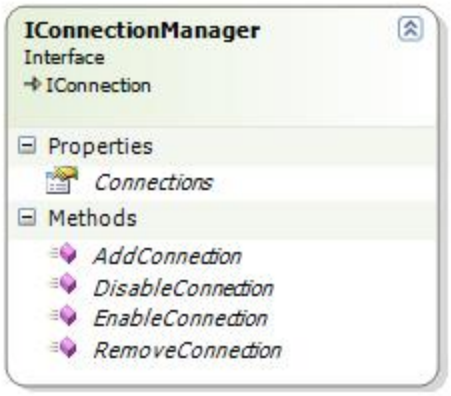
Working with Multiple Connections

When working with multiple connections, there are a few key items you should keep in mind:

- 1pcc outbound calls use a particular SIP Server site by dialing from an explicitly assigned connection. All other calls are put on hold. If the intended SIP Server site is down then the 1pcc outbound call is made from the second site.
- 3pcc inbound calls are received and answered using the corresponding connection, while all other calls are put on hold. (Note: The 2nd line is ringing while 1st line is on call.) If the first site is down or unavailable then the SIP Endpoint SDK receives 3pcc inbound calls by using second connection.
- If a call disconnection notification is received, check the reason why call was disconnected along with the connection ID. A call disconnection reason of "SipError" along with the message "408" or "Request Timeout" can be interpreted as a sign that the corresponding SIP Server is down or unavailable.

SIP Endpoint SDK API changes

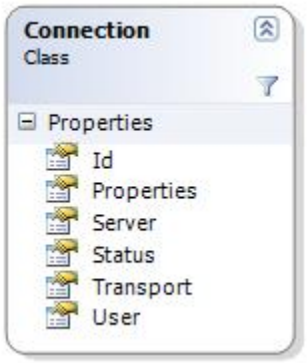
Handling Multiple Connections



To support handling more than one connection, the following methods have been added to IConnectionManager interface class that is implemented by CpProvider and ConnectionManager:

Description	Method Signature
Add a connection	void AddConnection(int connectionId);
Enable a connection	void EnableConnection (int connectionId);
Disable a connection	void DesableConnection (int connectionId);
Remove a connection	void RemoveConnection (int connectionId);
Get connections collection	ConnectionCollection Connections { get; }

Properties that can be accessed from the collection of connections are shown in the following class diagram:

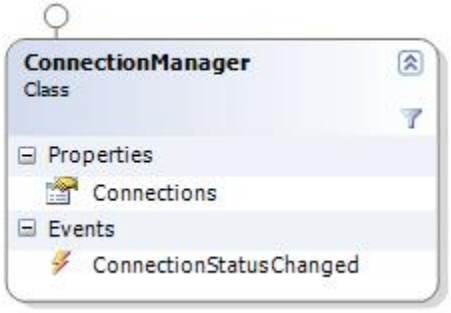


Managing Connection Status

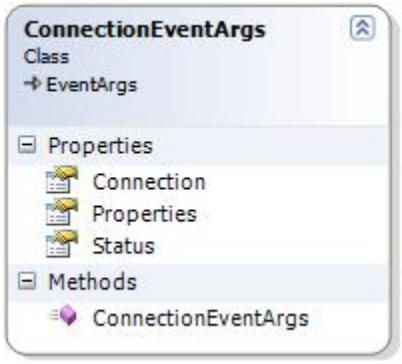
Notification of changes to the connection status is given by the ConnectionStatusChanged event.

```
event EventHandler<ConnectionEventArgs> ConnectionStatusChanged;
```

This event has been added to IConnection interface as shown below:



The event includes ConnectionEventArgs, which provides your application access to the following properties:



Finding the Connection ID in Session Manager

The SessionStatusChanged event in session manager now includes a *connectionID* property in the *SessionEventArgs.Session.Properties* collection, indicating which connection is available.

Making Calls Using First-Party Call Control

The *Dial* method of the *ICallControl* interface now includes a *connectionID* parameter that determines which connection will be used for outgoing 1pcc calls.

```
void Dial(int connectionId, String^ destination);
```

Configuration Settings

In addition to the basic configuration details you should be aware of, the following settings must be added to your SipEndpoint.config file to support Disaster Recovery and Geo-Redundancy.

Adding Multiple Connections

To support configuring more than one connection, the section: <Container name ="Basic"> is extended to have more than one Connectivity tag. See the following example for details:

```
<Container name ="Basic">
    <Connectivity user ="DN1" server="SipServer1:port1" protocol="Protocol1"/>
    <Connectivity user ="DN2" server=" SipServer2:port2" protocol=" Protocol2"/>
</Container>
```

Where:

- DN1 and DN2 are extension objects in CME
- SipServer1:port1 and SipServer2:port2 are SIP Server host names and ports
- Protocol1 and Protocol2 are supported transport protocols by SIP Endpoint SDK

Changing the Re-Registration Timeout Interval in SIP Endpoint SDK

The default SIP Endpoint SDK re-registration interval is 3600 seconds. However, you can redefine that interval by updating the SipEndpoint.config file to include the reregister_in_seconds setting. The example shown below demonstrates how different re-registration intervals can be specified for two connections, and the new configuration applied to your SIP Endpoint SDK applications. In this example, proxy0 and proxy1 are equivalent to the SIP Endpoint SDK connections with connection ID 0 and 1.

```
<domain name="proxies">
    <section name="proxy0">
        ...
        <setting name="reg_interval" value="10"/>
    </section>
    <section name="proxy1">
        ...
        <setting name="reg_interval" value="20"/>
    </section>
</domain>
```