3.11 Simplified Desktop .NET API Specification

Bright Pattern Documentation

Generated: 10/31/2024 11:54 pm

Table of Contents

Purpose

ServicePattern Simplified Desktop .NET API Specification describes the methods and events of the Simplified Desktop .NET API, which provides access to a number of functions of the ServicePattern Agent Desktop application from .NET-based third-party applications.

For more information about the Agent Desktop application, see *ServicePattern Agent Guide*.

Audience

This guide is intended for the IT personnel responsible for the data infrastructure of the ServicePattern-based contact centers. Readers of this guide are expected to have expertise in web application development as well as solid understanding of contact center operations.

General Information

The API supports the following functions:

- Making calls
- Getting notifications about initiated calls
- Getting notifications about incoming calls
- Getting notifications when both outgoing and incoming calls are released
- Pausing and resuming voice and screen recordings
- Getting notifications when recording is stopped and started

The API itself does not replace the softphone functionality; the ServicePattern Agent Desktop application with a softphone (or hardphone) must be running on the user's computer in order for the API to function.

The API communicates with the Agent Desktop locally and therefore an agent desktop session has to be started and the softphone plugin (Agent Desktop Helper Application) must be active.

The API uses identity of the user logged into the Agent Desktop application in the current Windows session.

Event Delivery

Communications with the softphone plugin are handled by the lowest level of the API; all transport is handled by the .NET platform and all transport-related events are delivered on threads maintained by the platform. Note that direct updating of the UI (either Windows Forms, or WPF) is only permitted on the UI thread - in most cases, the main thread of the application. Although different mechanisms exist to convey execution flow from non-UI to UI threads, the API has its own mechanism that allows receiving events emitted by the API classes on the UI thread - event dispatcher classes.

Each API class has a child class named eventDispatcher (referred to as "event dispatcher"). The event dispatcher class intercepts and re-emits all events emitted by its parent class, while guaranteeing that the events will be delivered on the thread on which the API has been created. Event handlers registered with the API's event dispatcher class created on the UI thread may directly update elements of the application's UI.

All events emitted by all objects follow the Microsoft's guideline that asks that all event handlers take two parameters - the object that has emitted the event and another object that encapsulates all arguments of the event. Such an arguments object should inherit the *System.EventArgs* class and provide any additional data or capabilities in the form of additional properties and methods. All events emitted by the event dispatcher objects carry the reference to the original sender object.

API Instantiation and Initialization

First, the client should create an *Api* object and attach event handlers:

```
m_api = new DesktopControlAPI.ApiLite();

m_api.eventDispatcher.evtApiUp += this.onApiUp;

m_api.eventDispatcher.evtApiDown += this.onApiDown;

m_api.eventDispatcher.evtCallOffered += this.onCallOffered;

m_api.eventDispatcher.evtCallDialing += this.onCallDialing;

m_api.eventDispatcher.evtCallDisconnected += this.onCallDisconnected;
```

Note that by default API logging is enabled:

```
ApiLite(bool _enableLog = true);
```

To disable logging, set enableLog to false.

```
m_api = new DesktopCOntrolAPI.ApiLite(false);
```

Once API is created and event handlers are attached, the client may initialize the API instance:

```
m_api.InitAPI();
```

Once API is connected to the softphone and the evtApiUp event is received, the client may dial calls:

```
m_api.CallDial(“4154556565”);
```

When application terminates the following method should be called to ensure clean termination of the TCP connection between API and the Agent Desktop:

```
m_api.ShutdownAPI();
```

API Methods

InitAPI

Syntax	bool InitAPI();

Description	Initializes the API. This request initializes the TCP connection procedure. Once connection is established the evtApiUp event is sent to the .NET application.

ShutdownAPI

Syntax	void ShutdownAPI();
Description	Terminates the TCP connection between API and the Agent Desktop.

CallDial

Syntax	UInt64 CallDial(string destination, string giid);
Description	Initiates a call to the number specified in *destination*. Specification of *giid* (global interaction identifier) is optional. Returns request ID.

MuteCallRecording

Syntax	UInt64 MuteCallRecording(string callId);
Description	Mutes audio recording of the call identified by *callId* on the given desktop. The recording will continue, but any voice signal will be replaced with silence.

UnmuteCallRecording

Syntax	UInt64 UnmuteCallRecording(string callId);
Description	Resumes previously muted audio recording for the call identified by *callId* on the given desktop.

MuteScreenRecordings

Syntax	UInt64 MuteScreenRecordings();
Description	Mutes screen recording on this desktop. For the period when screen recording is muted, the recording will contain a static snapshot of the desktop at the moment when mute was applied. This request is processed in the Agent Desktop Helper Application.

UnmuteScreenRecordings

Syntax	UInt64 UnmuteScreenRecordings();
Description	Unmutes previously muted screen recording on this desktop. This request is processed in the Agent Desktop Helper Application.

API Events

evtApiUp

Syntax	void evtApiUp(System.EventArgs args);
Description	Sent when the API is successfully connected (or reconnected) to the softphone. Note that no information is provided about calls which may already be present on the Agent Desktop.

evtApiDown

Syntax	void evtApiDown(System.EventArgs args);
Description	Sent when the ShutdownAPI method is called or when the API loses connection to the softphone. In case of connection loss, the API will try to reconnect every 15 seconds; once reconnected the *evtApiUp* event is sent.

evtCallDialing

Syntax	void evtCallDialing(DesktopControlAPI.CallArgs args);
Description	Sent when the softphone starts dialing a call. Note that this event reports calls both initiated by a prior CallDial method and dialed directly from the Agent Desktop. For parameters, see description of auxiliary class DesktopControlAPI.CallArgs.

evtCallOffered

Syntax	void evtCallOffered(DesktopControlAPI.CallArgs args);
Description	Sent when the softphone receives a new incoming call. For parameters, see description of auxiliary class DesktopControlAPI.CallArgs.

evtCallDisconnected

Syntax	void evtCallDisconnected(DesktopControlAPI.CallArgs args);
Description	Sent when an existing call is released. For parameters, see description of auxiliary class DesktopControlAPI.CallArgs.

onError

Syntax	void onError(DesktopControlAPI. ErrorEventArgs args);
Description	Sent when an error occurs while executing a method. For parameters, see description of auxiliary class DesktopControlAPI.ErrorEventArgs.

onCallRecordingStarted

Syntax	void onCallRecordingStarted(DesktopControlAPI.CallArgs args);
Description	Sent when call recording is started. For parameters, see description of auxiliary class DesktopControlAPI.CallArgs.

onCallRecordingStopped

Syntax	void onCallRecordingStopped(DesktopControlAPI.CallArgs args);
Description	Sent when call recording is stopped. For parameters, see description of auxiliary class DesktopControlAPI.CallArgs.

onCallRecordingMuted

Syntax	void onCallRecordingMuted(DesktopControlAPI.CallArgs args);
Description	Sent when call recording is muted. For parameters, see description of auxiliary class DesktopControlAPI.CallArgs.

onCallRecordingUnmuted

Syntax	void onCallRecordingUnmuted(DesktopControlAPI.CallArgs args);
Description	Sent when call recording is unmuted. For parameters, see description of auxiliary class DesktopControlAPI.CallArgs.

onScreenRecordingStarted

Syntax	void onScreenRecordingStarted(DesktopControlAPI.ScreenRecordingArgs args);
Description	Sent when screen recording is started. For parameters, see description of auxiliary class DesktopControlAPI.ScreenRecordingArgs.

onScreenRecordingCompleted

Syntax	void onScreenRecordingCompleted(DesktopControlAPI.ScreenRecordingArgs args);
Description	Sent when screen recording is stopped. For parameters, see description of auxiliary class DesktopControlAPI.ScreenRecordingArgs.

onScreenRecordingsMuted

Syntax	void onScreenRecordingsMuted(DesktopControlAPI. BaseEventArgs args);
Description	Sent when screen recording is muted. For parameters, see description of auxiliary class DesktopControlAPI.BaseEventArgs.

onScreenRecordingsUnmuted

Syntax	void onScreenRecordingsUnmuted (DesktopControlAPI.BaseEventArgs args);
Description	Sent when screen recording is unmuted. For parameters, see description of auxiliary class DesktopControlAPI.BaseEventArgs.

API Properties

connected

Type	boolean
Description	Used to check if API is connected to the Agent Desktop (True if evtApiUp was called before; False if evtApiDown was called or evtApiUp was never called).

calls

Type	System.Collections.Generic.Dictionary <string,DesktopControlAPI.Call>
Description	Collection of calls on Agent’s phone (there maybe more than one call). See description of auxiliary class DesktopControlAPI.Call.

Auxiliary Classes

DesktopControlAPI.BaseEventArgs

Description	Base event arguments class.

Members	- reqId – request ID returned by a call that caused the error

DesktopControlAPI.Call

Description	Describes a single phone call.
Members	- m_id – call ID; string - m_direction – call direction; enum; (DesktopControlAPI.CallDirection.Inbound for incoming calls; DesktopControlAPI.CallDirection.Outbound for outgoing calls) - m_ANI – caller ID number for incoming calls or Agent's phone number for outgoing calls; string - m_DNIS – number dialed by the calling party for incoming calls or number dialed by the Agent for outgoing calls; string - m_callerName – caller ID text name for incoming calls or Agent's display name for outgoing calls; string - m_serviceName – name of the service associated with this call; string - m_agentId – agent's username; string - m_agentPhone – agent's phone number; string

DesktopControlAPI.CallArgs

Description	Class for call attributes used in all call-related events.
Members	- callId – call ID; string - requestId – identifier of the request related to this event; string - ANI – caller ID number for incoming calls or Agent's phone number for outgoing calls; string - DNIS – number dialed by the calling party for incoming calls or number dialed by the Agent for outgoing calls - callerName – caller ID text name for incoming calls or Agent's display name for outgoing calls - serviceName – name of the service associated with this call - attachedData – key-value list of the data attached to this call by the scenario that processed it. Scenario should use the Attached Data block for this purpose. - requestId - Optional request ID of the method which the event responds to - globalInteractionId - global interaction identifier; string

DesktopControlAPI.ErrorEventArgs

Description	Error event arguments class.

Members	• reqId – request ID of the method that caused the error • callId – call ID; string • code - error code • description - error description

DesktopControlAPI.ScreenRecordingArgs

Description	Arguments of events related to screen recording.
Members	• sessionId – screen recording session ID (starting from version 3.9, all agent activities, including interaction handling, are recorded via a single screen recording session; i.e., this parameter has no purpose other than backward compatibility) • requestId - optional request ID of the method which the event responds to