A MITEL
PRODUCT
GUIDE

Unify OpenScape Contact Center Agile/Enterprise

Screen Pop API V11R1

Integration Guide
07/2022

A31003-S22B1-N101-01-7620

Mitel

Notices

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Europe Limited. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes. No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

Trademarks

The trademarks, service marks, logos, and graphics (collectively “Trademarks”) appearing on Mitel’s Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC) or its subsidiaries (collectively “Mitel), Unify Software and Solutions GmbH & Co. KG or its affiliates (collectively “Unify”) or others. Use of the Trademarks is prohibited without the express consent from Mitel and/or Unify. Please contact our legal department at iplegal@mitel.com for additional information. For a list of the worldwide Mitel and Unify registered trademarks, please refer to the website: http://www.mitel.com/trademarks.

Contents

1 About this guide

This guide provides an overview of the OpenScape Contact Center Screen Pop Application Programming Interface (API) and describes how to integrate the API with a custom application to display information about a contact in a screen pop on a user's desktop when a contact arrives.

1.1 Who should use this guide

This guide is intended for system integrators who want to integrate their custom application with the OpenScape Contact Center Screen Pop API.

1.2 Formatting conventions

The following formatting conventions are used in this guide:

Bold
This font identifies OpenScape Contact Center components, window and dialog box titles, and item names.

Italic
This font identifies references to related documentation.

`Monospace Font`
This font distinguishes text that you should type, or that the computer displays in a message.

NOTE: Notes emphasize information that is useful but not essential, such as tips or alternative methods for performing a task.

IMPORTANT: Important notes draw special attention to actions that could adversely affect the operation of the application or result in a loss of data.

1.3 Documentation feedback

To report an issue with this document, call the Customer Support Center.

When you call, be sure to include the following information. This will help identify which document you are having issues with.

- **Title**: Screen Pop API Integration Guide
- **Order Number**: A31003-S22B1-N101-01-7620

2 About the OpenScape Contact Center Screen Pop API

This chapter provides an overview of the OpenScape Contact Center Screen Pop Application Programming Interface (API).

2.1 Overview

The OpenScape Contact Center Screen Pop API is a 32-bit COM control that enables users to monitor an extension or user ID and display a screen pop in a third-party custom application when a contact arrives.

- If the contact is a call, the following scenarios are possible:
  - Call is routed to a user (see the example in Section 2.2.1, "Receiving a routed call", on page 9).
  - User consults on a call with another user (see the example in Section 2.2.2, "Consulting on a routed call", on page 11).
  - User requeues or transfers a call and the call is offered to another user (see the example in Section 2.2.3, "Requeuing a routed call", on page 12).
  - Call is made directly to the user's extension by an internal or external user.
  - Call is the result of conferencing or is transferred to the user by an IVR or another party.
  - Call is recalled (transferred to another extension; if the user does not answer or the call times out, it is sent back to the original internal user).
  - Call is picked by another user.
  - Call is forwarded to a user's extension.
- If the contact is a callback, the following scenario is possible:
  - Callback is routed to a user.
- If the contact is an e-mail message, the following scenarios are possible:
  - E-mail message is routed to a user.
  - User requeues or forwards an e-mail message and the message is offered to another user.

- User resumes a deferred e-mail message.

- If the contact is a Web collaboration contact, the following scenarios are possible:
  - Web collaboration contact is routed to a user.
  - User receives an invitation to join a Web collaboration session.
  - User requeues a Web collaboration contact and the contact is offered to another user.

2.2 Sample scenarios

This section describes some common call scenarios using the ScreenPopTelephoneEvent.

2.2.1 Receiving a routed call

The following diagram illustrates how a ScreenPopTelephoneEvent is sent to a user's desktop when they receive a routed call.

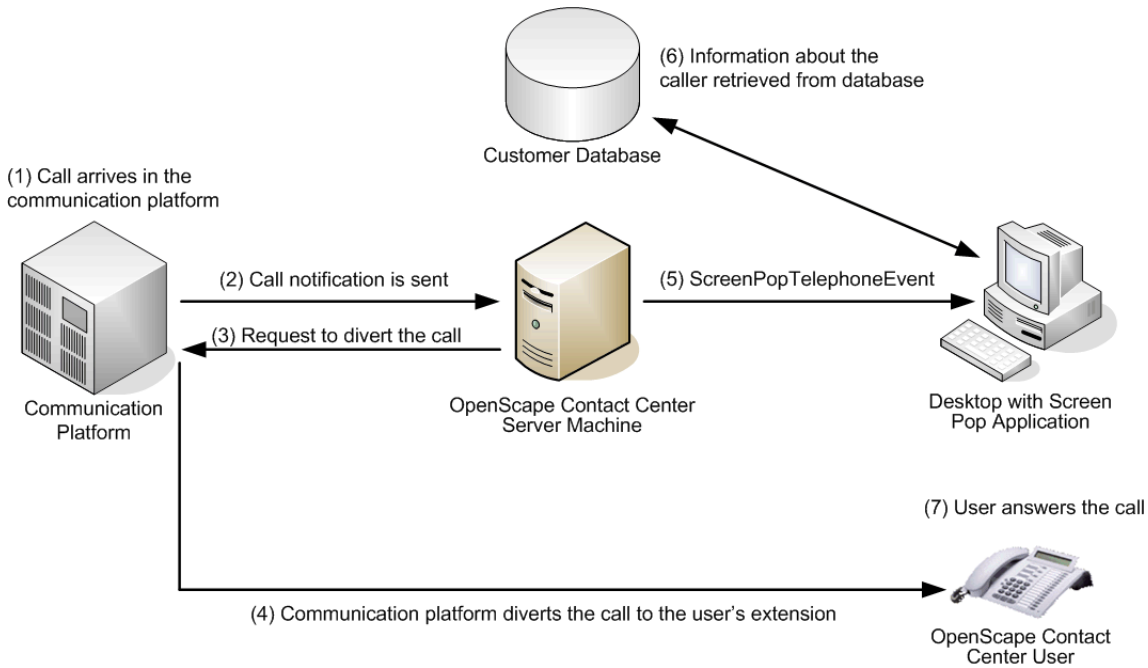


Figure 1 Receiving a routed call

1. A call arrives in the communication platform. The source of the incoming call is "12345" and the pilot number is "2468".
2. A notification of the call's arrival is sent from the communication platform to the OpenScape Contact Center system.

3. The OpenScape Contact Center system enqueues the call to the "Sales Queue" with a contact description of "Sales" and contact data of "key1" and "string1". A user is located after a wait time of 10 seconds and the system sends a request to the communication platform to divert the call to the user's extension at "13579".
4. The communication platform diverts the call to the user's extension at "13579".
5. A ScreenPopTelephoneEvent is generated and sent to the custom application running on the user's desktop.
6. The custom application searches the customer database for information gathered from the ScreenPopTelephoneEvent (optional if you decide to implement this functionality) and displays the information on the user's desktop in real time while the call is ringing at their extension.

Field Name	Field Value
Resource	13579
Source	12345
Destination	2468
Queue Name	Sales Queue
Contact Description	Sales
Contact Data	(key1, string1)
Wait Time	10
From	[empty]
Redirect	[empty]

7. The user answers the call after five seconds of ringing time.

2.2.2 Consulting on a routed call

The following diagram illustrates what occurs when a user answers the routed call from the previous example and consult with another user.

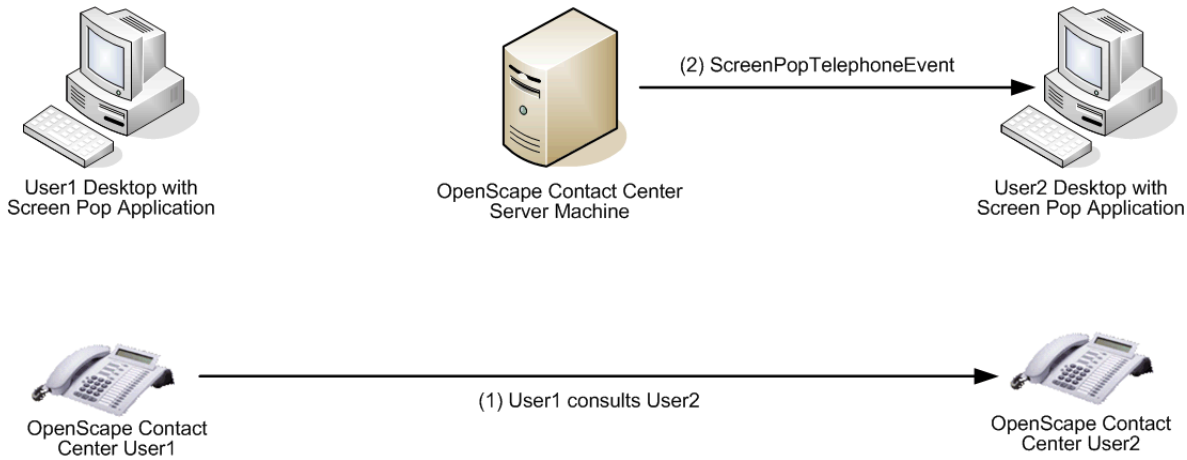


Figure 2 Consulting on a routed call

1. User1 at extension "13579" consults with User2 at extension "54321".
2. A ScreenPopTelephoneEvent is sent to User2 with the following information:

Field Name	Field Value
Resource	54321
Source	12345
Destination	2468
Queue Name	Sales Queue
Contact Description	Sales
Contact Data	(key1, string1)
Wait Time	15
From	13579
Redirect	[empty]

The Wait Time has increased to 15 seconds. This is the total amount of time that the caller spent waiting before User1 answered the call.

2.2.3 Requeuing a routed call

The following diagram illustrates what occurs when a user answers the routed call from the previous example and requeues the call.

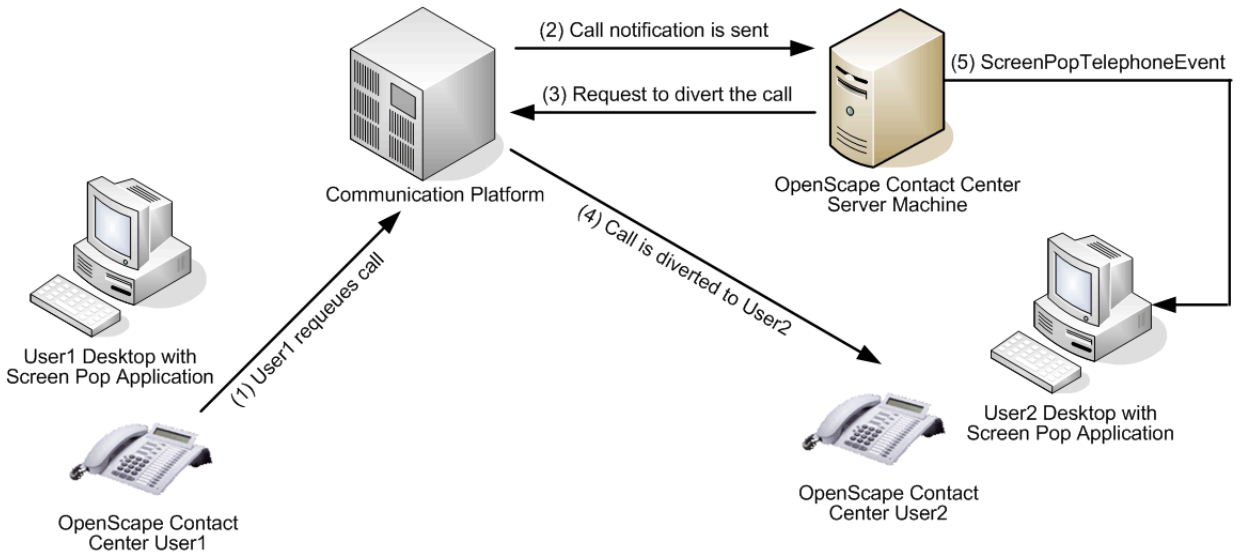


Figure 3 Requeuing a routed call

1. User1 requeues a call by dialing the pilot number "1369".
2. A notification is sent from the communication platform to the OpenScape Contact Center system.
3. The OpenScape Contact Center system enqueues the call to the "Service Queue" with a contact description of "Service" and contact data of "key2" and "string2". User2 is located after a wait time of 5 seconds and the system sends a request to the communication platform to divert the call to the user's extension at "1111".
4. The communication platform diverts the call to the User2 at "1111".
5. A ScreenPopTelephoneEvent is sent to User2 with the following information:

Field Name	Field Value
Resource	1111
Source	12345
Destination	2468
Queue Name	Service Queue
Contact Description	Service
Contact Data	(key2, string2)
Wait Time	5

Field Name	Field Value
From	[empty]
Redirect	[empty]

The original Source and Destination numbers are used. The Queue Name, Contact Description, Contact Data, and Wait Time fields are updated accordingly.

2.3 System requirements

The OpenScape Contact Center Screen Pop API can be installed on a machine that is running the following Microsoft Windows operating systems:

- Windows 10 Professional or Enterprise Edition
- Windows 8 or 8.1 Professional or Enterprise Edition or later
- Windows 7 Professional or Enterprise Edition or later
- Windows Vista Business or Enterprise Edition with Service Pack 2 or later

---

NOTE: For the Windows 7, Windows 8, and Windows 8.1 operating systems, both the 32-bit and 64-bit versions are supported. For the other Windows operating systems, only the 32-bit versions are supported.

2.4 Installing the OpenScape Contact Center Screen Pop API

This section describes how to install the OpenScape Contact Center Screen Pop API.

To install the OpenScape Contact Center Screen Pop API:

1. Log on to the computer as an Administrator. Only users with administrative rights can install OpenScape Contact Center software.
2. Insert the OpenScape Contact Center DVD into the DVD-ROM drive.
3. On the DVD, browse to the **OpenScape Contact Center Screen Pop API** folder, and then double-click **setup.exe**.

4. Select a language for the installation, then click **OK**. This launches the **OpenScape Contact Center Screen Pop API Setup Program**, which will guide you through the rest of the installation.
5. In the **Welcome** dialog box, click **Next**.
6. In the **License Agreement** dialog box, read the license agreement carefully, click **I accept the terms in the license agreement**, and then click **Next**.
7. In the **Customer Information** dialog box, type your name and the name of your organization, and then click **Next**.
8. In the **Ready to Install** dialog box, click **Install t**o begin the installation.
9. When the installation process is complete, click **Finish**.

3 Using the OpenScape Contact Center Screen Pop API

This chapter describes how to use the OpenScape Contact Center Screen Pop Application Programming Interface (API) to display information about a contact in a screen pop application on a user's desktop.

The chapter also describes enumerations, the state model, the location of the diagnostic files, and an example of how to configure the objects in the API.

3.1 OpenScape Contact Center Screen Pop API objects

This section describes how to use the objects in the hppcspa.dll to write a custom application that displays information about a contact in a screen pop application.

3.1.1 ContactDataCollection

The ContactDataCollection object stores contact data information from the respective media server's screen pop event as a collection of ContactDataItem objects. The ContactDataCollection object can be iterated by the "for each" operator in Visual Basic, or equivalent operations in other languages, to return the contact data object as well as the number of contact data objects in the collection.

3.1.2 ContactDataItem

The ContactDataItem object consists of key and value read-only properties.

3.1.3 ScreenPopCallbackEvent

The ScreenPopCallbackEvent object is sent by a ScreenPopCallbackListener object every time a callback screen pop event occurs for the user with the user ID specified in the StartListening method.

The ScreenPopCallbackEvent object collects data from the corresponding Callback Server event with the following read-only properties.

Property	Format	Description
ContactData	object	The key and value pairs attached to the callback. The ContactData property returns a ContactDataCollection object.
ContactDescription	string	The description of the callback.
Destination	string	The telephone number of the customer.
QueueName	string	The queue associated with the callback.
Resource	string	The user ID specified in the StartListening method of the ScreenPopCallbackListener.
WaitTime	integer	The amount of time that the callback has been waiting.

Table 1 ScreenPopCallbackEvent properties

3.1.4 ScreenPopCallbackListener

The ScreenPopCallbackListener object is used to listen for callback screen pop events using the StartListening method.

The ScreenPopCallbackListener object consists of the following methods:

- Initialize
- StartListening
- StopListening
- ShutDown

The ScreenPopCallbackListener object consists of the following properties:

Property	Format	Description
AdministrationServerAddress	string	Returns the address of the Administration Server from the most recent successful Initialize() call
ReconnectingInterval	string	Sets the automatic reconnection interval when the state of the ScreenPopListener object is Status_Reconnecting. The default value is 30 seconds.

Property	Format	Description
Status	value	Returns the current status of the ScreenPopListener object as a value from the enListenerStatus enumeration.
UserID	string	Returns the user ID from the most recent successful StartListening method call.

Table 2 ScreenPopCallbackListener properties

The ScreenPopCallbackListener object supports the following events:

- ScreenPopCallbackEvent — Sent to the user every time a callback screen pop event occurs for the monitored user ID.
- StatusChangedEvent — Sent to the user every time the Listener status changes, and includes the appropriate enumeration value. For more information, see Section 3.2, "Enumerations", on page 34.

3.1.4.1 Initialize method

The Initialize method initializes the ScreenPopCallbackListener object and connects to the system to retrieve configuration information. This is a blocking method that must receive either a connection result or an error from the system before continuing.

Input parameters

[in] BSTR *<AdminServerAddress>*

where *<AdminServerAddress>* is the address of the Administration Server on an OpenScape Contact Center main server machine.

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, "Error enumeration", on page 35.

Code	Description
S_OK	Method was successful.
SCREENPOP_Cannot_Create_Diag	Cannot create diagnostic file.
SCREENPOP_Cannot_Find_HiPath_ProCenter	Cannot connect to the Administration Server.

Table 3 Initialize method return codes

Code	Description
SCREENPOP_Initialization_Error	Cannot initialize the ScreenPopCallbackListener object.
SCREENPOP_Memory_Allocation_Error	Low system resources.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 3 Initialize method return codes

3.1.4.2 ShutDown method

The ShutDown method disconnects the ScreenPopCallbackListener object from the system. You can restart the instance using the Initialize method.

Input parameters

None

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, "Error enumeration", on page 35.

Code	Description
S_OK	Method was successful.
SCREENPOP_Cannot_Register	Cannot register the event. The Callback Server might not be operational.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 4 ShutDown method return codes

3.1.4.3 StartListening method

The StartListening method requests that the ScreenPopCallbackListener object begin receiving callback screen pop events from the system. This is a blocking method that must receive either a connection result or an error from the system before continuing.

Input parameters

[in] BSTR <*UserID*>

where <*UserID*> is the ID of the user that you want to monitor.

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, "Error enumeration", on page 35.

Code	Description
S_OK	Method was successful.
SCREENPOP_Cannot_Register	Cannot register the event. The Callback Server might not be operational.
SCREENPOP_Framework_Error	An internal error occurred.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 5 StartListening method return codes

3.1.4.4 StopListening method

The StopListening method requests that the system stop sending callback screen pop events to the current instance of the ScreenPopCallbackListener object.

Input parameters

None

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, "Error enumeration", on page 35.

Code	Description
S_OK	Method was successful.
SCREENPOP_Framework_Error	An internal error occurred.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 6 StopListening method return codes

3.1.5 ScreenPopEmailEvent

The ScreenPopEmailEvent object is sent by a ScreenPopEmailListener object every time an e-mail screen pop event occurs for the user with the user ID specified in the StartListening method.

The ScreenPopEmailEvent object collects data from the corresponding E-mail Server event with the following read-only properties.

Property	Format	Description
ContactData	object	The key and value pairs attached to the e-mail message. The ContactData property returns a ContactDataCollection object.
ContactDescription	string	The description of the e-mail message.
Destination	string	The e-mail address where the e-mail message was sent.
QueueName	string	The queue associated with the e-mail message.

Table 7 ScreenPopEmailEvent properties

Property	Format	Description
Resource	string	The user ID specified in the StartListening method of the ScreenPopEmailListener.
Source	string	The e-mail address of the customer.
WaitTime	integer	The amount of time that the e-mail message has been waiting.

Table 7 ScreenPopEmailEvent properties

3.1.6 ScreenPopEmailListener

The ScreenPopEmailListener object is used to listen for e-mail screen pop events using the StartListening method.

This ScreenPopEmailListener object consists of the following methods:

- Initialize
- StartListening
- StopListening
- ShutDown

The ScreenPopEmailListener object consists of the following properties:

Property	Format	Description
AdministrationServerAddress	string	Returns the address of the Administration Server from the most recent successful Initialize() call
ReconnectingInterval	string	Sets the automatic reconnection interval when the state of the ScreenPopListener object is Status_Reconnecting. The default value is 30 seconds.
Status	value	Returns the current status of the ScreenPopListener object as a value from the enListenerStatus enumeration.
UserID	string	Returns the user ID from the most recent successful StartListening method call.

Table 8 ScreenPopEmailListener properties

The ScreenPopEmailListener object supports the following events:

- ScreenPopEmailEvent — Sent to the user every time an e-mail screen pop event occurs for the monitored user ID.

- StatusChangedEvent — Sent to the user every time the Listener status changes, and includes the appropriate enumeration value. For more information, see Section 3.2, "Enumerations", on page 34.

3.1.6.1 Initialize method

The Initialize method initializes the ScreenPopEmailListener object and connects to the system to retrieve configuration information. This is a blocking method that must receive either a connection result or an error from the system before continuing.

Input parameters

[in] BSTR *<AdminServerAddress>*

where *<AdminServerAddress>* is the address of the Administration Server on an OpenScape Contact Center main server machine.

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, "Error enumeration", on page 35.

Code	Description
S_OK	Method was successful.
SCREENPOP_Cannot_Create_Diag	Cannot create diagnostic file.
SCREENPOP_Cannot_Find_HiPath_ProCenter	Cannot connect to the Administration Server.
SCREENPOP_ Initialization_Error	Cannot initialize the ScreenPopEmailListener object.
SCREENPOP_Memory_Allocation_Error	Low system resources.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 9 Initialize method return codes

3.1.6.2 ShutDown method

The ShutDown method disconnects the ScreenPopEmailListener object from the system. You can restart the instance using the Initialize method.

Input parameters

None

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, "Error enumeration", on page 35.

Code	Description
S_OK	Method was successful.
SCREENPOP_Cannot_Register	Cannot register the event. The E-mail Server might not be operational.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 10 ShutDown method return codes

3.1.6.3 StartListening method

The StartListening method requests that the ScreenPopEmailListener object begin receiving e-mail screen pop events from the system. This is a blocking method that must receive either a connection result or an error from the system before continuing.

Input parameters

[in] BSTR *<UserID>*

where *<UserID>* is the ID of the user that you want to monitor.

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, "Error enumeration", on page 35.

Code	Description
S_OK	Method was successful.
SCREENPOP_Cannot_Register	Cannot register the event. The E-mail Server might not be operational.

Table 11 StartListening method return codes

Code	Description
SCREENPOP_Framework_Error	An internal error occurred.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 11 StartListening method return codes

3.1.6.4 StopListening method

The StopListening method requests that the system stop sending e-mail screen pop events to the current instance of the ScreenPopEmailListener object.

Input parameters

None

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, "Error enumeration", on page 35.

Code	Description
S_OK	Method was successful.
SCREENPOP_Framework_Error	An internal error occurred.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 12 StopListening method return codes

3.1.7 ScreenPopTelephoneEvent

The ScreenPopTelephoneEvent object is sent by a ScreenPopTelephoneListener object every time a voice screen pop event occurs on the extension controlled by the user.

The ScreenPopTelephoneEvent collects data from the corresponding T-Server event with the following read-only properties.

Property	Format	Description
ContactData	object	The key and value pairs attached to the call. The ContactData property returns a ContactDataCollection object.
ContactDescription	string	The description of the call.
Destination	string	The telephone number that the customer dialed.
From	string	The telephone number of the user who transferred the call or initiated the consultation. This field is populated only when a transfer or consultation occurs.
QueueName	string	The queue associated with the call.
Redirect	string	The telephone number from which the call was last forwarded or deflected. This field is populated only when a redirect occurs and the system is connected to an OpenScape Voice, OpenScape 4000, or HiPath 4000 communication platform.
Resource	string	The device ID of the extension for which the event is being sent.
Source	string	The telephone number of the customer.
WaitTime	integer	The amount of time that the call has been waiting.

Table 13 ScreenPopTelephoneEvent properties

3.1.8 ScreenPopTelephoneListener

The ScreenPopTelephoneListener object is used to listen for voice screen pop events using the StartListening method. This object can be used to monitor only one extension at a time.

This ScreenPopTelephoneListener object consists of the following methods:

- Initialize
- StartListening
- StopListening
- ShutDown

The ScreenPopTelephoneListener object consists of the following properties:

Property	Format	Description
AdministrationServerAddress	string	Returns the address of the Administration Server from the most recent successful Initialize() call
Extension	string	Returns the extension of the most recent successful StartListening method call.
ReconnectingInterval	string	Sets the automatic reconnection interval when the state of the ScreenPopListener object is Status_Reconnecting. The default value is 30 seconds.
Status	value	Returns the current status of the ScreenPopListener object as a value from the enListenerStatus enumeration.

Table 14 ScreenPopTelephoneListener properties

The ScreenPopTelephoneListener object supports the following events:

- ScreenPopTelephoneEvent — Sent to the user every time a voice screen pop event occurs for the monitored extension.
- StatusChangedEvent — Sent to the user every time the Listener status changes, and includes the appropriate enumeration value. For more information, see Section 3.2, “Enumerations”, on page 34.

3.1.8.1 Initialize method

The Initialize method initializes the ScreenPopTelephoneListener object and connects to the system to retrieve configuration information. This is a blocking method that must receive either a connection result or an error from the system before continuing.

Input parameters

[in] BSTR *<AdminServerAddress>*

where *<AdminServerAddress>* is the address of the Administration Server on an OpenScape Contact Center main server machine.

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, “Error enumeration”, on page 35.

Code	Description
S_OK	Method was successful.
SCREENPOP_Cannot_Create_Diag	Cannot create diagnostic file.
SCREENPOP_Cannot_Find_HiPath_ProCenter	Cannot connect to the Administration Server.
SCREENPOP_ Initialization_Error	Cannot initialize the ScreenPopTelephoneListener object.
SCREENPOP_Memory_Allocation_Error	Low system resources.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 15 Initialize method return codes

3.1.8.2 ShutDown method

The ShutDown method disconnects the ScreenPopTelephoneListener object from the system. You can restart the instance using the Initialize method.

Input parameters

None

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, "Error enumeration", on page 35.

Code	Description
S_OK	Method was successful.
SCREENPOP_Cannot_Register	Cannot register the event. The T-Server might not be operational.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 16 ShutDown method return codes

3.1.8.3 StartListening method

The StartListening method requests that the ScreenPopTelephoneListener object begin receiving voice screen pop events from the system. This is a blocking method that must receive either a connection result or an error from the system before continuing.

Input parameters

[in] BSTR <*extension*>

where <*extension*> is the telephone extension that you want to monitor.

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, "Error enumeration", on page 35.

Code	Description
S_OK	Method was successful.
SCREENPOP_Cannot_Register	Cannot register the event. The T-Server might not be operational.
SCREENPOP_Framework_Error	An internal error occurred.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 17 StartListening method return codes

3.1.8.4 StopListening method

The StopListening method requests that the system stop sending voice screen pop events to the current instance of the ScreenPopTelephoneListener object.

Input parameters

None

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, "Error enumeration", on page 35.

Code	Description
S_OK	Method was successful.
SCREENPOP_Framework_Error	An internal error occurred.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 18 StopListening method return codes

3.1.9 ScreenPopWebCollaborationEvent

The ScreenPopWebCollaborationEvent object is sent by a ScreenPopWebCollaborationListener object every time a Web collaboration screen pop event occurs for a user.

The ScreenPopWebCollaborationEvent object collects data from the corresponding Web Interaction Server event and keeps the following read-only properties.

Property	Format	Description
ContactData	object	The key and value pairs attached to the Web collaboration contact. The ContactData property returns a ContactDataCollection object.
ContactDescription	string	The description of the Web collaboration contact.
Destination	string	The destination of the Web collaboration contact as defined by the contact center configuration (for example, Sales or Support).
QueueName	string	The queue associated with the Web collaboration contact.
Resource	string	The user ID specified in the StartListening method of the ScreenPopWebCollaborationListener.
Source	string	The source of the Web collaboration contact as defined by the contact center configuration (for example, account number or IP address).
WaitTime	integer	The amount of time that the Web collaboration contact has been waiting.

Table 19 ScreenPopWebCollaborationEvent properties

3.1.10 ScreenPopWebCollaborationListener

The ScreenPopWebCollaborationListener object is used to listen for Web collaboration screen pop events using the StartListening method.

This ScreenPopWebCollaborationListener object consists of the following methods:

- Initialize
- StartListening
- StopListening
- ShutDown

The ScreenPopWebCollaborationListener object consists of the following properties:

Property	Format	Description
AdministrationServerAddress	string	Returns the address of the Administration Server from the most recent successful Initialize() call
ReconnectingInterval	string	Sets the automatic reconnection interval when the state of the ScreenPopListener object is Status_Reconnecting. The default value is 30 seconds.
Status	value	Returns the current status of the ScreenPopListener object as a value from the enListenerStatus enumeration.
UserID	string	Returns the user ID from the most recent successful StartListening method call.

Table 20 ScreenPopWebCollaborationListener properties

The ScreenPopWebCollaborationListener object supports the following events:

- ScreenPopWebCollaborationEvent — Sent to the user every time a Web collaboration screen pop event occurs for the monitored user.
- StatusChangedEvent — Sent to the user every time the Listener status changes, and includes the appropriate enumeration value. For more information, see Section 3.2, “Enumerations”, on page 34.

3.1.10.1 Initialize method

The Initialize method initializes the ScreenPopWebCollaborationListener object and connects to the system to retrieve configuration information. This is a blocking method that must receive either a connection result or an error from the system before continuing.

Input parameters

[in] BSTR *<AdminServerAddress>*

where *<AdminServerAddress>* is the address of the Administration Server on an OpenScape Contact Center main server machine.

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, "Error enumeration", on page 35.

Code	Description
S_OK	Method was successful.
SCREENPOP_Cannot_Create_Diag	Cannot create diagnostic file.
SCREENPOP_Cannot_Find_HiPath_ProCenter	Cannot connect to the Administration Server.
SCREENPOP_ Initialization_Error	Cannot initialize the ScreenPopWebCollaborationListener object.
SCREENPOP_Memory_Allocation_Error	Low system resources.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 21 Initialize method return codes

3.1.10.2 ShutDown method

The ShutDown method disconnects the ScreenPopWebCollaborationListener object from the system. You can restart the instance using the Initialize method.

Input parameters

None

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, "Error enumeration", on page 35.

Code	Description
S_OK	Method was successful.

Table 22 ShutDown method return codes

Code	Description
SCREENPOP_Cannot_Register	Cannot register the event. The Web Interaction Server might not be operational.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 22 ShutDown method return codes

3.1.10.3 StartListening method

The StartListening method requests that the ScreenPopWebCollaborationListener object begin receiving Web collaboration screen pop events from the system. This is a blocking method that must receive either a connection result or an error from the system before continuing.

Input parameters

[in] BSTR <*UserID*>

where <*UserID*> is the ID of the user that you want to monitor.

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, "Error enumeration", on page 35.

Code	Description
S_OK	Method was successful.
SCREENPOP_Cannot_Register	Cannot register the event. The Web Interaction Server might not be operational.
SCREENPOP_Framework_Error	An internal error occurred.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 23 StartListening method return codes

3.1.10.4 StopListening method

The StopListening method requests that the system stop sending Web collaboration screen pop events to the current instance of the ScreenPopWebCollaborationListener object.

Input parameters

None

Output parameters

None

Return codes

Return codes display values from the Error enumeration. For more information, see the diagnostics file and Section 3.2.2, "Error enumeration", on page 35.

Code	Description
S_OK	Method was successful.
SCREENPOP_Framework_Error	An internal error occurred.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 24 StopListening method return codes

3.1.11 StatusChangedEvent

The StatusChangedEvent object is sent every time the Listener status changes, and includes the appropriate enumeration value. For more information, see Section 3.2.1, "ListenerStatus enumeration", on page 35.

3.2 Enumerations

This section describes the enumerations provided by the Screen Pop API.

3.2.1 ListenerStatus enumeration

The ListenerStatus enumeration represents the various states for the Screen Pop API. For example, if the state of the respective media server's screen pop event is Idle, then Status_Idle represents the Idle state in the objects that use the ListenerStatus enumeration.

State	Description
Status_Down	The ScreenPopListener object is not initialized.
Status_Idle	ScreenPopListener object is initialized.
Status_Reconnecting	The connection to the system has been lost.
Status_Unknown	The status is unknown.
Status_Up	The user is receiving screen pop events.

Table 25 ListenerStatus enumeration states

3.2.2 Error enumeration

The Error enumeration represents the various errors for the Screen Pop API. For example, if the ScreenPopTelephoneListener object cannot connect to the system, then SCREENPOP_Cannot_Find_HiPath_ProCenter represents the error in the objects that use the Error enumeration.

Error	Description
SCREENPOP_Already_Registered	The ScreenPopListenerEvent object is already registered.
SCREENPOP_Cannot_Create_Diags	Cannot create diagnostic file.
SCREENPOP_Cannot_Find_HiPath_ProCenter	Cannot connect to the Administration Server.
SCREENPOP_Cannot_Register	Cannot register the event. The related server might not be operational.
SCREENPOP_FrameWork_Error	An internal error occurred.
SCREENPOP_Initialization_Error	Cannot initialize the ScreenPopListenerEvent object.

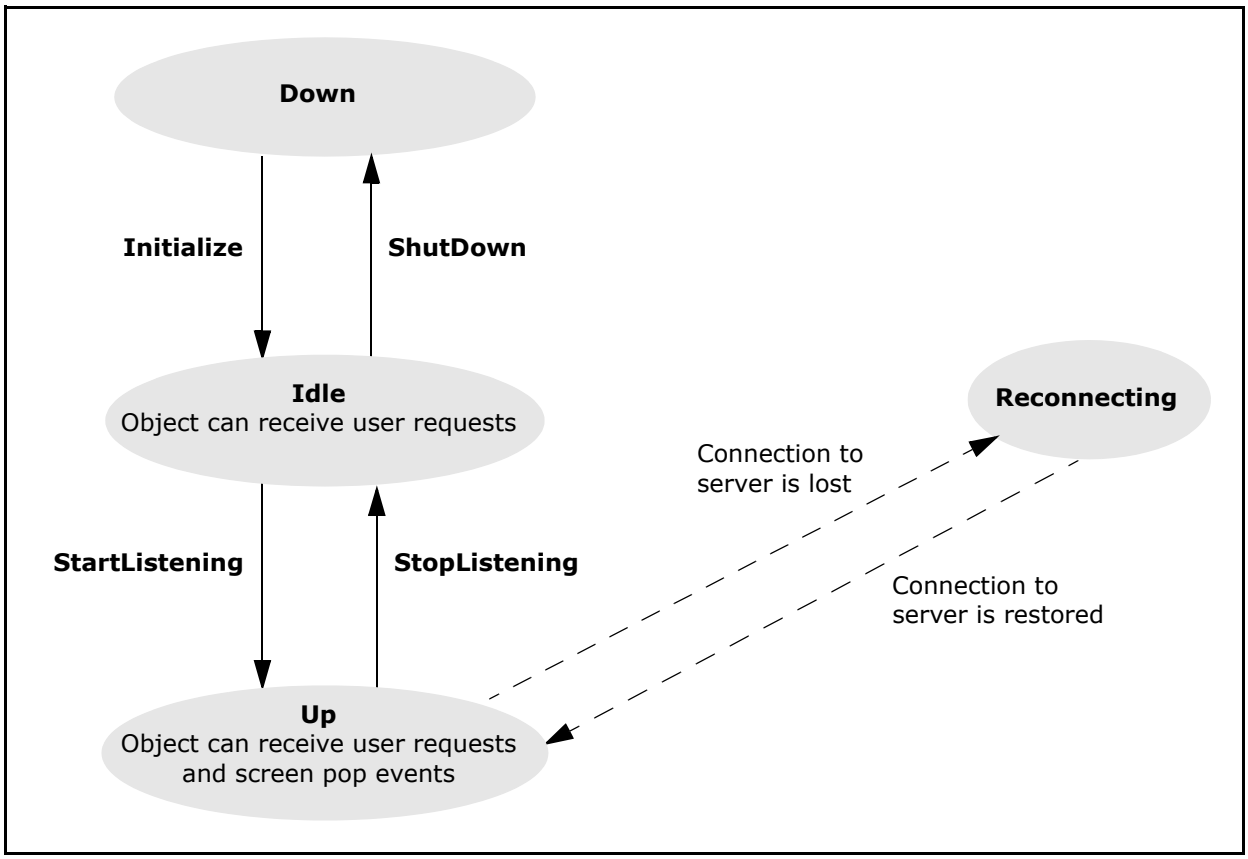
Table 26 Error enumeration errors

Error	Description
SCREENPOP_Memory_Allocation_Error	Low system resources.
SCREENPOP_Wrong_Key	Cannot retrieve the contact data because the contact data key is either invalid or missing.
SCREENPOP_Wrong_Status	Cannot perform the requested action because the ScreenPopListener object status is not correct.

Table 26 Error enumeration errors

3.3 State model

The following diagram illustrates the various states and interaction of events when using the ScreenPopListener objects. For more information on the status values shown here, see Section 3.2.1, "ListenerStatus enumeration", on page 35.



3.4 Diagnostic files

When a ScreenPopListener object is initialized, all levels of diagnostics are saved to diagnostic log files. A total of three log files can be created. When the log files reach their maximum size of 1 MB, they are renamed by incrementing the extension by one. For example, .000 becomes .001, .001 becomes .002, and a new log file is created with the .000 extension.

3.5 Sample configuration (voice)

The following example describes how to configure the objects in the Screen Pop API.

Create object

```
Dim WithEvents Listener
As HiPathProCenterScreenPopAPILibrary.ScreenPopTelephoneListener

Set Listener = New
HiPathProCenterScreenPopAPILibrary.ScreenPopTelephoneListener
```

Initialize object

```
Listener.Initialize "6000@servername"
```

Start Listening for events

```
Listener.StartListening "13579"
```

Stop Listening for events

```
Listener.StopListening
```

Shut Down

```
Listener.ShutDown
```

Destroy object

```
Set Listener = Nothing
```

Events

In between StartListening and StopListening, the screen pop event is sent by means of the event handler:

```
Public Sub Listener_ScreenPop (objScreenPopTelephoneEvent As
HiPathProCenterScreenPopAPILibrary.IScreenPopTelephoneEvent
```

In the body of the function, information from the event can be obtained:

```
objScreenPopTelephoneEvent.ContactDescription
objScreenPopTelephoneEvent.Destination
objScreenPopTelephoneEvent.From
objScreenPopTelephoneEvent.QueueName
objScreenPopTelephoneEvent.Resource
objScreenPopTelephoneEvent.Source
objScreenPopTelephoneEvent.WaitTime
objScreenPopTelephoneEvent.Redirect
```

To expand on the contact data information, you can extract the contact data:

```
HiPathProCenterScreenPopAPILibrary.ContactDataCollection
Set objContactData = objScreenPopTelephoneEvent.ContactData
Dim objContactDataItem
As HiPathProCenterScreenPopAPILibrary.ContactDataItem
For Each objContactDataItem In objContactData
objContactDataItem.key
objContactDataItem.value
Next
End Sub
```

Error handling

All screen pop errors should be handled similar to the following:

```
On Error GoTo ErrorHandler
Listener.Initialize "6000@servername"
Exit Sub
If Err.Number = SCREENPOP_Cannot_Find_HiPath_ProCenter
Then MsgBox "Cannot find OpenScape Contact Center - Check
Administration Server address"
End If
```

Index