IP Office DevLink Programmer's Guide

Release 11.1
Issue 4
February 2023

Notices

You may, at your own risk, assemble a MyDocs collection solely for your own internal business purposes, which constitutes a modification to the original published version of the publications. Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of publications. You agree to defend, indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, your modifications, additions or deletions to the publications.

A single topic or a collection of topics may come from multiple Avaya publications. All of the content in your collection is subject to the legal notices and disclaimers in the publications from which you assembled the collection. For information on licenses and license types, trademarks, and regulatory statements, see the original publications from which you copied the topics in your collection.

Except where expressly stated by Avaya otherwise, no use should be made of materials provided by Avaya on this site. All content on this site and the publications provided by Avaya including the selection, arrangement and design of the content is owned by Avaya and/or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. Avaya owns all right, title and interest to any modifications, additions or deletions to the content in the Avaya publications.

Notice

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

Documentation disclaimer

“Documentation” means information published in varying mediums which may include product information, operating instructions and performance specifications that are generally made available to users of products. Documentation does not include marketing materials. Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of Documentation unless such modifications, additions, or deletions were performed by or on the express behalf of Avaya. End User agrees to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation, to the extent made by End User.

Link disclaimer

Avaya is not responsible for the contents or reliability of any linked websites referenced within this site or Documentation provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

Warranty

Avaya provides a limited warranty on Avaya hardware and software. Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya’s standard warranty language, as well as information regarding support for this product while under warranty is available to Avaya customers and other parties through the Avaya Support website: https://support.avaya.com/helpcenter/getGenericDetails?detailId=C20091120112456651010 under the link “Warranty & Product Lifecycle” or such successor site as designated by Avaya. Please note that if You acquired the product(s) from an authorized Avaya Channel Partner outside of the United States and Canada, the warranty is provided to You by said Avaya Channel Partner and not by Avaya.

“**Hosted Service**” means an Avaya hosted service subscription that You acquire from either Avaya or an authorized Avaya Channel Partner (as applicable) and which is described further in Hosted SAS or other service description documentation regarding the applicable hosted service. If You purchase a Hosted Service subscription, the foregoing limited warranty may not apply but You may be entitled to support services in connection with the Hosted Service as described further in your service description documents for the applicable Hosted Service. Contact Avaya or Avaya Channel Partner (as applicable) for more information.

Hosted Service

THE FOLLOWING APPLIES ONLY IF YOU PURCHASE AN AVAYA HOSTED SERVICE SUBSCRIPTION FROM AVAYA OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE), THE TERMS OF USE FOR HOSTED SERVICES ARE AVAILABLE ON THE AVAYA WEBSITE, HTTPS://SUPPORT.AVAYA.COM/LICENSEINFO UNDER THE LINK “Avaya Terms of Use for Hosted Services” OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, AND ARE APPLICABLE TO ANYONE WHO ACCESSES OR USES THE HOSTED SERVICE. BY ACCESSING OR USING THE HOSTED SERVICE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE DOING SO (HEREINAFTER REFERRED TO INTERCHANGEABLY AS “YOU” AND “END USER”), AGREE TO THE TERMS OF USE. IF YOU ARE ACCEPTING THE TERMS OF USE ON BEHALF A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT THAT YOU HAVE THE AUTHORITY TO BIND SUCH ENTITY TO THESE TERMS OF USE. IF YOU DO NOT HAVE SUCH AUTHORITY, OR IF YOU DO NOT WISH TO ACCEPT THESE TERMS OF USE, YOU MUST NOT ACCESS OR USE THE HOSTED SERVICE OR AUTHORIZE ANYONE TO ACCESS OR USE THE HOSTED SERVICE.

Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, HTTPS://SUPPORT.AVAYA.COM/LICENSEINFO, UNDER THE LINK “AVAYA SOFTWARE LICENSE TERMS (Avaya Products)” OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, ARE APPLICABLE TO ANYONE WHO DOWNLOADS, USES AND/OR INSTALLS AVAYA SOFTWARE, PURCHASED FROM AVAYA INC., ANY AVAYA AFFILIATE, OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE) UNDER A COMMERCIAL AGREEMENT WITH AVAYA OR AN AVAYA CHANNEL PARTNER. UNLESS OTHERWISE AGREED TO BY AVAYA IN WRITING, AVAYA DOES NOT EXTEND THIS LICENSE IF THE SOFTWARE WAS OBTAINED FROM ANYONE OTHER THAN AVAYA, AN AVAYA AFFILIATE OR AN AVAYA CHANNEL PARTNER; AVAYA RESERVES THE RIGHT TO TAKE LEGAL ACTION AGAINST YOU AND ANYONE ELSE USING OR SELLING THE SOFTWARE WITHOUT A LICENSE. BY INSTALLING, DOWNLOADING OR USING THE SOFTWARE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE INSTALLING, DOWNLOADING OR USING THE SOFTWARE (HEREINAFTER REFERRED TO INTERCHANGEABLY AS “YOU” AND “END USER”), AGREE TO THESE TERMS AND CONDITIONS AND CREATE A BINDING CONTRACT BETWEEN YOU AND AVAYA INC. OR THE APPLICABLE AVAYA AFFILIATE (“AVAYA”).

Avaya grants You a license within the scope of the license types described below, with the exception of Heritage Nortel Software, for which the scope of the license is detailed below. Where the order documentation does not expressly identify a license type, the applicable license will be a Designated System License as set forth below in the Designated System(s) License (DS) section as applicable. The applicable number of licenses and units of capacity for which the license is granted will be one (1), unless a different number of licenses or units of capacity is specified in the documentation or other materials available to You. “Software” means computer programs in object code, provided by Avaya or an Avaya Channel Partner, whether as stand-alone products, pre-installed on hardware products, and any upgrades, updates, patches, bug fixes, or modified versions thereto. “Designated Processor” means a single stand-alone computing device. “Server” means a set of Designated Processors that hosts (physically or virtually) a software application to be accessed by multiple users. “Instance” means a single copy of the Software executing at a particular time: (i) on one physical machine; or (ii) on one deployed software virtual machine (“VM”) or similar deployment.

License type(s)

Designated System(s) License (DS). End User may install and use each copy or an Instance of the Software only: 1) on a number of Designated Processors up to the number indicated in the order; or 2) up to the number of Instances of the Software as indicated in the order, Documentation, or as authorized by Avaya in writing. Avaya may require the Designated Processor(s) to be identified in the order by type, serial number, feature key, Instance, location or other specific designation, or to be provided by End User to Avaya through electronic means established by Avaya specifically for this purpose.

Concurrent User License (CU). End User may install and use the Software on multiple Designated Processors or one or more Servers, so long as only the licensed number of Units are accessing and using the Software at any given time as indicated in the order, Documentation, or as authorized by Avaya in writing. A “Unit” means the unit on which Avaya, at its sole discretion, bases the pricing of its licenses and can be, without limitation, an agent, port or user, an e-mail or voice mail account in the name of a person or corporate function (e.g., webmaster or helpdesk), or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software. Units may be linked to a specific, identified Server or an Instance of the Software.

Cluster License (CL). End User may install and use each copy or an Instance of the Software only up to the number of Clusters as

indicated on the order, Documentation, or as authorized by Avaya in writing with a default of one (1) Cluster if not stated.

Enterprise License (EN). End User may install and use each copy or an Instance of the Software only for enterprise-wide use of an unlimited number of Instances of the Software as indicated on the order, Documentation, or as authorized by Avaya in writing.

Named User License (NU). End User may: (i) install and use each copy or Instance of the Software on a single Designated Processor or Server per authorized Named User (defined below); or (ii) install and use each copy or Instance of the Software on a Server so long as only authorized Named Users access and use the Software as indicated in the order, Documentation, or as authorized by Avaya in writing. “Named User”, means a user or device that has been expressly authorized by Avaya to access and use the Software. At Avaya’s sole discretion, a “Named User” may be, without limitation, designated by name, corporate function (e.g., webmaster or helpdesk), an e-mail or voice mail account in the name of a person or corporate function, or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software.

Shrinkwrap License (SR). End User may install and use the Software in accordance with the terms and conditions of the applicable license agreements, such as “shrinkwrap” or “clickthrough” license accompanying or applicable to the Software (“Shrinkwrap License”) as indicated in the order, Documentation, or as authorized by Avaya in writing.

Transaction License (TR). End User may use the Software up to the number of Transactions as specified during a specified time period and as indicated in the order, Documentation, or as authorized by Avaya in writing. A “Transaction” means the unit by which Avaya, at its sole discretion, bases the pricing of its licensing and can be, without limitation, measured by the usage, access, interaction (between client/server or customer/organization), or operation of the Software within a specified time period (e.g. per hour, per day, per month). Some examples of Transactions include but are not limited to each greeting played/message waiting enabled, each personalized promotion (in any channel), each callback operation, each live agent or web chat session, each call routed or redirected (in any channel). End User may not exceed the number of Transactions without Avaya’s prior consent and payment of an additional fee.

Heritage Nortel Software

“Heritage Nortel Software” means the software that was acquired by Avaya as part of its purchase of the Nortel Enterprise Solutions Business in December 2009. The Heritage Nortel Software is the software contained within the list of Heritage Nortel Products located at https://support.avaya.com/LicenseInfo under the link “Heritage Nortel Products” or such successor site as designated by Avaya. For Heritage Nortel Software, Avaya grants Customer a license to use Heritage Nortel Software provided hereunder solely to the extent of the authorized activation or authorized usage level, solely for the purpose specified in the Documentation, and solely as embedded in, for execution on, or for communication with Avaya equipment. Charges for Heritage Nortel Software may be based on extent of activation or use authorized as specified in an order or invoice.

Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation, Software, Hosted Service, or hardware provided by Avaya. All content on this site, the documentation, Hosted Service, and the product provided by Avaya including the selection, arrangement and design of the content is owned either by Avaya or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. You may not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any content, in whole or in part, including any code and software unless expressly authorized by Avaya. Unauthorized reproduction, transmission, dissemination, storage, and or use without the express written consent of Avaya can be a criminal, as well as a civil offense under the applicable law.

Virtualization

The following applies if the product is deployed on a virtual machine. Each product has its own ordering code and license types. Unless otherwise stated, each Instance of a product must be separately licensed and ordered. For example, if the end user customer or Avaya Channel Partner would like to install two Instances of the same type of products, then two products of that type must be ordered.

Third Party Components

“Third Party Components” mean certain software programs or portions thereof included in the Software or Hosted Service may contain software (including open source software) distributed under third party agreements (“Third Party Components”), which contain terms regarding the rights to use certain portions of the Software (“Third Party Terms”). As required, information regarding distributed Linux OS source code (for those products that have distributed Linux OS source code) and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply is available in the products, Documentation or on Avaya’s website at: https://support.avaya.com/Copyright or such successor site as designated by Avaya. The open source software license terms provided as Third Party Terms are consistent with the license rights granted in these Software License Terms, and may contain additional rights benefiting You, such as modification and distribution of the open source software. The Third Party Terms shall take precedence over these Software License Terms, solely with respect to the applicable Third Party Components to the extent that these Software License Terms impose greater restrictions on You than the applicable Third Party Terms.

The following applies only if the H.264 (AVC) codec is distributed with the product. THIS PRODUCT IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO (i) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD (“AVC VIDEO”) AND/OR (ii) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE HTTP://WWW.MPEGLA.COM.

Service Provider

THE FOLLOWING APPLIES TO AVAYA CHANNEL PARTNER’S HOSTING OF AVAYA PRODUCTS OR SERVICES. THE PRODUCT OR HOSTED SERVICE MAY USE THIRD PARTY COMPONENTS SUBJECT TO THIRD PARTY TERMS AND REQUIRE A SERVICE PROVIDER TO BE INDEPENDENTLY LICENSED DIRECTLY FROM THE THIRD PARTY SUPPLIER. AN AVAYA CHANNEL PARTNER’S HOSTING OF AVAYA PRODUCTS MUST BE AUTHORIZED IN WRITING BY AVAYA AND IF THOSE HOSTED PRODUCTS USE OR EMBED CERTAIN THIRD PARTY SOFTWARE, INCLUDING BUT NOT LIMITED TO MICROSOFT SOFTWARE OR CODECS, THE AVAYA CHANNEL PARTNER IS REQUIRED TO INDEPENDENTLY OBTAIN ANY APPLICABLE LICENSE AGREEMENTS, AT THE AVAYA CHANNEL PARTNER’S EXPENSE, DIRECTLY FROM THE APPLICABLE THIRD PARTY SUPPLIER.

WITH RESPECT TO CODECS, IF THE AVAYA CHANNEL PARTNER IS HOSTING ANY PRODUCTS THAT USE OR EMBED THE H.264 CODEC OR H.265 CODEC, THE AVAYA CHANNEL PARTNER ACKNOWLEDGES AND AGREES THE AVAYA CHANNEL PARTNER IS RESPONSIBLE FOR ANY AND ALL RELATED FEES AND/OR ROYALTIES. THE H.264 (AVC) CODEC IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO: (I) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD (“AVC VIDEO”) AND/OR (II) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION FOR H.264 (AVC) AND H.265 (HEVC) CODECS MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE HTTP://WWW.MPEGLA.COM.

Compliance with Laws

You acknowledge and agree that it is Your responsibility for complying with any applicable laws and regulations, including, but not

limited to laws and regulations related to call recording, data privacy, intellectual property, trade secret, fraud, and music performance rights, in the country or territory where the Avaya product is used.

Preventing Toll Fraud

“Toll Fraud” is the unauthorized use of your telecommunications system by an unauthorized party (for example, a person who is not a corporate employee, agent, subcontractor, or is not working on your company's behalf). Be aware that there can be a risk of Toll Fraud associated with your system and that, if Toll Fraud occurs, it can result in substantial additional charges for your telecommunications services.

Avaya Toll Fraud intervention

If You suspect that You are being victimized by Toll Fraud and You need technical assistance or support, call Technical Service Center Toll Fraud Intervention Hotline at +1-800-643-2353 for the United States and Canada. For additional support telephone numbers, see the Avaya Support website: https://support.avaya.com or such successor site as designated by Avaya.

Security Vulnerabilities

Information about Avaya’s security support policies can be found in the Security Policies and Support section of https://support.avaya.com/security.

Suspected Avaya product security vulnerabilities are handled per the Avaya Product Security Support Flow (https://support.avaya.com/css/P8/documents/100161515).

Trademarks

The trademarks, logos and service marks (“Marks”) displayed in this site, the Documentation, Hosted Service(s), and product(s) provided by Avaya are the registered or unregistered Marks of Avaya, its affiliates, its licensors, its suppliers, or other third parties. Users are not permitted to use such Marks without prior written consent from Avaya or such third party which may own the Mark. Nothing contained in this site, the Documentation, Hosted Service(s) and product(s) should be construed as granting, by implication, estoppel, or otherwise, any license or right in and to the Marks without the express written permission of Avaya or the applicable third party.

Avaya is a registered trademark of Avaya Inc.

All non-Avaya trademarks are the property of their respective owners.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Contents

Chapter 1: CTI DevLink

DevLink is an historical telephone API used by 3rd-party Windows applications to interface with IP Office systems. It uses a Windows DevLink DLL file to connect to the IP Office system using the system's system password. This interface is supported up to IP Office Release 9.1.

DevLink3 is a enhanced telephony API available for 3rd-party applications to interface with IP Office 10 and higher systems. It is an operating system independent interface which, unlike the previous DevLink API, no longer requires the use of a specific Windows DLL file provided by Avaya. Full documentation for DevLink3 and example files are available as an SDK from Avaya DevConnect.

This manual covers a sub-set API provided within DevLink3 to support existing pre-Release 10 DevLink applications to still run with IP Office systems upgraded to IP Office Release 10 and higher. Throughout this manual the term "legacy application" is used for such applications.

Use of both DevLink3 and legacy DevLink applications requires the IP Office system configuration to contain a CTI Link ***Pro*** license.

Changes Required to Support Legacy Applications

The follow changes to both the IP Office system configuration and the DevLink application are need for the legacy application to connect to an IP Office Release 10 or higher system.

1. Change the IP Office Security Settings – DevLink3 makes use of the enhanced security applied to all connections to IP Office systems. This requires creation of a specific service user and rights group in the IP Office security configuration.
2. Update the Application Operation – The following changes to the operation of the legacy application are required:
    a. Replace the Existing DevLink DLL – Legacy applications use a Windows specific DevLink DLL file. To support operation with IP Office Release 10, this original DLL needs to be replaced with a new DevLink3 DLL file. The DEVLink3 SDK includes separate DLL files for 32-bit and 64-bit applications.
    b. Install the Microsoft DSS Cryptographic Provider – The username/password is authenticated using SHA1 and the Microsoft DSS Cryptographic Provider. This requires the provider to be installed on the Windows PC running the application. Refer to https://msdn.microsoft.com/en-us/library/windows/desktop/aa386985(v=vs.85).aspx.
    c. Connection Password – Previously applications connected to the IP Office using the system password. The security changes above require applications to now connect using the password set for the new DevlinkDLL service user.
    d. Connection Address – Previously applications could connect to an IP Office using the broadcast address 255.255.255.255. The security enhancements require applications, including legacy applications, to connect using the specific IP address of the IP Office system.

Configuring the IP Office for DevLink

In addition to requiring a CTI Link Pro license, for IP Office Release 10 and higher, DevLink connections require changes to the IP Office security settings. The connection uses the name and password of a ***DevlinkDLL*** service user who is a member of a rights group configured for ***DevLink3***.

To configure the IP Office security settings:

1. Using IP Office Manager, select **File** > **Advanced** > **Security Settings** and load the IP Office system's security configuration.
2. Click **Rights Groups**.
    a. Create a new group called **Devlink DLL** and click **OK**.
    b. Select the **Telephony API's** tab and select **DevLink3**.
    c. Click **OK**.
3. Click **Service Users**.
    a. Create a new user called **Devlink DLL** and set the password for the new service used. Click **OK**.
    b. Check the **Account Status** is set to **Enabled**.
    c. Select **No Account Expiry**.
    d. In the list of **Rights Group Membership** select **Devlink DLL**.
    e. Click **OK**.
4. Save the changes.

Using the DevLink DLL for Legacy Applications

DevLink uses a Windows Dynamic Link Library (DLL) file called DEVLINK.DLL. This file is installed by default into "Program Files/Avaya/IP Office/DEV Link" directory. Programs using this interface simply need to link to this library and use it's functions.

- · Any application that uses the DevLink DLL should include it in the application installation script. As the DLL is not a COM component, no registration is necessary. The DLL does not require any registry settings or supporting files.
- · When developing an application that uses the DLL, the DevLink header file (***devlink.h***) and appropriate library file (***devlink.lib*** or ***devlinkomf.lib***) should be copied to the same directory as the project source files. The **devlink.lib** library file contains export symbols in COFF format which can be used with projects written in Visual C++. The ***devlinkomf.lib*** library file contains export symbols in OMF format for other linkers.
- · To support a legacy application with DevLink3, this original DLL must be replaced with the appropriate 32-bit or 64-bit DLL from the DevLink3 SDK.

The ***DEVLINK.DLL*** files can be used in any language that supports Dynamic Link Libraries (DLLs), e.g. Microsoft Visual C++ or Borland Delphi.

DevLink allows developers to request call-backs, which are generated when significant events occur. For the real-time event stream, this occurs at various stages throughout a call's lifetime. Both telephony calls and data calls are included in the events generated.

Please note that all application call-backs are made on a thread which DevLink creates. As a result, consideration must be given by programmers to ensure that all functions called from within a call-back are thread-safe.

Using DevLink with Microsoft Visual C++

Appendix A contains the ***DEVLINK.H*** file which should be used with Microsoft Visual C++. Programs written in Microsoft Visual C++ should link to the ***DEVLINK.LIB*** library.

Using DevLink with Borland Delphi

Appendix B contains the ***DEVLINK.PAS*** file which should be used with Borland Delphi. Programs written using Borland Delphi should use ***DEVLINK.PAS***, which links directly to the ***DEVLINK.DLL*** library.

IP Office connectivity with DevLink

DevLink supports connections to multiple IP Office systems at the same time. To connect to an IP Office system, the **`DLOpen()`** function must be called with the following command line:

```
LONG DLOpen( LONG pbxh, TEXT *pbx_address, TEXT *pbx_password, TEXT *reserved1, TEXT
*reserved2, CommsEvent cb);
```

Considerations

- The application-supplied ***pbxh*** handle is used internally by DevLink to track the connected IP Office System. Each connected system must have a different pbxh handle, supplied by the application.
- · The ***pbx_address*** field is the IP address of the IP Office system unit. This must be explicitly set to the IP Address of the IP Office.
- · The ***cb*** parameter (Communications Status Callback) is required, and must not be set to NULL. The return result from ***DLOpen ()*** does not necessarily indicate whether or not the connection to the system unit was successful. If the connection attempt succeeds, then a ***COMMSEVENT*** callback will be generated, indicating that connection to the system has succeeded.
- · The ***pbx_password*** parameter is the password configured for a new service user ***DevlinkDLL***. The service user is created in IP Office security settings .
- · The ***reserved1*** and ***reserved2*** parameters are for future expansion, and should be set to NULL (nil in Delphi).
- · The username/password is authenticated using SHA1 and the Microsoft DSS Cryptographic Provider is required on the Windows PC running the application.

Delphi connection example

The system password in the call to **DLOpen()** must be replaced with the actual system password of your unit.

```
unit Unit1;
interface uses Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls;
type TForm1 = class(TForm) Button1: TButton;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
private { Private declarations }
public { Public declarations }
end;
var Form1: TForm1;
implementation uses devlink;
{$R *.DFM} var hEvent : THANDLE;
dwCommsEvent : DWORD;
bStarting: boolean;
procedure HandleCommsEvent( pbxh : LongInt;
Comms_status : DWORD; Parm1 : DWORD );
stdcall;
begin case Comms_status of DEVLINK_COMMS_OPERATIONAL, DEVLINK_COMMS_NORESPONSE,
DEVLINK_COMMS_REJECTED: begin if bStarting then begin dwCommsEvent := comms_status;
SetEvent( hEvent );
end;
end;
DEVLINK_COMMS_MISSEDPACKETS: begin
// parm1 indicates the number of packets missed...
end;
end;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin bStarting := TRUE;
hEvent := CreateEvent( nil, FALSE, FALSE, nil );
DLOpen( 0, '255.255.255.255', 'systempassword', nil, nil, - HandleCommsEvent );
dwCommsEvent := DEVLINK_COMMS_NORESPONSE;
WaitForSingleObject( hEvent, 10000 );
// 10-second timeout bStarting := FALSE;
if dwCommsEvent = DEVLINK_COMMS_OPERATIONAL then begin ShowMessage('Connected OK');
end else begin ShowMessage('Error connecting to IP Office');
end;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin DLClose( 0 ); CloseHandle( hEvent );
end;
end
```

Call connection example

The system password in the call to **DLOpen()** must be replaced with the actual system password of your unit.

```
#include <windows.h> #include <stdio.h> #include "devlink.h" LONG hEvent; DWORD
dwCommsEvent; BOOL bStarting; void CALLBACK HandleCommsEvent( LONG pbxh, DWORD
comms_evt, DWORD parm1 ) { switch( comms_evt ) { case DEVLINK_COMMS_OPERATIONAL: //
we are working fine... fall through case DEVLINK_COMMS_NORESPONSE: // system not
found (initial connection), // or network connection lost (rebooted?) // fall
```

```
through... case DEVLINK_COMMS_REJECTED: // incorrect system password specified...
if( bStarting ) { dwCommsEvent = comms_evt; SetEvent( hEvent ); } else { // insert
your code here... } break; case DEVLINK_COMMS_MISSEDPACKETS: // Indicates that the
system is under // heavy load. IP Office always prioritises // data routing
and call handling above CTI events. // (parm1 contains the number of packets
missed) break; } } int main(int argc, char* argv[]) { printf( "connecting...");
bStarting = TRUE; hEvent = CreateEvent( NULL, FALSE, FALSE, NULL ); DLOpen( 0,
"255.255.255.255" "systempassword", NULL, NULL, HandleCommsEvent ); dwCommsEvent
= DEVLINK_COMMS_NORESPONSE; WaitForSingleObject( hEvent, 10000 ); // 10 seconds
bStarting = FALSE; if( dwCommsEvent == DEVLINK_COMMS_OPERATIONAL ) { printf("Connected
OK\n"); } else { printf("Error connecting to IP Office\n"); } DLClose( 0 );
CloseHandle( hEvent ); return 0; }
```

Chapter 2: Real-time event stream reference

Calls using IP Office are designed as a communications line between two endpoints, A and B respectively. A is always present, but B might not be present, depending on the state of the call. B endpoints are typically extensions or trunks, but they can also be connected to the voice mail system or parked.

The following types of real-time events are generated by DevLink and used to track the call throughout its lifetime:

- S events provide information on call status. S events are generated when calls are first created, answered, or the status of a device involved in the call changes.
- D events are generated when the call is completed or abandoned. D events indicate that the call no longer exists.
- A events are generated when one end of a call is connected to a line, such as an ISDN, QSig, or VoIP line. IP Office swaps the A and B end of the call. This event occurs when:
  - A parked party hangs up.
  - An outgoing call is transferred.
  - A call is removed from the parked state.

Applications request information real-time events by calling the **`DLRegisterType2CallDeltas()`** function after they are connected to IP Office: `LONG PASCAL DLRegisterType2CallDeltas( HANDLE pbxh, CommsEvent cb );`.

This request registers a function provided by the CommsEvent type. The function is called by DevLink when a real-time event is generated: `typedef void (CALLBACK * CommsEvent) ( LONG pbxh, TEXT * info );`

The pbxh and info parameters are provided. The pbxh parameter is the identifier used in the call to **`DLOpen()`** and can be used to identify the system when the application is connected to multiple IP Office systems. The info parameter is a string containing the call record. The call record is a comma-separated string, with variable width fields. The string is always less than 1500 bytes in length inclusive of a terminating `Null` character.

S events

S events are sent when a call is first created and any part of the call changes state. The format of an S event is: `CALL: S` <*field1*>,<*field2*>, …<*field50*>.

All the information is provided in the character string. Therefore, numbers are represented in ASCII form. All numbers are integers and do not contain a decimal point. Each string can have a maximum length of 128 characters.

Table 1: S events

Note:

Events marked as reserved might contain information, but their contents and meaning change between IP Office core software releases.

Field	Event name	Type	Description
1	A call ID	String	Call ID for the A end of the call.
2	B calld ID	String	Call ID for the B end of the call.
3	Astate	Number	State of the A end of the call. Valid numbers are: • 0 Idle • 1 Ringing • 2 Connected • 3 Disconnected • 4 Suspending • 5 Suspended • 6 Resuming • 7 Dialling • 8 Dialled • 9 Local Dial • 10 Queued • 11 Parked • 12 Held • 13 Redialling
4	Bstate	Number	State of the B end of the call. If there is no B end, this will be 0 (Idle).
5	Aconnected	Number	0 indicates not connected to the switch if receiving a dial tone.
6	A is music	Number	Indicates whether the A end of the call is listening to music on hold.
7	Bconnected	Number	0 indicates not connected to the switch if receiving a dial tone.
8	B is music	Number	Indicates whether the B end of the call is listening to music on hold.

Table continues…

Field	Event name	Type	Description
9	Aname	String	Name of the A end of the call. The format of this string is: • For extensions: Name (Number) • For trunks: Line number
10	Bname	String	Name of the B end of the call. The format of this string is: • For extensions: Name (Number) • For trunks: Line number
11	Blist	String	List of possible targets for the call. Targets are separated by semicolons, and if there are more than four, the list is terminated after 4 elements with an ellipsis.
12	Aslot Achannel	String	Aslot is the slot number of the A side of the call and is the identify of the TDM trunk. Achannel is the channel number within the Aslot. The format of the string is `Number.Number`.
13	Bslot Bchannel	String	Bslot is the slot number of the B side of the call and is the identify of the TDM trunk. Bchannel is the channel number within the Bslot. The format of the string is `Number.Number`.
14	Called party presentation and type	String	The two numbers in this field are: 1. An indicator of how the called party details should be presented. 2. The type of data to be found in the called party number field. The format of the string is `Number.Number`. For more information on presentation and type, see Called party presentation and type on page 16.
15	Called party number	String	The identifier or number of the called party.
16	Calling party presentation & type	String	The two numbers are: 1. An indicator of how the calling party details should be presented. 2. The type of data to be found in the calling party number field. The format of the string is `Number.Number`.
17	Calling party number	String	The identifier or number of the calling party.
18	Called sub address	String	The sub address of the called party.
19	Calling sub address	String	The sub address of the calling party.

Table continues…

Field	Event name	Type	Description
20	Dialled party type	Number	The type of data to be found in the dialled party number field.
21	Dialled party number	String	The identifier or number of the dialled party.
22	Keypad type	Number	The type of data to be found in the keypad number field.
23	Keypad number	String	The dialled digits.
24	Ring attempt count	Number	The number of times this call has been presented to a target.
25	Cause	Number	The reason the call is in the current state.
26	Voice mail disallow	Number	1 if the call cannot divert to voice mail.
27	Sending complete	Number	1 if overlap dialling is complete.
28	Call type and Transport type	String	The two numbers are: 1. Call type, for example, speech, video. 2. Transport type (0=circuit and 2=packet). The format of the string is `Number.Number`.
29	Owner hunt group name	String	The name of the hunt group where the call is currently queued.
30	Original hunt group name	String	The name of the hunt group where the call was originally targeted.
31	Original user name	String	The name of the user where the call was originally targeted.
32	Target hunt group name	String	The name of the hunt group where the call has been targeted.
33	Target user name	String	Name of user to which the call is being targeted.
34	Target RAS name	String	The name of the internal port where the call is being targeted.
35	IsInternalCall	Number	1 the call is internal, otherwise 0.
36	Time stamp	String	The time the call was created (internal tick count).
37	Connected time	Number	The time the call was connected.
38	Ring time	Number	The time the call started ringing.
39	Connected duration	Number	The duration the call has been connected or was connected for.
40	Ring duration	Number	The duration the call has been ringing or was ringing for.
41	Locale	String	Locale identifier (same as the locale setting in Manager).
42	ParkslotNumber	String	When the call is parked, the B end is undefined, and this field contains the park slot number.

Table continues…

Field	Event name	Type	Description
43	Call waiting	String	1 if the call is a call waiting call.
44	Tag	String	Contains the tag that may be applied through TAPI, voice mail Pro, or SoftConsole.
45	Transferring	Number	Non-zero indicates that the call is being transferred.
46	Service active	Number	Non-zero if the service is active.
47	Service quota used	Number	Non-zero if the service quota is used.
48	Service quota time	Number	The service quota time.
49	Account code	String	The account code of the call.
50	CallID	Number	Unique call ID.

Table 2: List of cause codes

Field	Cause code	Description
0	CMCauseUnknown	-
1	CMCauseUnallocatedNumber	The number is not in the dial plan.
2	CMCauseForceIdle	Force clear down unconditionally.
3	CMCauseUnregister	Indicate the operation is for unregistration procedure.
16	CMCauseNormal	-
17	CMCauseBusy	The call target is busy.
18	CMCauseNoUserResponding	-
21	CMCauseCallRejected	The call has not been permitted (call barring).
31	CMCauseNormalUnspecified	-
34	CMCauseNoChannel	There is no available external line.
38	CMCauseNetworkOOO	Problem with the external network.
88	CMCauseIncompatible	-
113	CMCausePhoneInfo	-
114	CMCauseReminderFree	The call has returned because it was parked against a busy extension that is free now.
115	CMCauseReminderNoAns	The call has returned because it was not answered within the transfer return time-out.
116	CMCauseE911Emergency	-
117	CMCauseParked	-
118	CMCauseUnParked	-
119	CMCausePickup	-

Table continues…

Field	Cause code	Description
120	CMCauseReminder	A held or parked call is returning due to a hold or park time-out.
121	CMCauseRedirect	The call has been redirected.
122	CMCauseCallBarred	The call was barred.
123	CMCauseForwardToVoicemail	The call has been forwarded to voice mail.
124	CMCauseAnsweredByOther	The call has been answered by someone else.
125	CMCauseNoAccountCode	The call has not been allowed because a valid account code has not been provided.
126	CMCauseTransfer	The call has been transferred.
127	CMCauseConferencingMove	The call has been placed into a conference.
128	CMCauseRestrictedToPartner	-
129	CMCauseHeldCall	-
130	CMRingBackCheck	-
131	CMCauseAppearanceCallSteal	The call was answered on an appearance button on another phone.
132	CMCauseAppearanceBridgeInto	-
133	CMCauseBumpedCall	The call is no longer the primary call. The call is now a waiting call.
134	CMCauseLineAppearanceCall	-
135	CMCauseUnheldCall	-
136	CMCauseReplaceCurrentCall	-
137	CMCauseGlare	-
138	CMCauseR21CompatConfMove	The call has been placed into a conference.

Called party presentation and type

The values for ETSI and additional IP Office types are listed in the following tables:

Table 3: ETSI types

Presentation	Type
• 0 = Presentation allowed. • 1 = Presentation restricted (withhold CLI). • 2 = Presentation not available (call on trunk with no CLI).	• 0 = Unknown. • 1 = International. • 2 = National. • 3 = Network Specific. • 4 = Subscriber number.

Table 4: Additional IP Office types

Presentation	Type
100 = Default	• 100 = Default. • 102 = Voice mail call. • 104 = Paging call.

D events

D events signify that the call is deleted. The format of a D event is: `CALL: D` <*field1*>,<*field2*>,… <*field3*>.

Table 5: D events

Event name	Type	Description
A call ID	String	Call id for the A end of the call.
B call ID	String	Call id for the B end of the call.
CallID	Number	Unique call ID.

A events

A events indicate that the call ends have been swapped. This event occurs when the originating extension unparks an external call. The format of an A event is: `CALL: A` <*field1*>,<*field2*>,… <*field3*>.

Table 6: A events

Event name	Type
A call ID	String

Table continues…

Event name	Type
B call ID	String
CallID	Number

Chapter 3: DevLink features

The following sections describe the key DevLink features.

Related links

[Usage of DLOpen](#) on page 19
[Usage of DLClose](#) on page 20
[Usage of DLRegisterType2CallDeltas](#) on page 20

Usage of DLOpen

You can use the **`DLOpen()`** routine to connect to an IP Office system.

Parameters

- pbxh: A number used to identify the system. This user-supplied parameter must remain consistent across all calls to DevLink.
- pbx_address: The IP address of the IP Office You can either use an IP address or a host name. You can set this address to `255.255.255.255` for DevLink to make a network broadcast to locate the system unit. Only the first unit to respond is connected. To connect to multiple system units, you must specify the IP address or host name of each unit.
- pbx_password: The password of the IP Office system.
- reserved1: This parameter should be set to `Null`, or `nil` in Delphi.
- reserved2: This parameter should be set to `Null`, or `nil` in Delphi.
- cb: The address of a callback function for sending connection status events. This parameter is mandatory.

Return value

This routine may return either `0 (DEVLINK_SUCCESS)` or `1 (DEVLINK_UNSPECIFIEDFAIL).`

Note:

A return value of `DEVLINK_SUCCESS` only indicates that communications with the unit have been initiated. The subsequent connection fails for several reasons. Further information is provided to the `CommsEvent` callback function specified in the cb parameter.

C / C++

```
LONG PASCAL DLOpen( LONG pbxh , TEXT * pbx_address , TEXT * pbx_password , TEXT *
reserved1 , TEXT * reserved2 , CommsEvent cb );
```

Delphi

```
function DLOpen(pbxh: LongInt; pbx_address: PChar; pbx_password: PChar; reserved1:
PChar; reserved2: PChar; cb: TCommsEvent): LongInt; stdcall;
```

Related links

DevLink features on page 19

Usage of DLClose

You can use the **`DLClose()`** routine to disconnect from an IP Office system.

Parameters

pbxh: A number used to identify the system. This user-supplied parameter is used to connect to DevLink in the call to **`DLOpen()`**.

Return value

This routine may return either `0 (DEVLINK_SUCCESS)` or `1 (DEVLINK_UNSPECIFIEDFAIL)` in the event of an error.

C / C++

```
LONG PASCAL DLClose( LONG pbxh );
```

Delphi

```
function DLClose(pbxh: LongInt): LongInt; stdcall;
```

Related links

DevLink features on page 19

Usage of DLRegisterType2CallDeltas

You can use the **`DLRegisterType2CallDeltas()`** routine to request Call Delta information.

Parameters

- pbxh: A number used to identify the system. This user-supplied parameter is used to connect to DevLink in the call to **`DLOpen()`**.
- cb: The address of the callback function that receives real-time events. Only one callback can receive real-time events at a time. If this parameter is `NULL`, then real-time events will no longer be sent to the application.

Return value

This routine returns:

- `0 = DEVLINK_SUCCESS.`
- `1 = DEVLINK_UNSPECIFIEDFAIL`: In the event of an error.

- `2 = DEVLINK_LICENCENOTFOUND`: If no CTI licence is activated on the IP Office system.

C / C++

```
LONG PASCAL DLRegisterType2CallDeltas( LONG pbxh, CallLogEvent cb );
```

Delphi

```
function DLRegisterType2CallDeltas(pbxh: LongInt; cb: TCallLogEvent): LongInt; stdcall;
```

Related links

DevLink features on page 19

Chapter 4: Callback events

The following sections describe the two types of callback events.

CommsEvent callback

The CommsEvent callback is called by DevLink when the state of communication with the IP Office unit changes.

Parameters

- pbxh: A number used to identify the system. The user-supplied parameter is used to connect to the IP Office system unit in **`DLOpen()`**.
- comms_state: A number indicating the state of the communications.
- parm1: This value is only defined for DEVLINK_COMMS_MISSEDPACKETS events. The value indicates the number of packets dropped.

Table 7: comms_state parameter valid values

Name	Value	Description
DEVLINK_COMMS_OPERATIONAL	0	Communication established. This response occurs either after the initial call to **`DLOpen()`**, or when the unit is available after being turned off or rebooted.
DEVLINK_COMMS_NORESPONSE	1	No response from system unit. This response occurs either after the initial call to **`DLOpen()`**, or if the system unit is turned off or rebooted. This response can also occur if network problems prevent communication.
DEVLINK_COMMS_REJECTED	2	Reserved for future use.
DEVLINK_COMMS_MISSEDPACKETS	3	Packets are generated by the IP Office unit, but are not received by DevLink. This response occurs when the IP Office unit is under heavy load, or when the application using DevLink does not return from a callback quickly. Applications must not take more than 100 milliseconds to process events.

Return value

No return value.

C / C++

```
typedef void (CALLBACK * CommsEvent)( LONG pbxh, DWORD comms_state, DWORD parm1 );
```

Delphi

```
type TCommsEvent = procedure( pbxh : LongInt; comms_state : DWORD; Parm1 : DWORD );
```

CallLogEvent callback

The CallLogEvent callback is called by DevLink to deliver a real-time (Delta2) event.

Note:

You must have a CTI license for returning Delta2 events.

Parameters

- pbxh: A number used to identify the system. The user-supplied parameter is used to connect to the IP Office system unit in **`DLOpen()`**.
- info: String containing the event.

Return value

No return value.

C / C++

```
typedef void (CALLBACK * CallLogEvent)( Long pbxh, TEXT * info );
```

Delphi

```
type TCallLogEvent = procedure( pbxh : LongInt; info : PChar );
```

Related links

Real-time event stream reference on page 11

Chapter 5: DEVLINK.PAS and DEVLINK.H

DEVLINK.PAS

A copy of the `DEVLINK.PAS` file, used for Borland Delphi programs. Programs written using Borland Delphi must use `DEVLINK.PAS`, which links directly to the `DEVLINK.DLL` library.

```
unit DEVLINK;
{************************************************************************} { Delphi
unit for DevLink (c) 2001 Avaya Global SME Solutions }
{ Contents:- } { IP Office DevLink DLL provides an interface for
managing } { the IP Office product ranges from a Windows PC }
{************************************************************************} interface
uses Windows; const DEVLINK_SUCCESS = 0; DEVLINK_UNSPECIFIEDFAIL =
1; DEVLINK_LICENCENOTFOUND = 2; const DEVLINK_COMMS_OPERATIONAL = 0;
DEVLINK_COMMS_NORESPONSE = 1; DEVLINK_COMMS_REJECTED = 2; DEVLINK_COMMS_MISSEDPACKETS
= 3; type TCallLogEvent = procedure( pbxh : LongInt; info : PChar ); stdcall;
type TCommsEvent = procedure( pbxh : LongInt; Comms_status : DWORD; Parm1 : DWORD );
stdcall; function DLOpen(pbxh: LongInt; pbx_address: PChar; pbx_password: PChar;
reserved1: PChar; reserved2: PChar; cb: TCommsEvent): LongInt; stdcall; function
DLClose(pbxh: THandle): LongInt; stdcall; function DLRegisterType2CallDeltas(pbxh:
LongInt; cb: TCallLogEvent): LongInt; stdcall; implementation function DLOpen;
external 'DEVLINK.DLL'; function DLClose; external 'DEVLINK.DLL'; function
DLRegisterType2CallDeltas; external 'DEVLINK.DLL'; end.
```

DEVLINK.H

A copy of the `DEVLINK.H` header file, used for C and C++ programs. Programs written in Microsoft Visual C++ must link to the `DEVLINK.LIB` library.

```
/*****************************************************************/ /* */ /* C/C++
Header File (c) 2001 Avaya Global SME Solutions */ /* */ /* Contents:- */ /* IP Office
Dev link DLL provides an interface for managing */ /* the IP Office product ranges from
a Windows PC. */ /*****************************************************************/
#ifndef _DEVLINK_H_ #define _DEVLINK_H_ typedef char TEXT; #define DEVLINK_SUCCESS
- 0 #define DEVLINK_UNSPECIFIEDFAIL 1 #define DEVLINK_LICENCENOTFOUND 2
#define DEVLINK_COMMS_OPERATIONAL 0 #define DEVLINK_COMMS_NORESPONSE 1 #define
DEVLINK_COMMS_REJECTED 2 #define DEVLINK_COMMS_MISSEDPACKETS 3 #ifdef __cplusplus
extern "C" { #endif typedef void (CALLBACK * CALLLOGEVENT)( LONG pbxh, TEXT * info );
typedef void (CALLBACK * COMMSEVENT)( LONG pbxh, DWORD comms_state, DWORD parm1 );
LONG PASCAL DLOpen( HANDLE pbxh , TEXT * pbx_address , TEXT * pbx_password , TEXT *
reserved1 , TEXT * reserved2 , COMMSEVENT cb ); LONG PASCAL DLClose( LONG pbxh ); LONG
PASCAL DLRegisterType2CallDeltas( LONG pbxh, CALLLOGEVENT cb ); #ifdef __cplusplus };
#endif #endif // _DEVLINK_H_
```

Chapter 6: Additional Help and Documentation

The following pages provide sources for additional help.

Related links

Additional Manuals and User Guides on page 25
Getting Help on page 25
Finding an Avaya Business Partner on page 26
Additional IP Office resources on page 26
Training on page 27

Additional Manuals and User Guides

The Avaya Documentation Center website contains user guides and manuals for Avaya products including IP Office.

- For a listing of the current IP Office manuals and user guides, look at the Avaya IP Office™ Platform Manuals and User Guides document.
- The Avaya IP Office Knowledgebase and Avaya Support websites also provide access to the IP Office technical manuals and users guides.
  - Note that where possible these sites redirect users to the version of the document hosted by the Avaya Documentation Center.

For other types of documents and other resources, visit the various Avaya websites (see Additional IP Office resources on page 26).

Related links

Additional Help and Documentation on page 25

Getting Help

Avaya sells IP Office through accredited business partners. Those business partners provide direct support to their customers and can escalate issues to Avaya when necessary.

If your IP Office system currently does not have an Avaya business partner providing support and maintenance for it, you can use the Avaya Partner Locator tool to find a business partner. See Finding an Avaya Business Partner on page 26.

Related links

Additional Help and Documentation on page 25

Finding an Avaya Business Partner

If your IP Office system currently does not have an Avaya business partner providing support and maintenance for it, you can use the Avaya Partner Locator tool to find a business partner.

Procedure

1. Using a browser, go to the Avaya Website at https://www.avaya.com
2. Select **Partners** and then **Find a Partner**.
3. Enter your location information.
4. For IP Office business partners, using the **Filter**, select **Small/Medium Business**.

Related links

Additional Help and Documentation on page 25

Additional IP Office resources

In addition to the documentation website (see Additional Manuals and User Guides on page 25), there are a range of website that provide information about Avaya products and services including IP Office.

- **Avaya Website** (https://www.avaya.com)

  This is the official Avaya website. The front page also provides access to individual Avaya websites for different regions and countries.

- **Avaya Sales & Partner Portal** *(*https://sales.avaya.com*)*

  This is the official website for all Avaya business partners. The site requires registration for a user name and password. Once accessed, the portal can be customized for specific products and information types that you wish to see and be notified about by email.

- **Avaya IP Office Knowledgebase** *(*https://ipofficekb.avaya.com*)*

  This site provides access to an online, regularly updated version of IP Office user guides and technical manual.

- **Avaya Support** *(*https://support.avaya.com*)*

This site provide access to Avaya product software, documentation and other services for Avaya product installers and maintainers.

- **Avaya Support Forums** *(*https://support.avaya.com/forums/index.php*)*

  This site provides a number of forums for discussing issues.

- **International Avaya User Group** *(*https://www.iuag.org*)*

  This is the organization for Avaya customers. It provides discussion groups and forums.

- **Avaya DevConnect** *(*https://www.devconnectprogram.com/*)*

  This site provides details on APIs and SDKs for Avaya products, including IP Office. The site also provides application notes for 3rd-party non-Avaya products that interoperate with IP Office using those APIs and SDKs.

- **Avaya Learning** *(*https://www.avaya-learning.com/*)*

  This site provides access to training courses and accreditation programs for Avaya products.

Related links

Additional Help and Documentation on page 25

Training

Avaya training and credentials are designed to ensure our Business Partners have the capabilities and skills to successfully sell, implement, and support Avaya solutions and exceed customer expectations. The following credentials are available:

- Avaya Certified Sales Specialist (APSS)
- Avaya Implementation Professional Specialist (AIPS)
- Avaya Certified Support Specialist (ACSS)

Credential maps are available on the Avaya Learning website.

Related links

Additional Help and Documentation on page 25

Index