IP Office™ Platform

Description of Devlink3 API Introduced in Release 10.0

Issue 1.2

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 by Avaya in varying mediums which may include product information, operating instructions and performance specifications that Avaya may generally make available to users of its products and Cloud Services. 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 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.

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

Cloud Service

THE FOLLOWING APPLIES IF YOU PURCHASE A CLOUD SERVICE SUBSCRIPTION FROM AVAYA OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE), THE TERMS OF USE FOR CLOUD SERVICES ARE AVAILABLE ON THE AVAYA WEBSITE,
https://www.avaya.com/en/legal/license-terms/ UNDER THE LINK “Avaya Terms of Use for Cloud Services” OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, AND ARE APPLICABLE TO ANYONE WHO ACCESSES OR USES THE CLOUD SERVICE. BY ACCESSING OR USING THE CLOUD 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 CLOUD SERVICE OR AUTHORIZE ANYONE TO ACCESS OR USE THE CLOUD SERVICE. YOUR USE OF THE CLOUD SERVICE SHALL BE LIMITED BY THE NUMBER AND TYPE OF LICENSES PURCHASED UNDER YOUR CONTRACT FOR THE CLOUD SERVICE, PROVIDED, HOWEVER, THAT FOR CERTAIN CLOUD SERVICES IF APPLICABLE, YOU MAY HAVE THE OPPORTUNITY TO USE FLEX LICENSES, WHICH WILL BE INVOICED ACCORDING TO ACTUAL USAGE ABOVE THE CONTRACT LICENSE LEVEL. CONTACT AVAYA OR AVAYA’S CHANNEL PARTNER FOR MORE INFORMATION ABOUT THE LICENSES FOR THE APPLICABLE CLOUD

SERVICE, THE AVAILABILITY OF ANY FLEX LICENSES (IF APPLICABLE), PRICING AND BILLING INFORMATION, AND OTHER IMPORTANT INFORMATION REGARDING THE CLOUD SERVICE.

Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, https://www.avaya.com/en/legal/license-terms/, 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 LLC, 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 LLC OR THE APPLICABLE AVAYA AFFILIATE ("AVAYA").

Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation, Software, Cloud Service, or hardware provided by Avaya. All content on this site, the documentation, Cloud 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. Note that 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 Cloud 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://www.avaya.com/en/legal/third-party-terms/ 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 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 VIA LICENSING ALLIANCE. SEE https://www.via-la.com/.

Service Provider

THE FOLLOWING APPLIES TO AVAYA CHANNEL PARTNER'S HOSTING OF AVAYA PRODUCTS OR SERVICES. THE PRODUCT OR CLOUD 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 G.729 CODEC, 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 G.729 CODEC IS LICENSED BY Sangoma Technologies Corporation SEE https://www.asterisk.org/products/add-ons/g729-codec/. 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 VIA LICENSING ALLIANCE. SEE https://www.via-la.com/.

Compliance with Laws

Customer acknowledges and agrees that it is responsible 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 Centre 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).

Downloading Documentation

For the most current versions of Documentation, see the Avaya Support website: https://support.avaya.com, or such successor site as designated by Avaya.

Contact Avaya Support

See the Avaya Support website: https://support.avaya.com for product or Cloud Service notices and articles, or to report a problem with your Avaya product or Cloud Service. For a list of support telephone numbers and contact addresses, go to the Avaya Support website: https://support.avaya.com (or such successor site as designated by Avaya), scroll to the bottom of the page, and select Contact Avaya Support

Contents

1. Overview

This document details a protocol that is supported since IP Office Release 10.0 which has superceded the existing Delta2 record reporting using the Devlink DLL.

In addition to the existing Delta2 events that will be streamed unchanged, additional functionality has been implemented which will be described in more detail in this document.

Since there is a plethora of different operating systems and languages in use, this protocol description will be supplied instead of targeting Windows exclusively.

2. Introduction

Third party developers would be expected to develop applications using this document as a reference.

The protocol will work in stand-alone or SCN mode.

There are a number of pre-requisites required on the IP Office for the application to successfully connect to an IP Office.

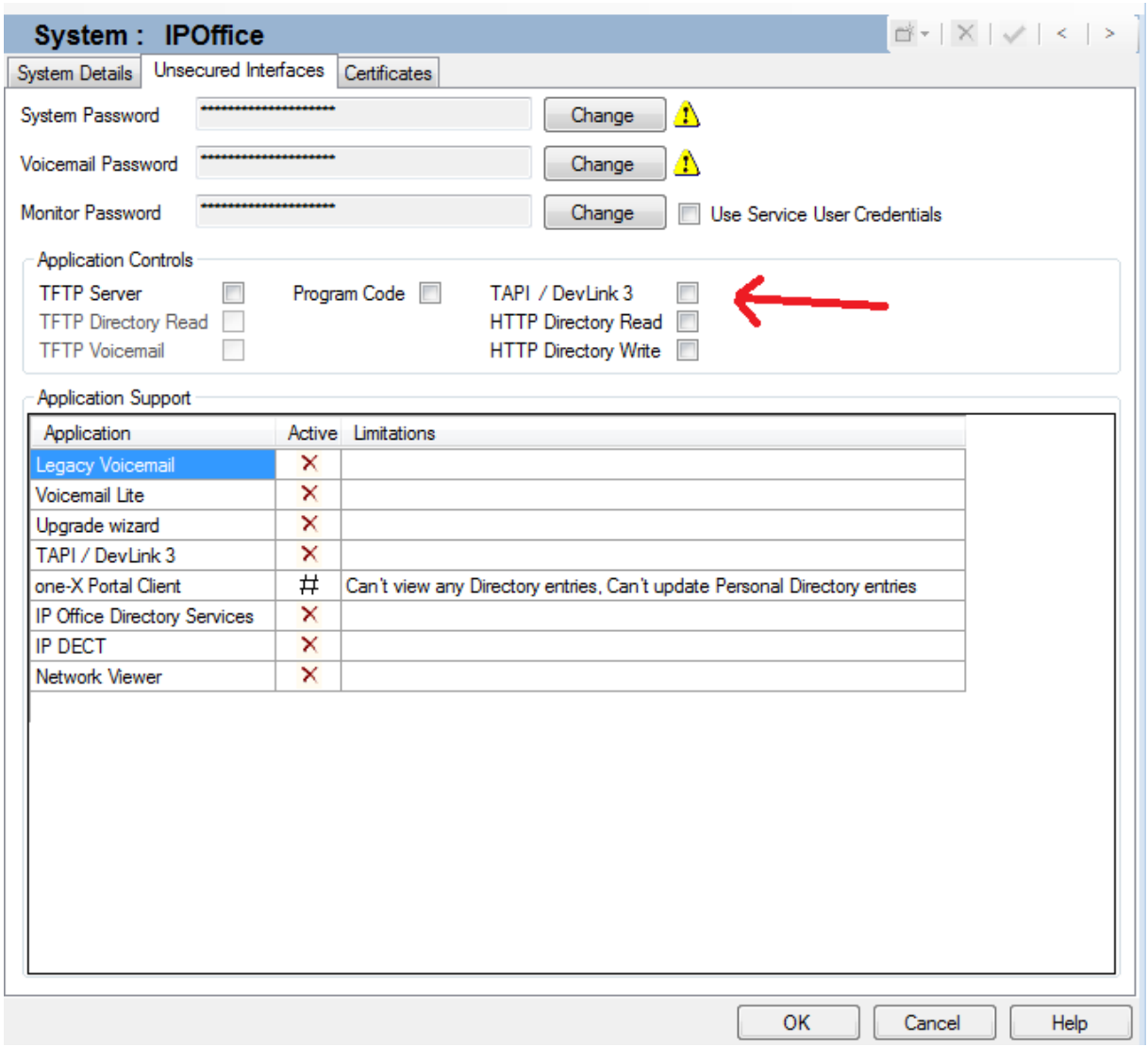
- CTI-PRO licences are required, the exact number depends on the setup. For details see Section 5.
- Configuration on the IP Office is required with a username and password which is described in this document.
- Authentication between the IP Office and the application uses SHA1 and this requires the appropriate wincrypt libraries to be installed on the device running the application. Please check the information provided here: https://msdn.microsoft.com/en-us/library/windows/desktop/aa386985(v=vs.85).aspx

3. Transport Options

This protocol is available over TCP or TLS. For TCP the port on IP Office is 50797. For TLS the port is 50796.

Availability of the TCP service is governed by the IP Office security setting: TAPI / DevLink 3 on the “Unsecured interfaces” TAB in System Security.

If you uncheck this interface, Devlink3 will only be able to connect over TLS, and it will be requested to supply a certificate (mutual certificate authentication). The certificate supplied must be in the IPOffice certificate store and be at least “medium” trust level.



Once connected, the protocol carried is packetized TCP. Each packet in either direction must be formatted as:

1 octet	0x49 - the DevLink3 discriminator
2 octets	Frame length (N+3) in network byte order
N octets	Payload

All packets must be formatted correctly. There cannot be any padding added to the stream.

After TCP/TLS is established, the DevLink3 Client must send a packet. IP Office will remain passive until it receives valid data.

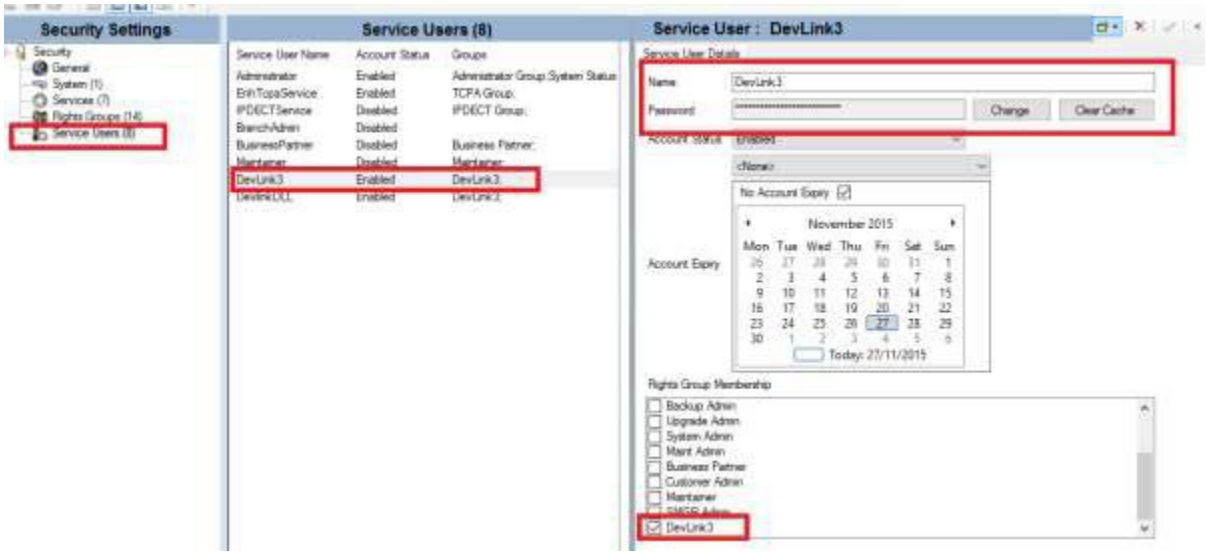
4. IP Office Configuration for Authentication

A combination of username and password is used to authenticate the application. This is configurable on the IP Office by adding a new Rights Group and Service User via the Security Settings on the IP Office.

Configure a new Rights Group and enable the checkbox DevLink3 in the Telephony API's tab.

Example below for new Rights Group DevLink3

Configure a new Service User with username and password and assign it to the DevLink3 rights group



5. Licensing

1 x CTILINK_PRO licence = No SCN option

2 x CTILINK_PRO licence = SCN option allowed on networks <= 5 nodes

3 x CTILINK_PRO licence = SCN option allowed on networks <= 20 nodes

4 x CTILINK_PRO licence = SCN option allowed on large networks (over 20 nodes)

These licences must be installed on the node into which the application is connecting.

6. Protocol Description

All 2-octet or 4-octet fields carrying numbers must be formatted in network-byte order. The following packet descriptions are all carried in the "Payload" section of the protocol.

All Unicode strings are double-byte in network byte order.

6.1. Test Packets

A test packet may be sent before authentication, in order to verify that the link is established. Test packets may be sent at any time.

4 octets	0x002A0001 – Test packet
4 octets	RequestID – to solicit a response, this must be a non-zero value.
4 octets	Length of optional payload (max 1000)
N octets	payload

The IP Office will respond with:

4 octets	0x802A0001 – Test packet response
4 octets	RequestID – this is the value from the request.
4 octets	Result: 0 = SUCCESS 0x80000001 = PAYLOAD_TOO_LONG 0x80000002 = STREAM_INTEGRITY_LOST 0x80000003 = STREAM_AUTHENTICATION_FAILED
4 octets	Length of optional payload (max 1000)
N octets	payload

If you receive a SUCCESS result, your payload will be returned to you.

If you receive PAYLOAD_TOO_LONG, you will not get payload returned to you.

If you receive STREAM_INTEGRITY_LOST, the DevLink3 stream is no longer functioning, and will have to be disconnected and restarted. You would get this if you sent an optional payload larger than the system limit (unspecified but greater than 4000), or if the payload size exceeded the frame size.

If you receive STREAM_AUTHENTICATION_FAILED, it means that an Authentication process has been previously attempted on this stream and has failed. The DevLink3 stream is no longer functioning and will have to be disconnected and restarted.

6.2. Authentication

Authentication is username/password validation. This is done through a SHA1 challenge. There is a mechanism in IPOffice to block repeated failed password attempts even if you disconnect and re-connect the link.

6.2.1. Request

4 octets	0x00300001 – Authenticate Request
4 octets	RequestID
4 octets	1 = Request
N octets	“Username” NULL terminated UTF8 string:
N octets	Table of Tuples (optional)

6.2.1.1. Request Tuples

The following tuples are supported:

4 octets	0x007E0001 – Application name
2 octets	Length of application name (including trailing 0)
N octets	NULL terminated UTF8 string: Eg “Acme BizCustomer DevLink Application 8.0”

6.2.2. Challenge

4 octets	0x80300001 – Authenticate Response
4 octets	RequestID (same RequestID as previous exchange)
4 octets	2 = Challenge (SHA1)
4 octets	Length of challenge
N octets	Random Challenge data

6.2.3. Challenge Response

4 octets	0x00300001 – Authenticate Request
4 octets	RequestID
4 octets	0x50 = Challenge Response (SHA1)
4 octets	Length of challenge response (should be 20 for SHA1)
N octets	Challenge response

The SHA1 challenge response is created by appending the 16 octets of challenge data, followed by the password encoded as plaintext (UTF8) and padded to 16 octets with 0. If the password is longer than 16 octets when converted to UTF8, only the first 16 octets are used.

So 32 octets total is passed into the SHA1 HASH, and we expect to get 20 octets back.

If using windows security you might perform the hash using code that looks like this:

```
unsigned char input[32];
memcpy( &input[0], challenge, 16 );
memset( &input[16], 0, 16 ); // pre-fill the space with 0
strncpy( &input[16], password_str, 16 ); // the password can fill
to the end
unsigned_long input_len = 32;
unsigned long output_len = MAX_DIGEST_RESULT_LEN;
HCRYPTPROV hProv = 0;
```

```
 HCRYPTHASH hHash = 0;
 if( !CryptAcquireContext( &hProv, NULL, NULL, PROV_RSA_FULL, 0 ) )
 {
   return( 0 );
 }
 if( !CryptCreateHash( hProv, CALG_SHA1, 0, 0, &hHash ) )
 {
   CryptReleaseContext( hProv, 0 );
   return( 0 );
 }
 CryptHashData( hHash, input, input_len, 0 );
 CryptGetHashParam( hHash, HP_HASHVAL, &output_data[0], &output_len,
0 );
 if( hHash )
 {
   CryptDestroyHash( hHash );
 }
 if( hProv )
 {
   CryptReleaseContext( hProv, 0 );
 }
 return( output_len );
```

output_len should be 20 at the end.

6.2.4. Authentication Result

4 octets	0x80300001 – Authenticate Response
4 octets	RequestID
4 octets	0 = Success
N octets	Table of Tuples

0 = SUCCESS

0x80000041 = AUTHENTICATION FAILED

0x80000042 = LIMITS EXCEEDED

0x80000043 = RETRY LATER

0x80000044 = LICENCE MISSING

6.2.4.1. Authentication Result Tuples

The following tuples are supported on SUCCESS:

4 octets	0x007D0001 – PBX Type and Version
2 octets	Length of version string (including trailing 0)
N octets	NULL terminated UTF8 string: Eg “Avaya IP500v2 10.0.2345.67894”

4 octets	0x007D0002 – DevLink Variant
2 octets	Length of variant string (including trailing 0)
N octets	NULL terminated UTF8 string: Eg “Standard 1.0”

6.3. Plain Challenge

Under certain conditions, the IP Office may initiate a PLAIN challenge instead of a SHA1 challenge. This will only occur over a TLS link, as it requires the client to send the password in Clear text. This packet replaces the challenge described in Challenge.

6.3.1. Plain Challenge

4 octets	0x80300001 – Authenticate Response
4 octets	RequestID (same RequestID as previous exchange)
4 octets	1 = Challenge (PLAIN)

The response should be a Plain Challenge response

6.3.2. Plain Challenge Response

4 octets	0x00300001 – Authenticate Request
4 octets	RequestID
4 octets	0x51 = Plain Challenge Response
4 octets	Length of challenge response (should be 32)
N octets	Plaintext (UTF8) password padded to 32 octets with 0.

6.4. Stream Requests

Stream requests are requests for un-solicited events from the IPOffice:

6.4.1. Devlink3 Event Request

4 octets	0x00300011 – Devlink3 Event Request
4 octets	RequestID – to solicit a response, this must be a non-zero value.
2 octets	Length of Flags section (null terminated)
N octets	Flags "-SCN -CMExtn"

The following flags are supported in the Flags:

"-SCN" stream events from every PBX on the SCN, will not work with -CD2
"-CD2" generate Delta2 events (if not present, Delta3 records are generated), prevents -SCN working

"-SIPTrack"	include Monitor SIPTrack Events
"-CMExtn"	include Monitor CMExtn events
"-CONN"	only report Delta3 events after call is initially connected
"-TEXT"	outputs CD2 as UTF8 not Unicode.

6.4.2. Devlink3 Event Response

4 octets	0x80300011 – Devlink3 Event response
4 octets	RequestID – this is the value from the request.
4 octets	Result: 0 = SUCCESS 9 = PARTIAL SUCCESS 0x80000021 = UNKNOWN FLAG 0x80000022 = INSUFFICIENT LICENCE 0x80000023 = FEATURE NOT AVAILABLE
2 octets	Length of accepted flags section
N octets	Copy of flags in the request which were accepted.

Following a successful Response, if Delta3 events are enabled, IP Office will send a Delta3 record for each eligible call currently in progress.

6.5. Stream Events

These contain the primary DevLink3 data

6.5.1. Devlink3 Event

4 octets	0x10300011 – Devlink3 Event
4 octets	Originating PBX IP Address
4 octets	Incrementing counter (per PBX)
N octets	Table of Devlink3 tuples (usually 1 entry)

6.5.1.1. Devlink3 Tuple CallDelta3

4 octets	0x00760001 – CallDelta3
2 octets	Length of CallDelta3 string (including trailing 0)
N octets	NULL terminated xml formatted UTF8 string: Description in section 5

6.5.1.2. Devlink3 Tuple CallDelta2

4 octets	0x00760001 – CallDelta3
4 octets	0x00760002 – CallDelta2
2 octets	Length of CallDelta2 string (including trailing 0x0000)
N octets	NULL terminated UNICODE string: This is the same as the old CallDelta2 record

6.5.1.3. Devlink3 Tuple SIPTrack Event

4 octets	0x00760003 – SIPTrack Event
2 octets	Length of SIPTrack Event string (including trailing 0x0000)
N octets	NULL terminated UNICODE string: Eg “START,c0a82a1f000003eb,17.1003.1,TRUNK,IN,0f5757c5c504c74197be9699697eab8d”

These SIPTrack events do not correspond to current SIP Events. They are explicitly to allow association of SIP CallID with endpoint reported in Delta2.

$event,$epid,$end_id,$siptype,$direction,$sip_callid

$event options are:

START | END

$siptype options are:

TRUNK | PHONE

$direction options are:

IN | OUT

6.5.1.4. Devlink3 Tuple CMExtn Event

4 octets	0x00760004 – CMExtn Event
2 octets	Length of CMExtn Event string (including trailing 0x0000)
N octets	NULL terminated UNICODE string: Eg "Extn4001: CALL LOST (CMCauseForceIdle)"

These events correspond to current CMExtn Events.

6.5.1.5. Devlink3 Tuple CMExtn Extended Event

Extension state changes have to be reported slightly differently as the event payload contains 3 numeric fields: v, p1, p2.

The familiar "v=1 State, new=Dialling old=Idle,0,0,Extn865001" is actually:

v=1

p1=2

p2 = 0

UNICODE string="Extn865001"

4 octets	0x00760004 – CMExtn Event
2 octets	Length of CMExtn Event string(including trailing 0x0000) + 12
4 octets	"v"
4 octets	p1
4 octets	p2
N octets	NULL terminated UNICODE string: Eg "Extn865001"

These Extension state reporting events are the only events of this type.

6.6. Commands

6.6.1. ReadFile(user_list)

You can use this command to use the Devlink3 protocol as an authenticated pbx pseudo-file reader.

A common pseudo-file that developers may want to access is the ipoffice user list.

4 octets	0x00300041 – Read file
4 octets	RequestID – to solicit a response, this must be a non-zero value.
2 octets	Length of filename (including NULL)
N octets	“nasystem/user_list”

Result…

4 octets	0x80300041 – Read file Response
4 octets	RequestID – from the ReadFile
4 octets	SUCCESS or NOT FOUND
N octets	File contents….

Note that if the frame payload is greater than 0x7fff octets, there are 3 octets of length in the frame header. This can occur on large files.

```
UBYTE header[6];
header[0] = 0x49;
if( len + 3 > 0x7fff )
{
    len += 4;
    header[1] = (UBYTE )(len >> 15) | 0x80;
    header[2] = (UBYTE )(len >> 8) & 0x7f;
    header[3] = (UBYTE )(len);
```

```
        }
        else
        {
            len += 3;
            header[1] = (UBYTE )(len >> 8);
            header[2] = (UBYTE )(len);
        }
….
```

It is not a problem to open several Devlink3 connections. The recommendation would be that for each FileRead you want to perform, you open a Devlink3 connection just for that purpose, and close it after the file is loaded.

6.7. Delta3 Protocol Records

There are 4 types of Delta3 protocol records:

Detail

This should contain detail equivalent to the content of a CALL record in a Delta2 record. It is divided into sections:

<Call>

<PartyA>

<PartyB>

<Targets>

A call always has a PartyA, and may have a PartyB or a list of Targets, or neither. These are alternatives. Targets only occur during call establishment, and once established the Targets list is replaced by PartyB.

CallLost

When IP Office reports a channel as PartyA or PartyB, it will guarantee to generate a CallLost (or LinkLost) for that Call party when it is disconnected. Note that you do not get any explicit event when a call is cleared, only you can deduce that if there are no longer any parties attached to the call, the call is gone.

LinkLost

A LinkLost is the same as a CallLost, but is generated for intermediate nodes. For calls transiting an SCN, the CallLost is generated only at the extremities of the call. The LinkLost is generated at intermediate points along the call.

AttemptReject

If a User was in the targets list but tries to push the call away by manual intervention (eg Drop key), an AttemptReject is generated.

6.7.1. Detail Record

The following records are examples of the Detail Record.

6.7.1.1. Dial Tone

(TDM phone 4001 goes offhook)

```
<Detail>
 <Call>
  c0a82a03,1,,,,0,,,,,
 </Call>
 <PartyA>

c0a82a03000003ea,1,7,1,3,0,0,0,1,8,106,,4001,0,0,,,,,,,4001,100
 </PartyA>
</Detail>
```

6.7.1.2. Incoming ISDN call

(DID=5678, CLI=01707123456, calling huntgroup 200, group ring 3 phones 4002, 4003, 4001)

(Trunk has '9' prefix, lineid=5, BChannel=23)

```
<Detail>
 <Call>
  c0a82a03,3,,,,0,,200,,200,
 </Call>
 <PartyA>
0a82a0300002329,1,19,1,2,0,1,0,0,2,2,5.23,,0,0,5678,,,,,,0170712
3456,100
 </PartyA>
 <Target_list>
  <Target>
  c0a82a03000003fe,0,0,8,106,,4002,0,4,,,200,901707123456,100
  </Target>
  <Target>
   c0a82a03000003ff,0,0,8,106,,4003,0,4,,,200,901707123456,100
  </Target>
  <Target>
   c0a82a0300000407,0,0,8,106,,4001,0,4,,,200,901707123456,100
  </Target>
 </Target_list>
</Detail>
```

6.7.1.3. Answered ISDN call

(call above was answered by 4001)

```
<Detail>
 <Call>
  c0a82a03,4,,,,0,,200,,200,
 </Call>
 <PartyA>

c0a82a030000232a,1,2,1,0,0,1,0,0,2,2,5.23,,0,0,5678,,,,,,0170712
3456,100
 </PartyA>
 <PartyB>

c0a82a0300000413,0,2,1,0,0,0,0,1,8,106,,4001,0,4,,,,,,200,901707
123456,100
 </PartyB>
</Detail>
```

6.7.1.4. Phone hung up

4001 hangs up on call above

```
<CallLost>

 c0a82a03,c0a82a0300000413,1,16

</CallLost>
```

6.7.2. Call Section

There are at least 10 fields in the Call record, comma-separated and described as

Node identifier	Utf8String 8 characters long. A unique number of the IPOffice that originated this record. While calls can span multiple nodes on an SCN, this is the view of this call as it transits this node.
CallID	Integer. This CallID only has local scope. It is the identity of this call on this PBX.
Account code	Utf8String. This is an account code assigned to this call.
Authorization code	Utf8String This is an authorization code assigned to this call. It will be masked as ***** if present.
Tag	Utf8String

	A scrap of text assigned to this call. If interworking with Aura equipment, this may contain UCID
Transfer cause	Integer (set to 0)
Owner huntgroup	Utf8String (Extn Number)
Originally targeted huntgroup	Utf8String (Extn number)
Originally targeted user	Utf8String (Extn number)
Currently targeted huntgroup	Utf8String (Extn number)
Currently targeted user	Utf8String (Extn number)
Is a voicemail message being left	Integer (local VMPro only)
Length of voicemail message left	Integer (seconds)
Is call queueing or listening to announcement	1 = Queueing 3 = Listening to announcement
Timestamp (1/10 seconds)	Integer
Huntgroup Overflow?	Integer
Transferred out of IVR Callflow	Quoted Utf8String
Time spent in IVR Callflow (1/10 seconds)	Integer

6.7.3. Party Section

(Both for PartyA and PartyB)

End identifier	Utf8String 16 characters long. This end identifier (like a SIP CallID) is unique within the SCN (notwithstanding reboots), and is the same for all views of this call on whichever PBX reports the Detail. The first 8 characters of the string identifies the IPOffice node where the end actually is local.
Direction	Integer If this is 1, the call came in on this end. So for an ISDN trunk, it is an incoming call. For a handset, it means the handset made the call.
State	Integer The primary (Q931) call state of this call end. 0 = Idle 1 = Ringing 2 = Connected 3 = Disconnecting 7 = Dialling 8 = Dialled 9 = Dial Initiated 15 = Offering 16 = Overlap receive 17 = Accept 18 = ConnectRequest 19 = Ringback 20 = OGConnect Request 21 = IC Disconnecting 22 = Seized 23 = Completed 24 = Completed Tone 25 = Preserved
Audiopath	Integer 1 = There is audio path

	0 = There is no audio path (ie the call is on Hold)
Tone	Integer 0 = There is no tone 1 = Listening to holdmusic 2 = Listening to RingBack tone 3 = Listening to dial tone 6 = Listening to busy tone
Remote	Integer 0 = This call end terminates on this node. The detail on this record is complete. 1 = This call end transits from a different node. You may not get 100% of the descriptive fields from a transit node.
Public line	Integer 0 = This call end does not terminate a public line 1 = This call end terminates a public line. (ISDN R2, T1, SIP)
Flags	Integer This is a bit-field of interesting facts about this call end: 0x00000001 = Privacy is requested by this call end 0x00000002 = This call was auto-answered 0x00000004 = This call end is being auto-recorded using VMPro. 0x00000010 = Consent question answered by this call end (release 11.1 onwards) 0x00000020 = Consent refused (release 11.1 onwards)
Emergency location	Integer The location ID of this call end, as configured in the emergency number location information.
Equipment classification	Integer What is this call end? Unknown = 1 ISDNTrunk = 2 AlogTrunk = 3 H323Trunk = 4 SIPTrunk = 5 T1Trunk = 6 R2Trunk = 7 TDMPhone = 8 H323Phone = 9 SIPDevice = 10 DECTPhone = 11 Voicemail = 12 WAVDriver = 13 Router = 14 DTEPort = 15 Parkslot = 16 Conference = 17 PagingConference = 18 RecordingHandler = 19 S0Trunk = 20 MobilePhone = 21 GhostHandler = 22 IPONotificationHandler = 23 Alarm call = 24

	VoiceScreening = 25 ConferenceRecorder = 26 Outdialer = 27 WebRTCPhone = 28 ListenOrCoach = 29 ConferenceMeetMe = 30
Equipment type	Integer Particularly for phones, this identifies the kind of equipment more precisely. Of particular interest, for a “SIPDevice”, these are the common contact center types: ACCS = 162 IPOCC = 163
Description	Utf8String This content depends on the equipment classification, and whether “remote” is set. For a local SIP trunk: “SIP CallID” For local ISDN trunk: “lineID.BChannel” For conference: “Join reason and conference role”
Identity	Utf8String Extn number of an IPOffice User using a handset (so if not a “user”, then not populated.)
Called Type	Integer The type of call being made (or received) (This list not comprehensive) 0=Unspecified 1=International 2=National 3=NetworkSpecific 4=Subscriber 100=Unspecified 101=Internal 102=Voicemail 103=ACD 104=Paging 105=Direct 106=Intrude 107=Priority 108=Pickup 109=CampOn 110=Steal 111=Whisper 112=Inclusione 113=Coverage 114=EConf 115=EConfDial 116=Conf 117=EConfInvite 118=Restricted 119=Emergency 120=FNE 121=Identifier 122=Receiver 123=Reminder 124=Listen 125=RemoteDial 126=FaxCall

	127=WakeupCall 129=LocalPhysExtn 130=XferToTwinMobile 131=Coach 132=WhisperPage 133=CallLog 134=ParkCall 135=UnParkCall 136=Precision 137=RequestSupervising 138=RParkCall 139=ACCSGroup 140=BlendingAcquireCall
Called Reason	Integer For a call with direction = 0: Reason for the call arriving here.
DID	Utf8String For call received off a trunk (direction=1) Called number received from ISDN, or To/ReqURI number received off SIP.
Dialled	Utf8String For a call made by a local phone or other equipment, this is the sequence of digits actually pressed (unless masked).
OnBehalf	Uft8String For outbound calls to eg a trunk, this will be populated if the call was made using a Bridge Appearance or an Authorization code to modify the apparent identity of the person making the call.
Target number	Utf8String For outbound calls (direction=0) this is the number sent to the trunk or other equipment
Target subaddr	Utf8String For outbound calls (direction=0) this is the subaddress sent to the trunk or other equipment
Purpose	Utf8String Nominal target for this call – may be a huntgroup, skill or diversion.
CLI	Utf8String Calling party number. On a local trunk device, this is the true CLI that was either sent or received.
CLI Presentation	Integer If the CLI is present/not present, there may be a reason: 0 = Allowed 1 = Withheld 2 = Not available 100 = Unspecified
Calling party name type	0 = default 1 = directory match 2 = explicitly set 4 = restricted 5 = internal user 6 = huntgroup 8 = conference 9 = external trunk 12 = voicemail callflow Additional enumerations may apply
Calling party name	Quoted Utf8String
Related End identifier	Utf8String

(during certain types of transfer)	16 characters long, or empty.

6.7.4. Target Section

For each target in the target section, there is a reduced equivalent of the PartyInfo record. Where the fields have the same name, they have the same meaning.

Of course all targets have direction=0

End identifier	Utf8String
Remote	Integer
Public line	Integer
Equipment classification	Integer
Equipment type	Integer
Description	Utf8String
Identity	Utf8String
Called Type	Integer
Called Reason	Integer
Target number	Utf8String
Target subaddr	Utf8String
Purpose	Utf8String
CLI	Utf8String
CLI Presentation	Integer

6.7.5. Call Lost

For each node that has reported a call end as either PartyA or PartyB, there will be a local "CallLost" record generated

The detail generated on the local node will be most exact.

Node identifier	Utf8String

	8 characters Unique identifier of the node generating this. CallLost
End identifier	Utf8String 16 characters The unique identity of this call end
Cleared Inbound	Integer 1 = this call was cleared from this device 0 = this call was cleared for another reason than this (phone hung up/trunk received Bye) endpoint positive activity.
Cause	Integer Clearing cause including: 16 = Normal 126 = Transfer
Timestamp (1/10 seconds)	Integer

6.7.6. Link Lost

For each intermediate node on an SCN that has reported a call end as either PartyA or PartyB, there will be a local "LinkLost" record generated

Node identifier	Utf8String 8 characters Unique identifier of the node generating this. CallLost
End identifier	Utf8String 16 characters The unique identity of this call end
Cleared Inbound	Integer 1 = this call was cleared from this device 0 = this call was cleared for another reason than this (phone hung up/trunk received Bye) endpoint positive activity.
Cause	Integer Clearing cause including: 16 = Normal 126 = Transfer
Timestamp (1/10 seconds)	Integer
Link local reference	Utf8String Matches "description" field

6.7.7. AttemptReject

For a User who is in the Target list and tries to send away the ringing call by manual intervention, an AttemptReject event is generated. This may or may not stop the User from ringing.

Node identifier	Utf8String 8 characters Unique identifier of the node generating this. CallLost
End identifier	Utf8String 16 characters The unique identity of this call end
User	Utf8String User trying to deflect the call
Qualifier	Utf8String May contain a description of the method being used (eg “dnd”)
Timestamp (1/10 seconds)	Integer

7. Limits

When monitoring events, a maximum of 3xDevLink3 connections are allowed to a single IPOffice.

If using multi-node flags, only 3xDevlink3 connections are allowed on a single SCN.

8. Version Compatibility

This is supported on IPOffice 10.0 and newer versions until such time as it is withdrawn.

9. Change History

Date	
30/7/2016	Initial Creation
03/05/2017	Updated for Release 10.1 to include extra content. Changes in Section 6.
30/05/2018	Up issued to version 1.0 for Release 11.0. No changes to DevLink3 interface or capabilities with Release 11.0. Clean up change tracking in Section 6.
31/3/2020	Added Consent flags in section 6.7.3. Additional bits in 'flags' integer for Release 11.1.
30/7/2025	Updated document for Release 12.2.