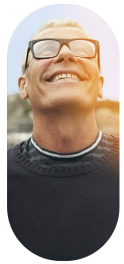
Getting/Setting SIP Headers



Getting and Setting SIP Headers

Use Case / Challenge

- Use of custom SIP headers increasingly requested as a mechanism for passing data with calls to/from CVP
- Pass user-data/context from third-party SIP PBX/ACD to CVP
- Forward user-data with transfers from CVP to third-party platforms
- Access additional signaling information such as Remote-Party-ID, physical trunk information, privacy settings

```
INVITE sip:90179017@10.52.200.50:5060 SIP/2.0
Via: SIP/2.0/UDP 10.58.16.170:5060;x-ds0num="Basic Rate Interface 0/1/0 1:DS0";branch=z9hG4bK45751B21
Remote-Party-ID: <sip:396298@10.58.16.170>;party=calling;screen=yes;privacy=off
```

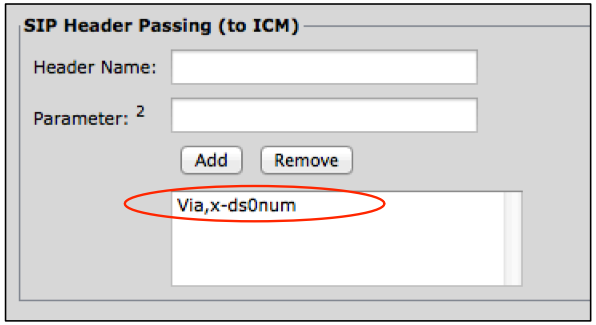
Problem: CVP Comprehensive Model allows SIP header content to be retrieved and added/modified but how is it done in CVP Standalone?

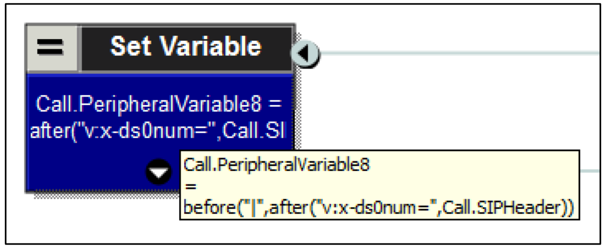
Getting SIP Headers

ICM Script / CVP Comprehensive Model

Documented in the CVP Configuration and Administration Guide

Configure CVP Call Server SIP settings via OAMP console to specify which headers and parameters should be extracted and passed up to ICM





Call.SIPHeader contains the headers configured to be passed to ICM. In this example, it contains:

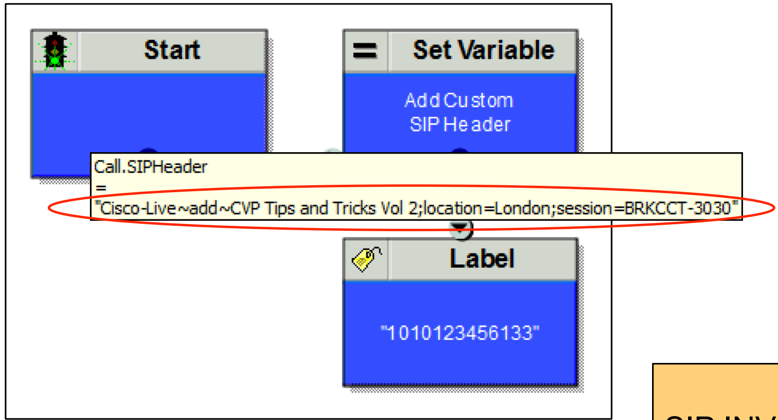
v:x-ds0num="Basic Rate Interface 0/1/0 1:DS0"

Call.SIPHeader contents are parsed to extract the right-hand-side into a call variable.

(Vertical bar "|" character is the separator if multiple items extracted)

Setting SIP Headers

ICM Script / CVP Comprehensive Model



Set variable Call.SIPHeader with the required header modifications

- Add, Modify, Remove operations
- Vertical bar used as separator between multiple headers

Example here adds Cisco-Live custom header before CVP transfers call

SIP INVITE on CVP transferred call leg showing the Cisco-Live custom header added by the ICM script

```
INVITE sip:1010123456133@10.58.16.170;transport=tcp SIP/2.0
Via: SIP/2.0/TCP 10.58.16.180:5060;branch=z9hG4bKV0kC39vJ3IT8H+sGkW2wcg~~809
To: <sip:1010123456133@10.58.16.170;transport=tcp>
From: 396298 <sip:396298@10.58.16.180:5060>;tag=ds1f6be9
Call-ID: 76918BCF560111E2811E001B0CFAA768-1357343908541127@10.58.16.180
User-Agent: CVP 9.0 (1) Build-670
Call-Info: <sip:10.58.16.170:5060>;purpose=x-cisco-origIP
Date: Fri, 04 Jan 2013 23:58:25 GMT
Cisco-Live: CVP Tips and Tricks Vol 2;location=London;session=BRKCCT-3030
```

The custom header has been inserted

How can a CVP Call Studio application read it?

Getting SIP Headers

CVP Standalone Model

- SIP headers can be retrieved using the Cisco VoiceXML session variable session.com.cisco.proto_headers
- Invoke simple external VoiceXML using a Subdialog Invoke element
- Return the header(s) from the subdialog
  – but may still need more parsing if header has multiple parameters and …
  – need to avoid problems with “=“ characters in the header parameters

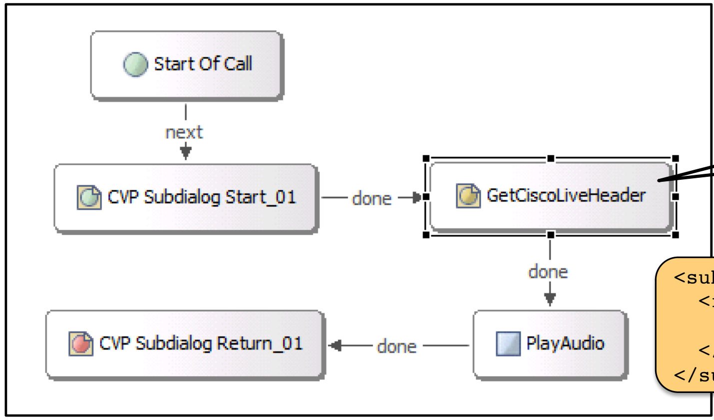
getCiscoLiveHeader.vxml

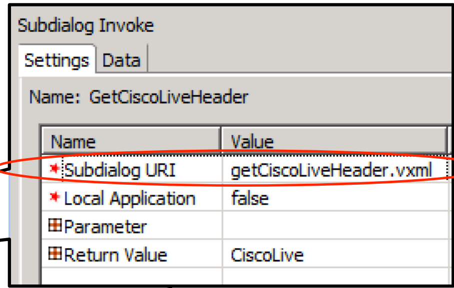
```
<?xml version="1.0"?>
<vxml version="2.0”>

  <form id="getheaders”>
    <var name="CiscoLive" expr="session.com.cisco.proto_headers['Cisco-Live'].replace(new RegExp('=', 'g' ),'::')"/>
    <block>
      <return namelist="CiscoLive"/>
    </block>
  </form>
</vxml>
```

Get Cisco-Live SIP header and return it as CiscoLive parameter from the subdialog

CVP Call Studio Application

Using Subdialog Invoke






```
<subdialog name="subdialog" src="getCiscoLiveHeader.vxml">
  <filled>
    <submit next="/CVP/Server" method="post" namelist="subdialog.CiscoLive audium_vxmlLog subdialog"/>
  </filled>
</subdialog>
```

```
INVITE sip:1010123456133@10.58.16.170;transport=tcp SIP/2.0
Via: SIP/2.0/TCP 10.58.16.180:5060;branch=z9hG4bKV0kC39vJ3IT8H+sGkW2wcg~~809
To: <sip:1010123456133@10.58.16.170;transport=tcp>
From: 396298 <sip:396298@10.58.16.180:5060>;tag=ds1f6be9
Call-ID: 76918BCF560111E2811E001B0CFAA768-1357343908541127@10.58.16.180
User-Agent: CVP 9.0 (1) Build-670
Call-Info: <sip:10.58.16.170:5060>;purpose=x-cisco-origIP
Date: Fri, 04 Jan 2013 23:58:25 GMT
Cisco-Live: CVP Tips and Tricks Vol 2;location=London;session=BRKCCT-303
```

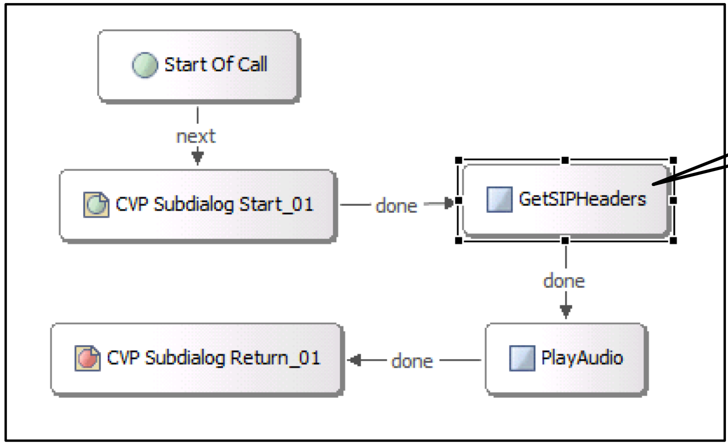
Good candidate for a custom element:

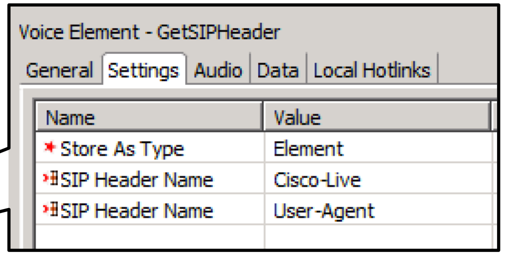
- Add settings flexibility
- Parse results into element/session data
- Avoid using external VoiceXML files

```
CL13_SIPHeader,01/06/2013 02:39:15.580,CVP Subdialog Start_01,exit,done
CL13_SIPHeader,01/06/2013 02:39:15.580,GetCiscoLiveHeader,enter,
CL13_SIPHeader,01/06/2013 02:39:15.658,GetCiscoLiveHeader,data,CiscoLive,CVP Tips and Tricks Vol 2;location::London;session::BRKCCT-3030
CL13_SIPHeader,01/06/2013 02:39:15.658,GetCiscoLiveHeader,exit,done
CL13_SIPHeader,01/06/2013 02:39:15.658,PlayAudio,enter,
```

CVP Call Studio Application

Using Custom Element





Settings flexibility

- Specify whether header info written to element or session data
- Allow any number of headers to be retrieved

More elegant results parsing and storage

- Separate data items for each parameter
- Create element data with naming “headername.parameter”

Activity Log

```
CL13_SIPHeader,01/06/2013 00:27:02.317,GetSIPHeaders,enter,
CL13_SIPHeader,01/06/2013 00:27:02.348,GetSIPHeaders,data,Cisco-Live,CVP Tips and Tricks Vol 2
CL13_SIPHeader,01/06/2013 00:27:02.348,GetSIPHeaders,data,Cisco-Live.location,London
CL13_SIPHeader,01/06/2013 00:27:02.348,GetSIPHeaders,data,Cisco-Live.session,BRKCCT-3030
CL13_SIPHeader,01/06/2013 00:27:02.348,GetSIPHeaders,data,User-Agent,CVP 9.0 (1) Build-670
CL13_SIPHeader,01/06/2013 00:27:02.348,GetSIPHeaders,exit,done
```

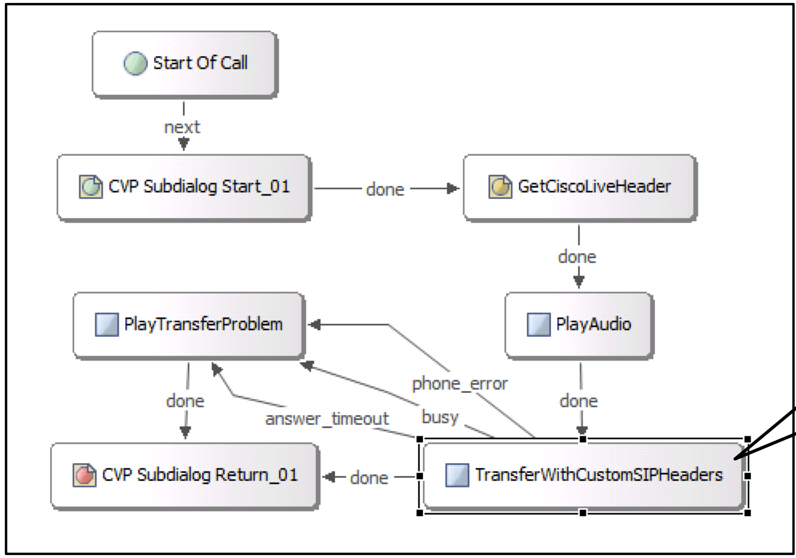
Setting SIP Headers

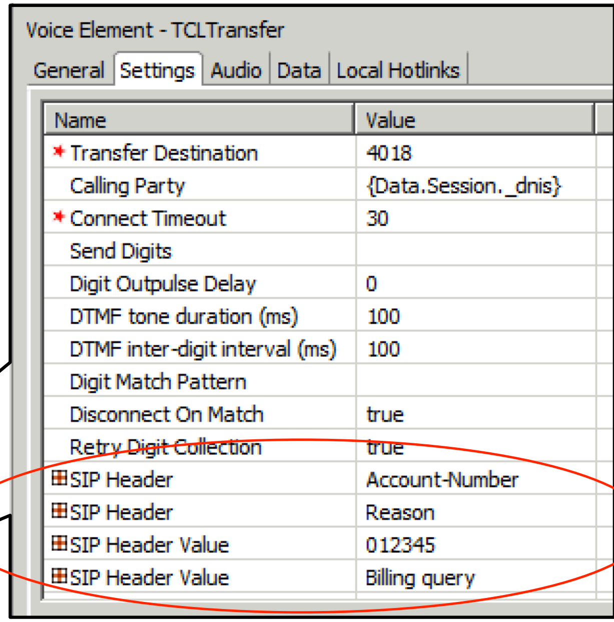
CVP Standalone Model

- Unfortunately no easy way to set SIP headers on VoiceXML transfers
- Necessary to handoff the call to TCL from VoiceXML
- An approach that enables a whole range of additional functionality, not just adding SIP headers
- Technique already used in several other places
  - CVP Standalone Outbound: sends SIP INFO messages to the voice gateway to make the call and return the outcome
  - Courtesy Callback: to initiate the callback on the ingress gateway
  - VideoConnect Element: transfers the caller to the video media server and listens for caller-side DTMF while video is playing
- Especially useful for adding custom transfer functionality as in the VideoConnect case

CVP Call Studio Application

Using Custom Transfers





- Custom transfer generates VoiceXML with <object> element to perform call handoff to TCL application cvp_tclxfer
- Data from element settings is passed to the TCL application via the arg-string param
- The TCL application retains control of the call during the transfer while the VoiceXML session is temporarily suspended

Extract from VoiceXML generated by custom transfer element

```
<object name="tclxfer" classid="builtin://com.cisco.callhandoff">
  <param name="return" expr="true" valuetype="data" />
  <param name="app-uri" expr="'builtin://cvp_tclxfer'" valuetype="data" />
  <param name="arg-string" expr="'dest=4018 rna=30 cli=651963499 pause=0 tonedur=100 tonegap=100 disc=true
reco=true siphdr=(Account-Number:012345&amp;Reason:Billing query)'" valuetype="data" />
</object>
```

CVP Call Studio Application

Using Custom Transfers

TCL transfer application is passed parameters from Call Studio script

```
CVP_TCLXFER, assumed control of call with argument: <dest=4018 rna=30 cli=651963499 pause=0 tonedur=100 tonegap=100 disc=true reco=true
siphdr=(Account-Number:012345&Reason:Billing query)>
```

Transfer leg is set up and SIP INVITE sent including custom SIP headers

```
INVITE sip:4018@10.58.16.175:5060 SIP/2.0
Via: SIP/2.0/UDP 10.58.16.170:5060;branch=z9hG4bK461E64D
From: <sip:651963499@10.58.16.170>;tag=C5AB8A80-F76
To: <sip:4018@10.58.16.175>
Date: Sun, 06 Jan 2013 18:05:10 GMT
Call-ID: 721DC9AF-576211E2-BD42CA6D-8EF6E38D@10.58.16.170
User-Agent: Cisco-SIPGateway/IOS-12.x
Reason: Billing query
Account-Number: 012345
Call-Info: <sip:10.58.16.170:5060>;purpose=X-cisco-forkingcapable
App-Info: <10.58.16.180:8000:8443>
Cisco-Live: CVP Tips and Tricks Vol 2;location=London;session=BRKCCT-3030
```

```
CVP_TCLXFER, event ev_proceeding on call leg 25268 received in state DIALING
CVP_TCLXFER, event ev_alert on call leg 25268 received in state DIALING
CVP_TCLXFER, event ev_connected on call leg 25268 received in state DIALING
CVP_TCLXFER, event ev_setup_done on call leg 25263 25268 received in state DIALING
CVP_TCLXFER, caller (leg 25263) connected to 4018 (leg 25268)
```

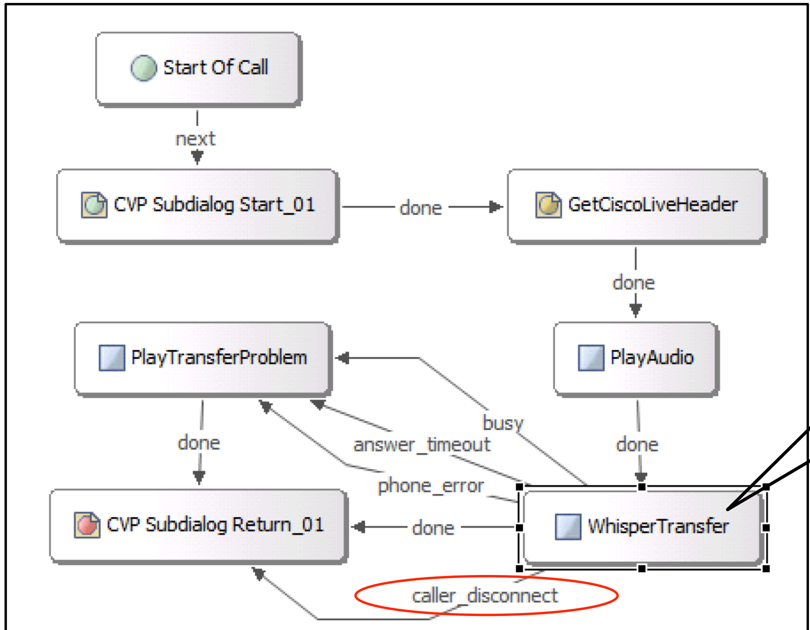
Called party answers and transfer is connected

```
CVP_TCLXFER, event ev_disconnected on call leg 25268 received in state SETUP_DONE
CVP_TCLXFER, far-end disconnected, returning caller to VoiceXML
CVP_TCLXFER, event ev_destroy_done received in state FAR_END_DISC
CVP_TCLXFER, handoff return with argstring <far_end_disconnect>
CVP_TCLXFER, event ev_disconnect_done on call leg 25268 received in state FAR_END_DISC
CVP_TCLXFER, exiting
```

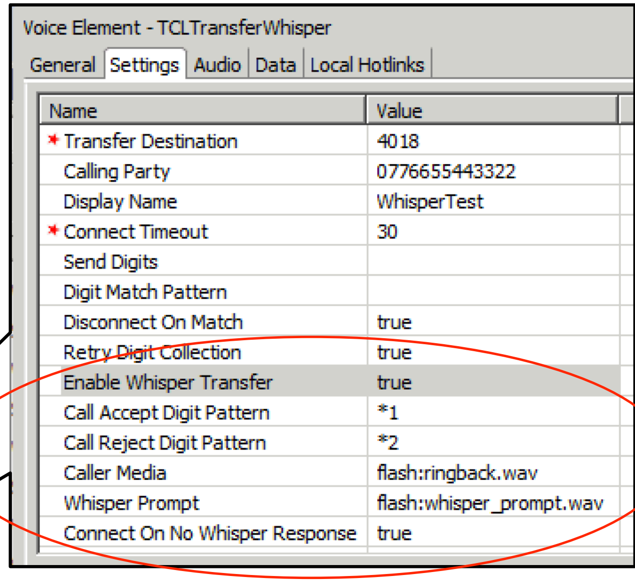
Called party hangs-up and call control is returned to VoiceXML

CVP Call Studio Application

Other Custom Transfer Capability



Also allows caller hangup during transfer to be caught and presented as an exit state to the application script



Voice Element - TCLTransferWhisper

General | Settings | Audio | Data | Local Hotlinks

Name	Value
* Transfer Destination	4018
Calling Party	0776655443322
Display Name	WhisperTest
* Connect Timeout	30
Send Digits	
Digit Match Pattern	
Disconnect On Match	true
Retry Digit Collection	true
Enable Whisper Transfer	true
Call Accept Digit Pattern	*1
Call Reject Digit Pattern	*2
Caller Media	flash:ringback.wav
Whisper Prompt	flash:whisper_prompt.wav
Connect On No Whisper Response	true

- Whisper transfer with explicit called party accept/reject
- Send DTMF to called party on answer
- Receive DTMF from called party
- End transfer on DTMF pattern match
- Configurable calling party number
- Configurable display name / remote party ID
- Call parking while other party connects-in (effectively reverse direction transfer)