Copyright (C) 2019 Gaman SP z o.o

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

UCCX Outbound Tricks & Tips

Have you thought about problems that you may experience once deploying outbound in UCCX? Below you can find information, tricks and potential problems that you may experience during the outbound deployment process. All topics are taken from real life based on Gaman experience (UCCX version 11.6).

# Outbound Licenses

One of the key elements of working outbound in UCCX are licenses. They need to be uploaded on UCCX before any outbound configuration process begins. Furthermore, they need to be properly uploaded.

In the license upload procedure, you can see that you need only to transfer file to the server. The instruction doesn’t mention that the system can be/should be restarted after valid upload.

During one of the projects Gaman team experienced strange behavior of the UCCX. Customer bought and uploaded the valid outbound license. License report indicated that the license is properly installed and ready to use. The problem appeared once the outbound call was made by dialer. When the call was transferred to the IVR Application connected to the outbound trigger, all the BA variable were not visible. System was able to connect the call, but the context was not outbound typical. Here is a log capture from MIVR service for working system:

*29474789: Nov 28 22:19:18.964 CET %MIVR-CFG\_MGR-7-UNK:[MIVR\_SS\_OB\_ReadContactsThread-83-0-ReadContactsThread] ConfigManagerImpl: ConfigManagerImpl-getAll():configObjs[0]=DialingListConfig[schema=DialingListConfig#4,time=2018-11-28 22:19:07.0,recordId=170,desc=,recordID=0,dialingListID=170,campaignID=2,accountNumber=﻿3423,firstName=Joe,lastName=Smith,phone01=XXXXXXXXXXX,phone02=,phone03=,gmtZonePhone01=455,dstPhone01=true,gmtZonePhone02=-1,dstPhone02=true,gmtZonePhone03=-1,dstPhone03=true,callbackNumber=,callbackDateTime=2018-11-28 21:19:07.0,callStatus=1,callResult=0,callResult01=0,callResult02=0,callResult03=0,lastNumberDialed=0,callsMadeToPhone01=0,callsMadeToPhone02=0,callsMadeToPhone03=0,numMissedCallback=0,retry=false,callsMadeToCallbackNum=0]*

As presented above, UCCX includes outbound variables (BA variables) and assigns them to call context. This log didn’t appear in non-rebooted UCCX. Problem disappeared once UCCX was rebooted.

<b>GAMAN team recommendation</b>

Gaman team highly recommends that after successful license upload, UCCX server/cluster needs to be rebooted.

# PSTN SIP Providers

When you look at the outbound call flow you will see that one of the mandatory parts of it is Call Progress Analysis (CPA). Please look at the screen shot taken from official Cisco doc (<https://www.cisco.com/c/en/us/support/docs/customer-collaboration/unified-contact-center-express/116084-trouble-ivr-dialer-00.html>):



1. The outbound IVR dialer determines the number of contacts to dial (as defined in the algorithm) and uses SIP to place outbound calls through the voice gateway.
2. The voice gateway detects non-live contact with its Call Progress Analysis (CPA) capabilities and sends the status of the non-live contact to the dialer. The dialer updates the contact status information in the configuration database.
3. The voice gateway detects live contact with its CPA capabilities and sends the status of the live contact to the dialer. The dialer updates the contact status information in the configuration database and sends a SIP refer message to the SIP gateway, which, in turn, transfers the call to the configured CTI route point on the Cisco Unified Communications Manager (CUCM).
4. The CUCM transfers the call to an IVR port on the Cisco UCCX server.
5. The IVR subsystem associates the call with the IVR application associated with the campaign. The engine starts execution of the application and an IVR session takes place between the IVR application for the campaign on UCCX and the customer contact.

During step 3 Cisco analysis the voice stream that comes from PSTN. In real life there are 2 possibilities:

1. Voice gateway is connected to E1/T1 circuits – in this type of deployment if the trunks are properly configured in most cases you should not experience issues.
2. Voice gateway acts as a CUBE and is connected to SIP trunk. Because SIP can have multiple standards the connection to PSTN may or may not work with dialer. Let’s focus on this case.

To get dipper into the problem we need to know how the voice streams are traveling once the call is answered by live person and the CPA is involved.

Dialer is a device that can only send SIP signaling and is not designed to stream audio to the customer – this is done by outbound IVR ports in the next phase of the call. The second call leg that connects Customer and the CUBE must be a live session (RTP packets need to travel from Customer to CUBE) to CPA (DSP resource) module/process. If this is not happening after 15 seconds dialer will drop the initial call.

In real life you can select between multiple vendors. Gaman team tested 2 most popular internet SIP trunk providers and one independent provider (X). Here are the results:

1. Twilio – dialer is supported (Appendix A)
2. Telnyx – dialer didn’t work, and the call was dropped as the voice stream didn’t arrive to CUBE. Gaman team raised the support case but we didn’t get any confirmation that the Telnyx team discovered the error on their end.
3. Provider (X) – dialer didn’t work, and the call was dropped as the voice stream didn’t arrive to CUBE.

As you can see the result is very poor only 1 out of 3 providers are able to support the dialer. But why this is happening this way. Because Gaman team got access to the diagrams – how to connect SIP with provider (X) – we did a deeper deep dive debugging.

From the received diagram we noticed that the provider (X) is using SBC (Session Border Controller) to terminate SIP connections. After quick look at the SBC RFC specification Gaman team noticed that SBC has on board “NAT traversal for media” feature:

*“The SBC also supports hosted NAT traversal for media and exhibits similar behavior for sending media as it does for signaling. More specifically, the addresses embedded in the SDP specification are, in terms of the SIP peer's private addressing, behind the NAPT device; whereas the SBC must send the RTP to the NAT'd public address/port. Since the RTP stream is generally symmetric, the SBC waits for RTP from the peer and then uses the source IP address/port from the received RTP packet as its destination IP address/port for the return direction.”*

In other words, if SBC has enabled NAT traversal media, it expects that the RTP packets will be initially send by the Dialer, but this will not happen.

Enabling/disabling this feature depends on the SBC version and may/may not be enabled for a customer.

For basic telephony it will work, as the call is always made by to endpoints that start to stream once the call is established from the signaling point of view.

For dialer we need to remember that the CPA is always used before the call is connected to Agent/IVR.

<b>GAMAN team recommendation</b>

Before you begin to deploy dialer based on PSTN SIP Trunks check with your provider if he supports Cisco Dialer. Good questions to ask are:

1. What is the termination device of the SIP Trunk – is it an SBC?
2. Is the NAT traversal media supported by provider, if so, can it be disabled for a connection?

# UCCX call handling for IVR Campaigns – facts, limitations

Have you ever wondered how UCCX treats the call transferred from Dialer to Outbound IVR Port? What are limitations based on this treatment and how to overcome them? All the comments in blue come from Cisco TAC.

Let’s start from the beginning and see how the IVR call is seen in the logs. Please look at the log:

*942359766: Nov 22 10:06:15.189 EDT %MIVR-SS\_TEL-7-UNK:[MIVR\_SS\_TEL\_TPG\_ROUTE\_EXE-41-434147-ROUTE\_CALL\_EV] CallImpl: Call.received() JTAPICallContact[id=23678,type=Cisco JTAPI Call,implId=12375879/3,active=true,state=CALL\_RECEIVED,****inbound=true****,handled=false,locale=en\_US,aborting=false,app=App[name=IVR\_TEST\_OB,type=Cisco Script Application,id=20,desc=IVR\_TEST\_OB,enabled=true,max=30,valid=true,cfg=[ApplicationConfig[schema=ApplicationConfig,time=2018-11-22 10:03:57.0,recordId=303,desc=IVR\_TEST\_OB,name=IVR\_TEST\_OB,type=Cisco Script Application,id=20,enabled=true,sessions=30,script=SCRIPT[OutboundIVR.aef],defaultScript=,vars=[<java.lang.String ApplicationParameter\_URL>,<java.lang.String ApplicationParameter\_DefaultCSQ>,<java.lang.Boolean ApplicationParameter\_RouteAllCallsToDefaultCSQ>],defaultVars=null]]],task=null,session=null,seqNum=-1,time=1542899175188,cn=4136,dn=4136,cgn=XXXXXXXXXXX,ani=null,dnis=null,clid=null,atype=DIRECT,lrd=null,ocn=4136,odn=null,uui=null,aniii=null,ced=null,OrigProtocolCallRef=0000000000BCD747032780C000000000,DestProtocolCallRef=null,route=RP[num=4136],port=null,aborting=false,transferring=false,disconnecting=false]*

Please look at the value: inbound=true. Although the call is made by UCCX Dialer, once transferred to IVR it is marked as inbound. Is this expected behavior – yes. Here is confirmation from Cisco TAC:

*“We have reviewed the logs and scripts provided.  The BU has determined this is expected behavior from using select resource or call redirect for outbound call to a trigger will result in an inbound call.”*

This means that every call initiated by Dialer from IVR Based campaign will be treated by UCCX as an inbound call. Having that in mind we need to ask what the consequences of this behavior are. During our deployment process we can think of 2:

1. Call back feature is not available once the call is transferred to agent. Here is an additional information from Cisco TAC:

“*The BU has determined this is expected behavior from using select resource or call redirect for outbound call to a trigger will result in an inbound call.  This means there will be no call back button.  You will need to use Agent predictive to permit the call back button.*”

The conclusion is simple – callback feature is only available in agent-based campaigns. What is also funny, in the IVR Campaign configuration you will see parameter called: Callback Time Limit (it appears on IVR and Agent based campaigns). When we asked TAC about this we got:

*“This setting is for a system generated call back, not an agent generated call back.  “during which the Outbound subsystem attempts to dial out a callback.”*

For us the system generated callback is called Retry in Cisco word. We didn’t accept this explanation. Please look at the conclusion.

Because this is not what customer wants to hear, Gaman team prepared a proposal of the solution that might be used to provide callback feature for IVR based campaigns. Read more here [https://gaman-gt.com/articles/uccx\_dialer\_ivr\_callback.html].

1. BA call variables are not directly transferred to Finesse. If you do a Reactive Debug you will see that the BA call variables are populated on the script level. Once the call is transferred to agent, all the call variables are blank. Does this mean they are erased? Probably not but mandatory thing is the call classification made before call is transferred. As you remember call is marked as inbound. This means that Finesse will not support the BA variables. What to do in this case – answer is simple, use either standard Peripheral call variables or ECC variables and rewrite to them values from standard BA variables.

<b>Conclusion</b>

Because of those problems and lack of information we continued our conversation with the TAC. At the end we received the following information:

*“After discussing this with the BU, it is determined that call back is not an option for IVR OB campaigns.*

*I will raise a document defect to get this corrected in the relevant documentation.*

*I will also raise another defect to have the callback configuration removed in the IVR campaigns for future versions.”*

And later update about Callback Time Limit parameter:

*“One follow up, since the call back timer is used in IVR campaigns for calculating the time range for retries (from the campaign), it cannot be removed.  Hopefully, it will be clarified and the documentation updated to help avoid this confusion in future releases.”*

# Cisco BUG CSCvg22356

Let’s start with complete bug description can be found at <https://bst.cloudapps.cisco.com/bugsearch/bug/CSCvg22356/?rfs=iqvred> .

As you probably saw this bug only affects a specific UCCX versions and may not be a problem in your deployment as the BABuddyName (conjunction of First Name and Last Name) may not be needed. But what to do if it’s mandatory?

The bug description gives us a one of possible solutions:

“*Use the phone number and/or account number to generate a database dip and pull the customer name and any other necessary information into the script.*”

Writing and establishing connection to UCCX database is not a simple thing to do (but is possible even without an Enhanced license and requires some coding skills). Here are 2 options that you may take for consideration once building your IVR script:

1. If the contacts are taken from API the script may call this API (in the same way as automated import) and pull the data directly to the script. We also recommend creating a separate method to pull a record based on the AccountNumber value.
2. Upload the import file to the UCCX document directory and reference this document from a script.

Both methods are not as efficient as exact query for one record. The efficiency may be different for different input sizes.

<b>GAMAN team recommendation</b>

Proposed 2 methods are workarounds and may be used only if the upgrade to higher version is not a simple process. When BABuddyName appearance is a mandatory thing, we recommend performing the UCCX upgrade as fast as possible.

# Appendix A - Twilio

In Twilio word PSTN SIP Access is called Elastic SIP Trunking. When you apply configuration for the look at the following parameter that can be found under: SIP Trunk > General Setting.



In the media section you will find parameter called: Symmetric RTP.

*“When Symmetric RTP is disabled, Twilio will send RTP to the destination negotiated in the SDP. This setting is more secure and therefore recommended.”*

In other words, when disabled, Twilio works as a Cisco CUBE, media is streamed to the IP address negotiated in SDP header. When enabled, Twilio acts as an SBC with NAT traversal for media feature enabled.

Please keep this in mind when working with Twilio.

If you would need help with Twilio configuration, please send us an email.