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 IVR Outbound Campaign Callback

# Propose

Information from Cisco indicate that the callback feature is not available for IVR based campaigns. Cisco will update documentation and UCCX Campaign UI for future releases indicating that this feature is not supported.

Because in outbound callback is one of the key features, Venga team prepared a solution that will fill gap in UCCX feature list. Below is high level description of proposed solution.

The presented solution is very complex (is build with multiple elements and contains multiple dataflows). Therefore, recommended approach to build entire solution is to spit it into smaller phases, ex:

1. Standard callback
2. Personal Callback
3. Callback status flow (Informix DB)

# Disclaimer

The following document describes only high-level description of the Callback feature design. Information’s exchanged between systems might change during low level design process.

# Data flow



# Dataflow description

Proposed Callback solution is designed to be an Outbound IVR Campaign for all skills, campaigns configured in the UCCX system. We have to possible scenarios:

1. If number (DDI/DID) presentation for customers is same for all campaigns, then there can be only one Callback Campaign
2. If number (DDI/DID) presentation for customers differs between campaigns, then number off callback campaigns will need to be equal to number of DDI’s/DID’s

In the above diagram we can distinguish 2 dataflows:

1. Callback dataflow – marked with numbers from 1 to 12
2. Callback status dataflow – marked with letters from A to C

### Callback dataflow

1. Agent logs to Finesse application.
2. At the beginning Finesse loads from the Finesse server basic list of setting and all gadgets defined for an agent. One of those gadgets is Callback Gadget. When UI is loaded, agent change status to ready.

During the day agent receives a customer call (originated by dialer via outbound IVR campaign). During conversation customer requests callback. Agent goes to Callback Gadget and enters parameters:

* Customer phone number – number will be auto populated but if customer wants to change it agent will have ability to enter new number
* Callback date and time
1. Callback Gadget communicates with Campaign Manager via Callback REST API passing the following information:
* Customer phone number
* Callback date and time
* Agent name – for personal callbacks
* Skill name – for regular (skill based) callbacks
* Customer First and Last Name
* Callback campaign name
1. When request is caught by Web server, all entered data are validated and checked if system will be able to make callback at the following time (check against Campaign Manager configuration). The checks that must be implemented:
* Validity of the phone number
* Validity of the set date
	+ Check if entered date is not a past date
	+ Check if skill is opened during the callback time
	+ Check if date is in the range of global rules set on the Campaign Manager
	+ Check against holidays

If all conditions are met, call back is written to database and waits for redial.

1. One of the key services deployed on the Campaign Manager is scheduler. During Callback dataflow it is responsible to pull scheduled callbacks and transfer them to dialer. The callback checks are performed on scheduled time period (ex. 1 min). Once the callback time conditions are met – information is pulled from Campaign Manager Database.
2. Based on the pulled data Scheduler passes this data to Web Client
3. Web client makes a UCCX API Rest call to import new contact to callback campaign.
4. Dialer begins to dial. Dialing process is a standard Dialer Outbound IVR dialing process where dialer exchanges information between Voice Gateway or Cube.
5. If the customer answers the call, Dialer will redirect it to standard Callback IVR Script (script provided as a part of Campaign Manager solution).
6. Triggered script makes a Callback API REST Call to Campaign Manager. At this point contact is distinguished by account number (this parameter must be passed to Campaign Manager).
7. Web server checks the Campaign Manager Database and pulls all information stored in the internal Database (agent name, skill). Information is returned to script.

At this point 2 scenarios may happen:

* Call can be queued based on the information about agent and skill (if agent name is available then system will try to queue the call to agent – personal callback).
* Call requires further processing (ex. gather additional data about client from internal systems). In this scenario call will be transferred to script for further processing. Name of the script will be configured on Campaign Manager configuration page.

**Note**:

If the call is transferred to external script all queueing must be build on this external script.

1. At the end call is queued to agent or a skill group. When agent becomes available call is transferred to agent. Callback is finished.
2. (Optional). Once callback arrives at agent workstation, Callback Gadget may send indication to Campaign Manager that the call arrived at agent. This option would also require designing additional flow for configuration (Callback Gadget needs to pull configuration from Campaign Manager).

### Callback status dataflow

Callback dataflow describes only how the callback is set by an agent and delivered to agent when the scheduled time comes. Second process that is marked on diagram show how Campaign Manager pulls information about dialing status for callbacks (dataflow marked with letters).

1. Based on regular time periods scheduler asks Informix database client to pull call result information’s from UCCX.
2. Informix Client connects directly to UCCX database and pulls all information only for callback campaigns. Once pulled data is transferred back to Scheduler.
3. Based on information, scheduler updates internal database and:
* Sets finish status to callbacks that where connected to agents
* Adjust information about call attempts

### Additional flows

Because the number of contacts that can reside on the UCCX dialing list is limited, system needs to have a way to purge the all finished contacts. Because of this Campaign Manager should have the ability to clear, purge all contacts on dialing list. This can be done on daily basis (recommended) or custom defined time period.

### To consider

Here are things that may be used as additional features for Callback feature:

1. Scenario: Customer request to reschedule the callback – customer was dialed back on set callback date time, but he is unable to speak. Open question – should this new callback be added to Campaign Manager as a new callback or current callback needs to be updated (all counters need to be reset)
2. The ability to enter customer number on callback gadget – what should be desired number format. If the system requires a dedicated phone number where to add additional number prefixes, digit manipulation. How to validate the number
3. Callback date time validation – should this feature be implemented, what criteria use for validation.

# Configuration

Because the proposed solution is complex and probably at the end there will be a lot of settings that business administrators would like to change the dedicated Web Application for configuration is needed (for example – recall settings for callbacks, number of retries).

# Reporting

All the Callback reporting will be made based on the Campaign Manager database. Reports can be available either on Campaign Manager nor CUIC. If the reports need to be available at the CUIC then the Campaign Manager database must be:

* MS SQL Server
* Informix

**Note**:

To verification if there is an option to add external data source in built-in CUIC (co-resident with UCCX). If it’s not possible then external CUIC is needed.

# Additional Info

The proposed solution may also be used in inbound call flow – ex in situation where customer is interested to continue conversation in later time.