Mindful Callback

Avaya Interaction Center (IC) Integration Guide

VHT documents and diagrams are sole ownership of VHT and use without VHT permission is prohibited. Depicted items are subject to change with future development and will be updated periodically to reflect any material changes with regard to design and logical operation. Please consult your VHT representative for additional information as it pertains to your particular deployment.

Contents

1. Document Summary

This document provides a guide to integrating Avaya Interaction Center (IC) as part of an end-to-end voice or digital callback flow using VHT Mindful Callback with an Avaya Aura platform.

2. Assumptions

This guide assumes the following:

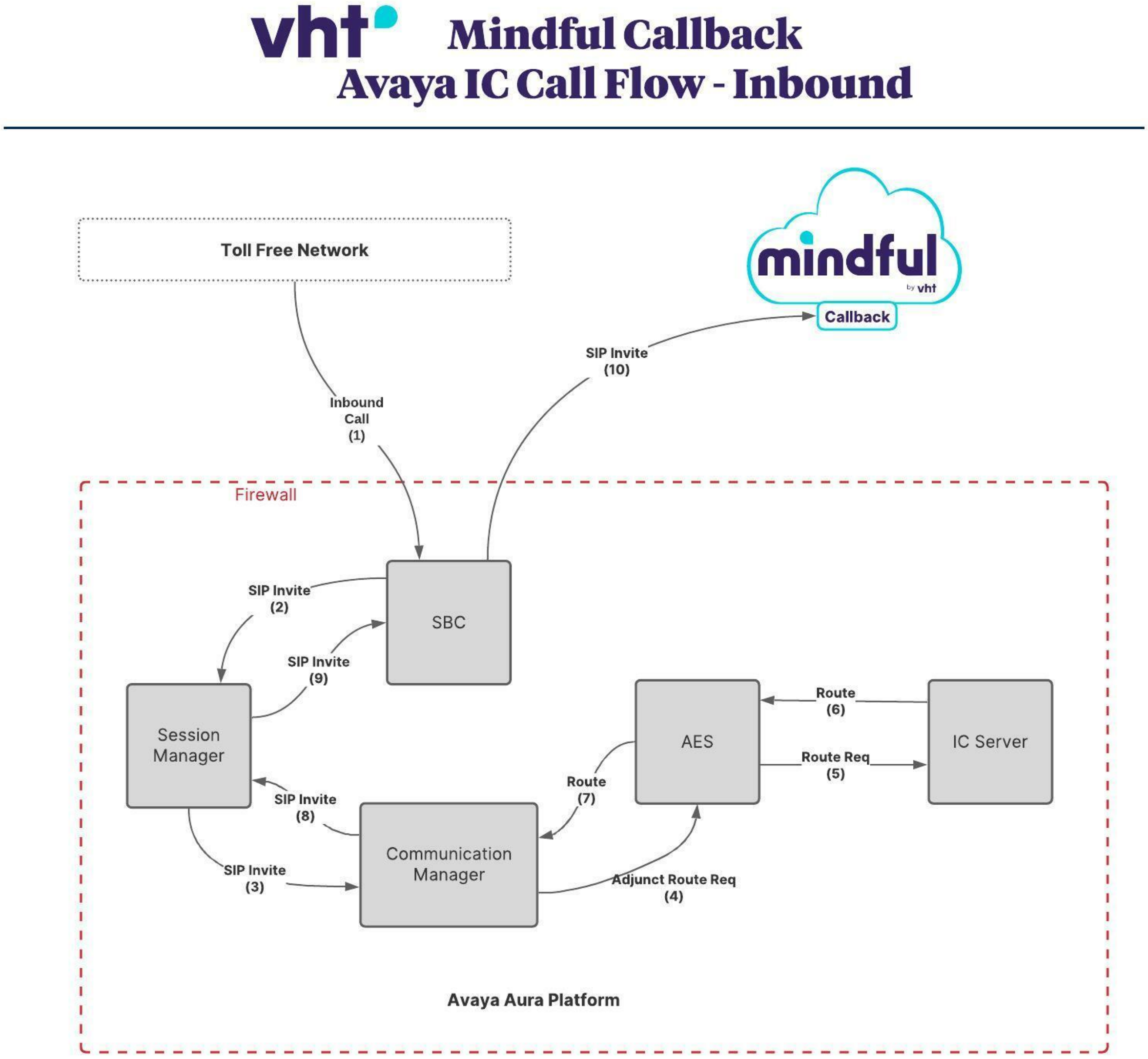
- For end-to-end voice call-flow, the inbound call and return (agent-leg) call are completely via SIP, and the Avaya platform configured to pass the User-to-User SIP header between Avaya Communication Manager and Mindful Callback.
- Avaya IC is already installed and configured, with the basic functionality to pass the EDU ID in the Avaya UUI and retrieve data from the IC database (e.g. for agent screen pop).
- For using Mindful Digital call-flow, the Avaya IC server should be able to reach the Mindful Datastore service API via HTTPs using the FQDN.

3. Definition Summary

Term	Description
AES	Avaya Application Enablement Server – an extension of the Avaya Aura platform that provides a set of APIs and CTI interfaces between Avaya Communication Manager and other Avaya / Third-party applications. This component acts as the CTI interface between CM and IC.
ANI (or CLI)	Automatic Number Identification – this is the number of the caller in a voice call. Also known as CLI (calling line identification) in some markets.
CM	Avaya Aura Communication Manager – the PBX server that forms the core element of the Avaya Aura platform.
DID	Direct Inward Dialing phone number
DNIS	Dialed Number Identification Service – this is the number of the callee (typically the number that is being dialed) in a voice call.
EDU	Electronic Data Unit – this is a unique data record used by Avaya IC to record and track an interaction with a contact. Once the interaction is over, the EDU is terminated and archived in the Avaya IC relational database. The EDU was formerly known as VDU (Voice Data Unit).
EDU ID	This is the unique ID of an EDU record. Formerly known as VDU ID.
IC	Avaya Interaction Center – a software suite that provides enhanced capabilities in an Avaya Contact Center via multi-channel workflows
Mindful Digital	A feature of Mindful Callback that allows callbacks to be requested via API from third party applications/platforms.
Mindful Datastore	This service adds the capability to store and retrieve custom data when registering a callback in Mindful Callback
PSTN	PSTN is the traditional circuit-switched telephone network that comprises of all the worlds telephone networks operated by local carriers.
RTP	Realtime protocol – the protocol used for the audio stream of a SIP call
SBC	Session Border Controller – the edge device that handles routing and security for SIP calls between a private network and a WAN or public network. In Avaya-based environments, this is usually the Avaya SBCE as it is designed to handle Avaya-specific SIP headers and events.
SDP	Session Description Protocol – part of the SIP message structure in specific SIP requests and responses, describing the media (audio).
SIP	Session Initiation Protocol
SM	Avaya Session Manager – a SIP proxy commonly used in Avaya Aura platforms.
UUI	User-to-User Information – the Avaya platform passes user data between CM, other Avaya applications, and third-party applications via CTI (using AES) or in the User-to-User SIP header. When used with Avaya IC, this will also contain the EDU ID.

4. Mindful Callback Call-Flow with Avaya IC

4.1 Typical Inbound Call-flow



(1) Customer places call into contact center,
(2) SBC invites Session Manager
(3) Session Manager in-turn sends an invite to Communication Manager, and call lands on the Inbound Entry VDN in CM

(4) Inbound Entry vector performs Adjunct step to give control of call to IC (if IC does not take control of call, the call is transferred to Queueing VDN to queue to skill at normal priority). Route request sent to AES

(5) Route request sent to IC Server from AES and IC invokes inbound workflow

(6) Inbound Workflow sends route instruction to AES to transfer call to the Inbound Mindful Transfer VDN

(7) AES forwards route instruction to Communication Manager and Call lands on Mindful Transfer VDN

(8) Mindful Transfer Vector routes call to Mindful Callback. If transfer fails, call is transferred to Queueing VDN to queue to skill at normal priority. CM sends invite to Session Manager using provisioned number in Mindful Callback.

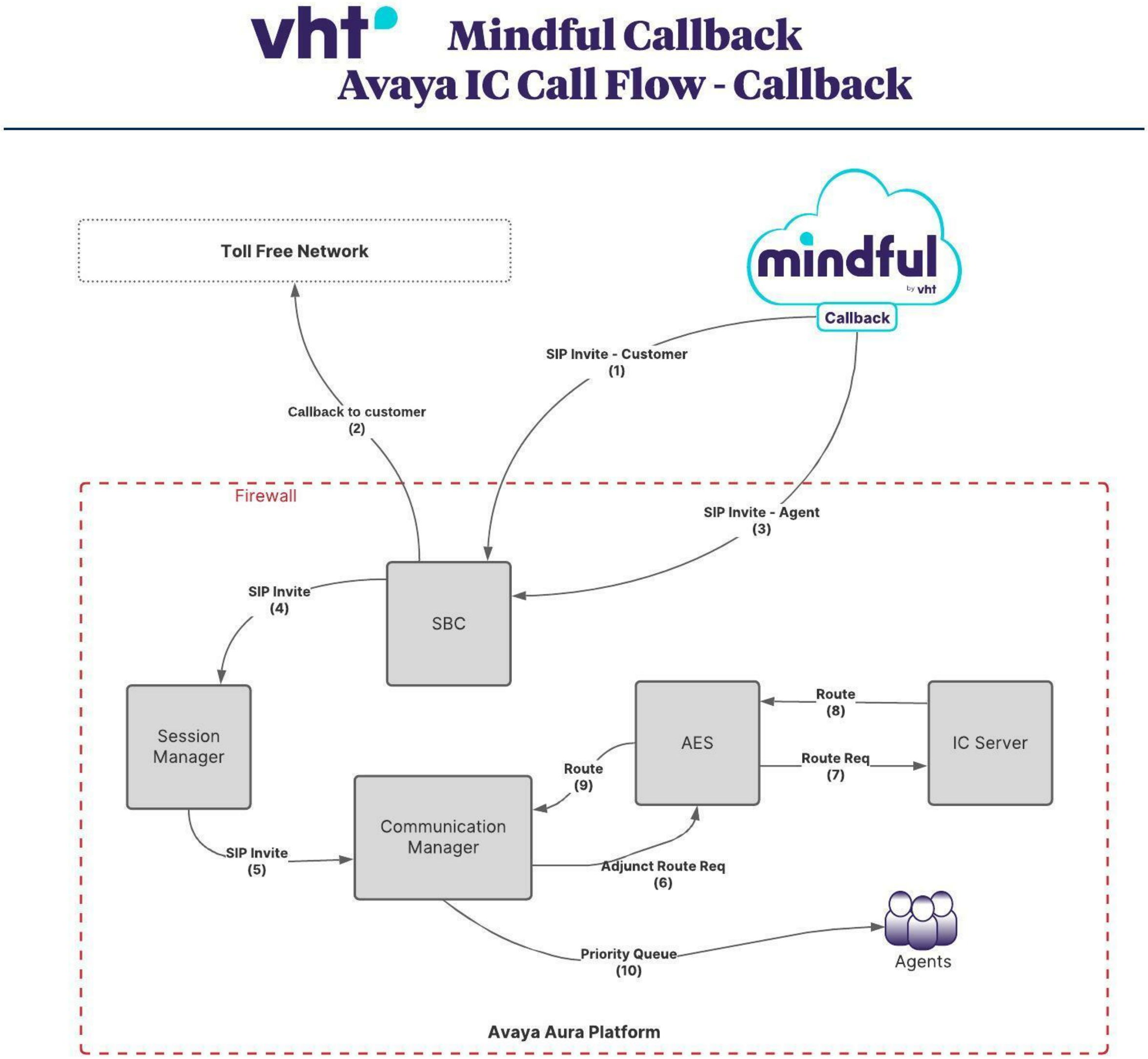
(9) Session Manager sends invite to SBC and Mindful Callback trunk is selected based on URI number.

(10) SBC invites Mindful Callback SIP Proxy and call arrives into Mindful Callback

   a. caller is offered callback
   b. caller accepts offer and confirms phone number to receive callback to
   c. the encoded UUI containing EDU ID is stored with the callback request in Mindful Callback.
   d. Call ends and caller waits for callback.

Note that if the Mindful Callback call-target is configured to allow caller to choose to remain on hold, the call will be transferred back into the Avaya to queue on the Queueing VDN at normal priority. The UUI containing the EDU ID is still associated with the call, so can be used for agent screen pop etc.

4.2 Typical Callback call-flow (customer-first)



(1) Mindful Callback calls customer on callback number – sends invite to SBC

(2) SBC sends invite out to PSTN to the number the customer requested a callback on

   a. Customer answers and presses 1 to speak to the next available agent.

(3) Mindful Callback sends SIP INVITE into to SBC with number of Callback Entry VDN

(4) SBC forwards invite to Session Manager

(5) Session Manager sends invite to Communication manager

   a. Call lands on Callback Entry VDN

(6) Callback Entry vector performs Adjunct step to give control of call to IC (if IC does not take control of call, the call is transferred to Priority Queueing VDN to queue to skill at high priority). Route request sent to AES

(7) AES sends route request to IC Server

    a. IC invokes callback workflow and checks for EDU ID in the UUI
    b. If EDU ID has expired, contact is checked using ANI against the database, and a new EDU ID is assigned.

(8) Workflow issues transfer route instruction to AES to send call to the Callback Queue VDN

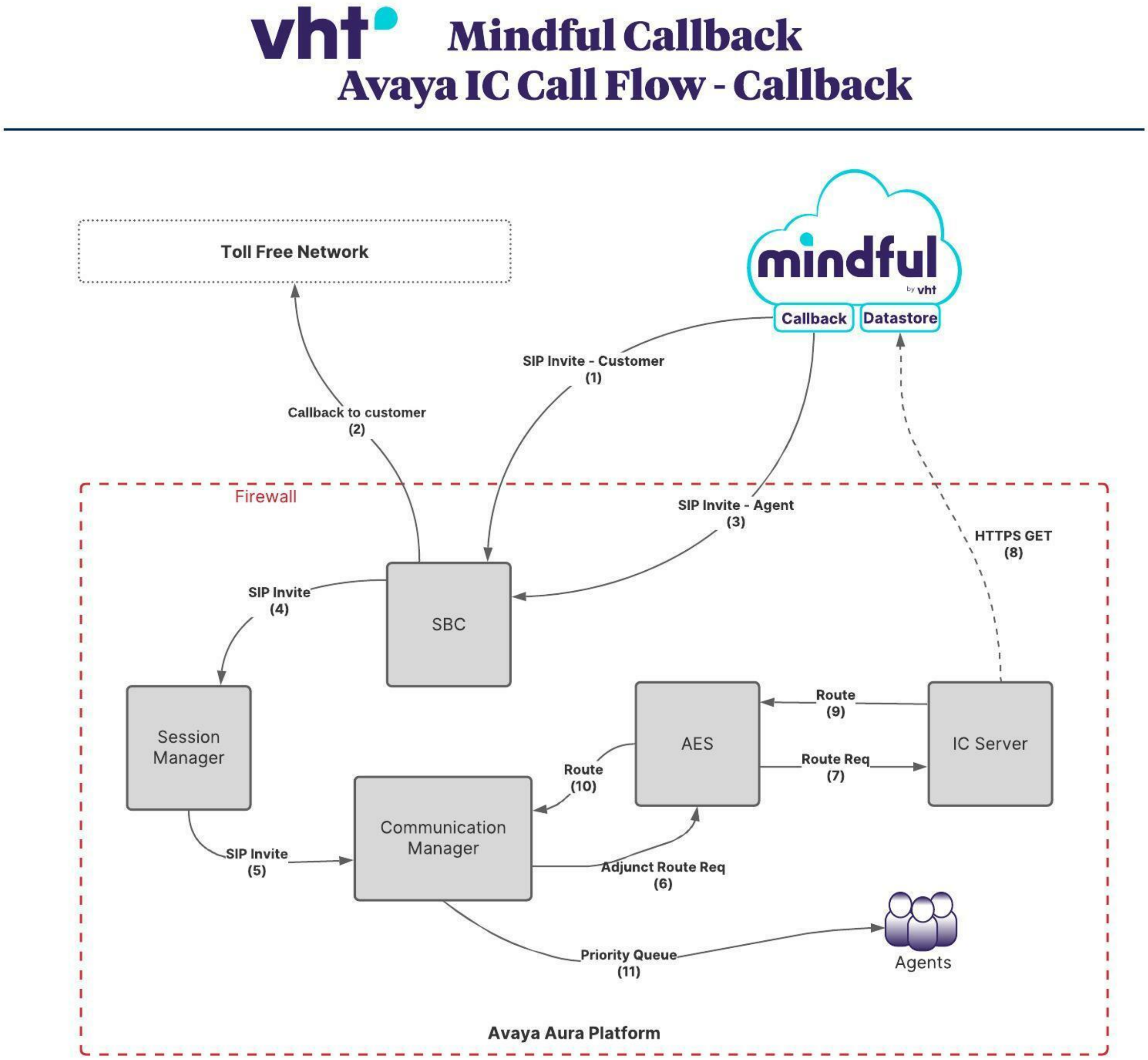
(9) AES forwards route instruction to Communication Manager

    a. Call lands on the Priority Queuing VDN to queue to skill at high priority.
    b. Agent answers call and data is retrieved from EDU by agent desktop application

Note that this shows a typical customer-first call flow. Mindful Callback also supports an agent-first call flow in which the agent leg is initiated first, and once answered by an agent, the customer leg is initiated. However, the configuration is the same regardless of which method is used.

4.3 Typical Digital Callback call-flow (customer-first)

A Mindful Digital callback is generated using the Widget APIs associated with each provisioned call-target in Mindful Callback, and the associated user data for the callback is stored in the MNindful Datastore using a HTTP POST. The use of these APIs are not in-scope for this document, and are referenced in the VHT Mindful Digital and Mindful Datastore documentation.



(1) Mindful Callback calls customer on callback number – sends invite to SBC
(2) SBC sends invite out to PSTN to the number the customer requested a callback on
    a. Customer answers and presses 1 to speak to the next available agent.
(3) Mindful Callback sends SIP INVITE into to SBC with number of Callback Entry VDN
(4) SBC forwards invite to Session Manager
(5) Session Manager sends invite to Communication manager

    a. Call lands on Callback Entry VDN

(6) Callback Entry vector performs Adjunct step to give control of call to IC (if IC does not take control of call, the call is transferred to Priority Queueing VDN to queue to skill at high priority). Route request sent to AES

(7) AES sends route request to IC Server

    a. IC invokes callback workflow and checks for EDU ID in the UUI
    b. If EDU ID has expired, contact is checked using ANI against the database, and a new EDU ID is assigned.

(8) Workflow checks to see if the UUI is empty, if so, this assumes the callback is a Mindful Digital Call and the custom java class is invoked to fetch data from the Mindful Datastore based on the callback ANI. The received data is written to the active EDU record

(9) Workflow issues transfer route instruction to AES to send call to the Callback Queue VDN

(10) AES forwards route instruction to Communication Manager

    a. Call lands on the Priority Queuing VDN to queue to skill at high priority.
    b. Agent answers call and data is retrieved from EDU by agent desktop application

Note that this shows a typical customer-first call flow. Mindful Callback also supports an agent-first call flow in which the agent leg is initiated first, and once answered by an agent, the customer leg is initiated. However, the configuration is the same regardless of which method is used.

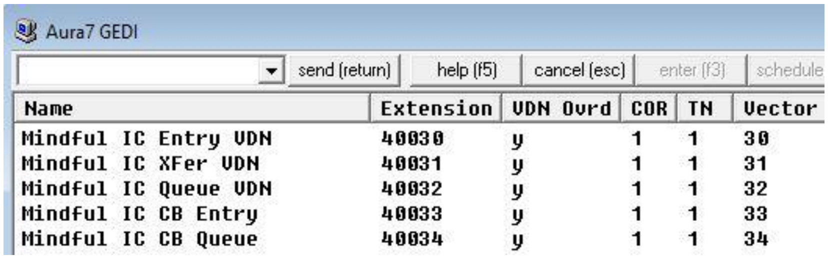
5. Avaya CM – Sample VDN and Vector configuration

This section shows five sample VDNs and vectors, configured in Communication Manager and used in a typical inbound and callback call-flow.   Note that if only using Mindful Digital to request callbacks, the Inbound Entry and Inbound Xfer VDNs and vectors may not be required.

Note: The example in this section shows five VDNs and five vectors for a single test skill/call-target. However, it may be possible to re-use vectors across multiple skills/call-targets using vector variables and VDN skills rather than vector skills.

5.1 VDNs

The five VDNs created will look something like this:



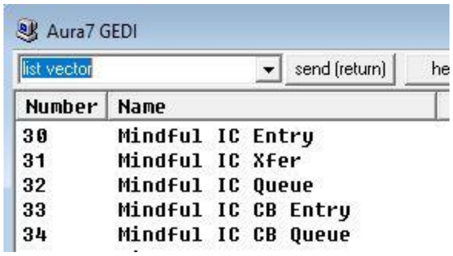
Name	Extension	VDN Ovrd	COR	TN	Vector
Mindful IC Entry VDN	40030	y	1	1	30
Mindful IC XFer VDN	40031	y	1	1	31
Mindful IC Queue VDN	40032	y	1	1	32
Mindful IC CB Entry	40033	y	1	1	33
Mindful IC CB Queue	40034	y	1	1	34

These are:

- Mindful IC Entry – this VDN is where the inbound call will land just before queueing to skill and will invoke the IC inbound (incomingcall) workflow.
- Mindful IC Xfer – this VDN is where IC will send the Inbound call once it has executed the Inbound (incomingcall) workflow, to be transferred to the provisioned number for Mindful Callback
- Mindful IC Queue VDN – This VDN is used to queue the call to skill at normal priority, and the Inbound call may be routed to this VDN if the IC adjunct step or the transfer to Mindful Callback fails, or if the caller chooses not to take the callback option.
- Mindful IC CB Entry – the agent leg of a Mindful Callback will arrive onto this VDN, where the vector will perform an adjunct step to allow IC to take control of the call and execute the callback workflow.
- Mindful IC CB Queue - this VDN is where IC will send the callback once it has executed the callback workflow, to queue to skill at high priority.

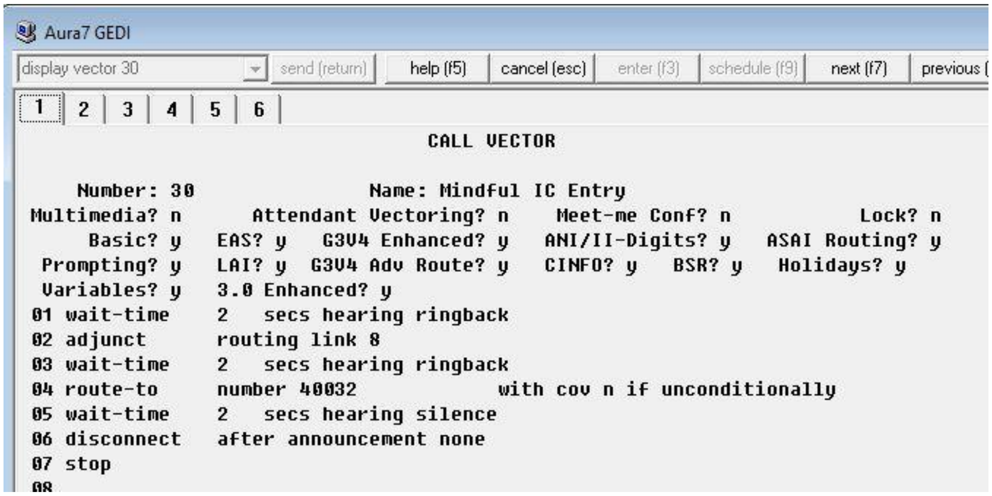
5.2 Vectors

For each VDN, a corresponding vector was created:



```
Number  Name
30      Mindful IC Entry
31      Mindful IC Xfer
32      Mindful IC Queue
33      Mindful IC CB Entry
34      Mindful IC CB Queue
```

5.2.1 Sample Inbound Entry Vector

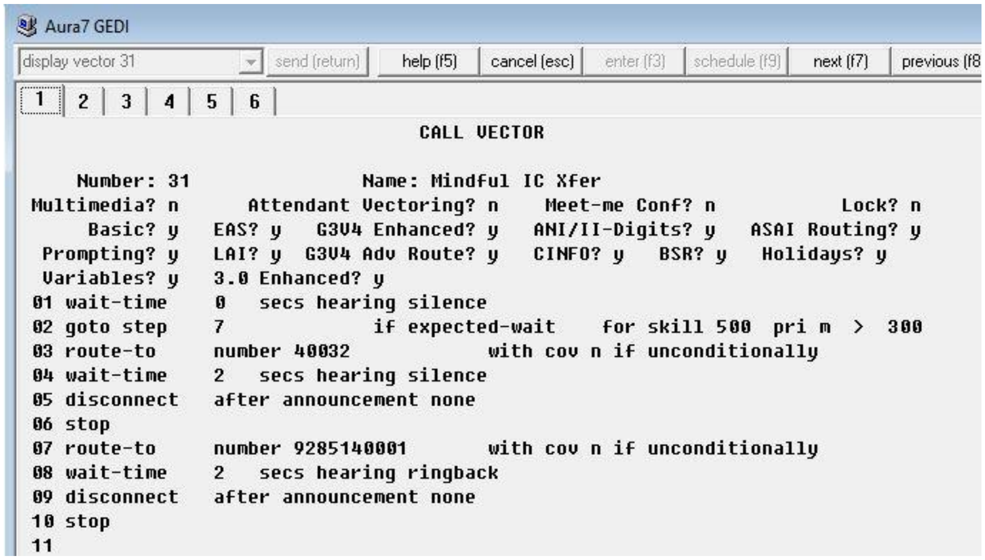


```
                                   CALL VECTOR


    Number: 30              Name: Mindful IC Entry
Multimedia? n      Attendant Vectoring? n    Meet-me Conf? n           Lock? n
     Basic? y   EAS? y   G3V4 Enhanced? y   ANI/II-Digits? y   ASAI Routing? y
 Prompting? y   LAI? y  G3V4 Adv Route? y   CINFO? y    BSR? y   Holidays? y
 Variables? y   3.0 Enhanced? y
01 wait-time       2   secs hearing ringback
02 adjunct         routing link 8
03 wait-time       2   secs hearing ringback
04 route-to        number 40032           with cov n if unconditionally
05 wait-time       2   secs hearing silence
06 disconnect      after announcement none
07 stop
08
```

This vector:

- Performs a routing link step against the AES cti link that Avaya IC is connected to. This sends a route request to IC to allow it to take control of the call
- Waits a few seconds for IC to respond
- If IC does not respond, the call is sent to the Queue VDN to queue to Skill at normal priority

5.2.2 Sample Inbound Transfer to Mindful Vector

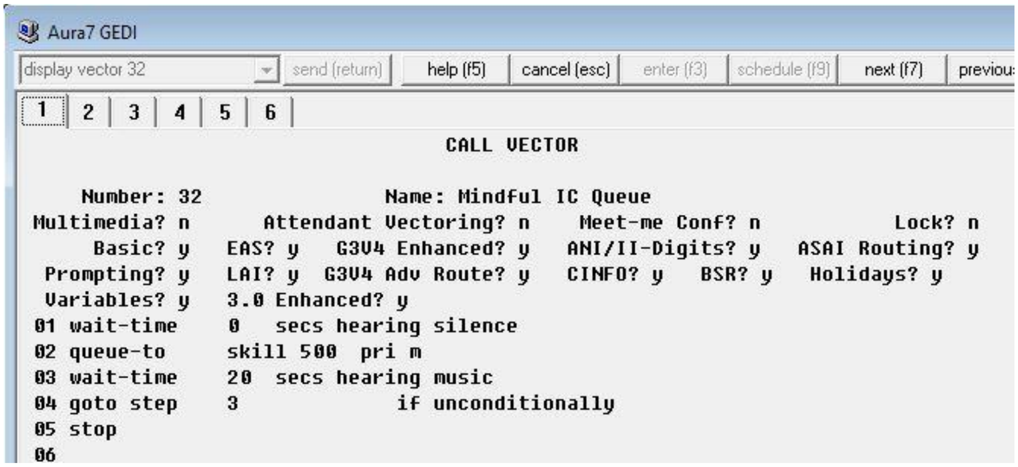


```
                                 CALL VECTOR

    Number: 31                Name: Mindful IC Xfer
Multimedia? n      Attendant Vectoring? n    Meet-me Conf? n           Lock? n
     Basic? y    EAS? y    G3V4 Enhanced? y    ANI/II-Digits? y    ASAI Routing? y
 Prompting? y    LAI? y   G3V4 Adv Route? y    CINFO? y    BSR? y    Holidays? y
 Variables? y    3.0 Enhanced? y
01 wait-time      0   secs hearing silence
02 goto step      7              if expected-wait    for skill 500  pri m  >  300
03 route-to       number 40032           with cov n if unconditionally
04 wait-time      2   secs hearing silence
05 disconnect     after announcement none
06 stop
07 route-to       number 9285140001       with cov n if unconditionally
08 wait-time      2   secs hearing ringback
09 disconnect     after announcement none
10 stop
11
```

This vector:

- Receives the call after the IC inbound (incomingcall) workflow transfers the call out to the VoiceQueueId (the VDN that calls this vector).
- Checks the EWT for the target skill of the call, and if EWT is greater than the configured threshold (example shows 300 seconds / 5 minutes), performs a route-to step to send the call to the SIP number provisioned against the corresponding call target in Mindful Callback.
- If EWT is under the configured threshold, or the transfer fails, then the call is sent to the Inbound Queue VDN to queue to Skill at normal priority

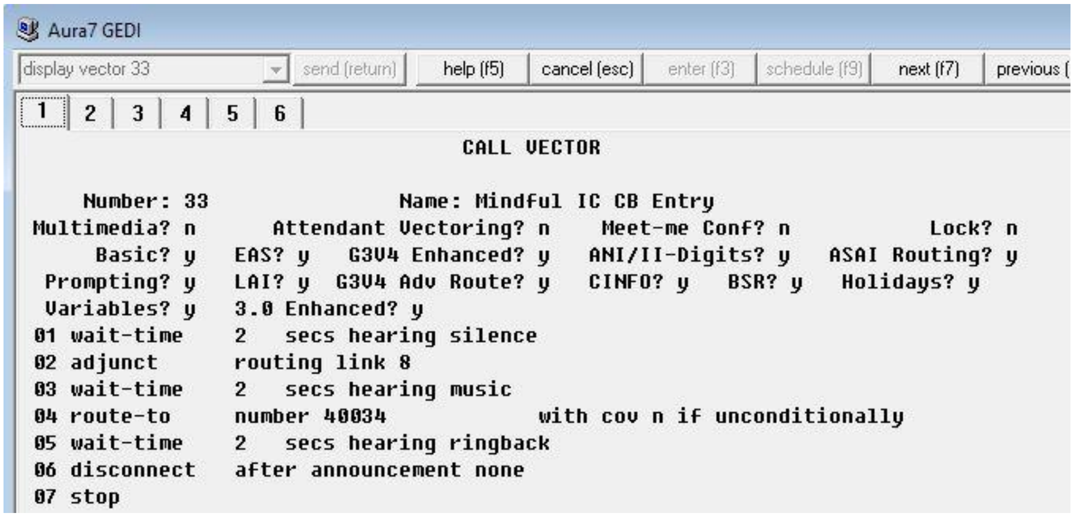
5.2.3 Sample Inbound Queue Vector



```
                                 CALL VECTOR

    Number: 32                Name: Mindful IC Queue
Multimedia? n      Attendant Vectoring? n    Meet-me Conf? n           Lock? n
     Basic? y    EAS? y    G3V4 Enhanced? y    ANI/II-Digits? y    ASAI Routing? y
 Prompting? y    LAI? y   G3V4 Adv Route? y    CINFO? y    BSR? y    Holidays? y
 Variables? y    3.0 Enhanced? y
01 wait-time      0   secs hearing silence
02 queue-to       skill 500  pri m
03 wait-time      20  secs hearing music
04 goto step      3              if unconditionally
05 stop
06
```

This vector simply queues to skill at normal priority

5.2.4 Sample Callback Entry Vector

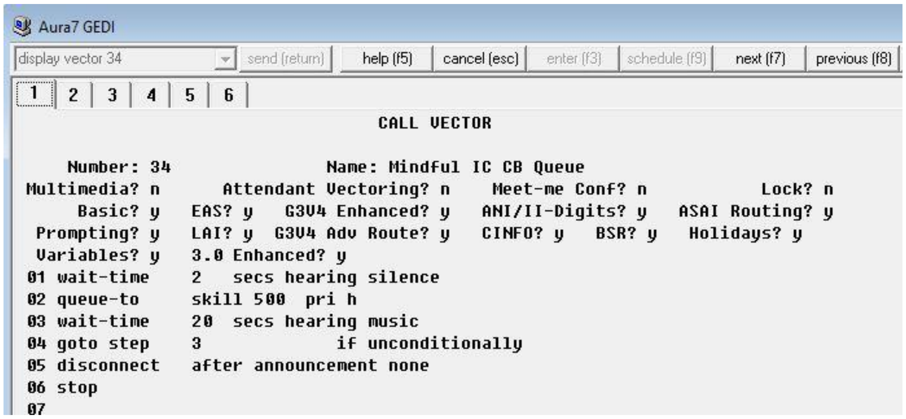


```
                                  CALL VECTOR

    Number: 33                   Name: Mindful IC CB Entry
Multimedia? n       Attendant Vectoring? n     Meet-me Conf? n            Lock? n
     Basic? y   EAS? y   G3V4 Enhanced? y   ANI/II-Digits? y   ASAI Routing? y
 Prompting? y   LAI? y  G3V4 Adv Route? y   CINFO? y    BSR? y   Holidays? y
 Variables? y   3.0 Enhanced? y
01 wait-time     2   secs hearing silence
02 adjunct       routing link 8
03 wait-time     2   secs hearing music
04 route-to      number 40034            with cov n if unconditionally
05 wait-time     2   secs hearing ringback
06 disconnect    after announcement none
07 stop
```

This vector:

- Performs a routing link step against the AES cti link that Avaya IC is connected to. This sends a route request to IC to allow it to take control of the call
- Waits a few seconds for IC to respond
- If IC does not respond, the call is sent to the Callback Queue VDN to queue to Skill at high priority

5.2.5 Sample Callback Queue Vector



```
                                  CALL VECTOR

    Number: 34                   Name: Mindful IC CB Queue
Multimedia? n       Attendant Vectoring? n     Meet-me Conf? n            Lock? n
     Basic? y   EAS? y   G3V4 Enhanced? y   ANI/II-Digits? y   ASAI Routing? y
 Prompting? y   LAI? y  G3V4 Adv Route? y   CINFO? y    BSR? y   Holidays? y
 Variables? y   3.0 Enhanced? y
01 wait-time     2   secs hearing silence
02 queue-to      skill 500   pri h
03 wait-time     20  secs hearing music
04 goto step     3            if unconditionally
05 disconnect    after announcement none
06 stop
07
```

This vector:

- Receives the call after the IC callback workflow transfers the call out to the VoiceQueueId (the VDN that calls this vector).
- Queues to skill at high priority

6. Avaya IC Configuration

6.1 Mindful Digital Call flow - Additional Files

When using the Mindful Digital call flow with Avaya IC, some additional files need to be copied onto the Avaya IC server – note that if Mindful Digital is not going to be used, these files are not required. These files will be provided by the VHT Solution Delivery team, and should be copied onto the following locations:

- vhtob.jar – this java archive file contains the applet that allows the Callback IC workflow to retrieve contact data from the VHT Mindful Datastore API via https GET. This file should be located into a folder on the IC Server– e.g. D:/VHT/ - this will be referenced later on during the configuration in IC Manager.
- ICJavaInterface.jar – this should be copied into the Avaya IC install folder, into the Java\jar folder - e.g. D:\Avaya\IC73\Java\jar\   - note that if the ICJavaInterface.jar file exists in this folder, rename the old one (e.g. ICJavaInterface.old) and copy the provided one in.
- config.properties – this file contains the URL and auth token for Mindful Datastore, and should be placed into the bin folder in the IC Server install folder – e.g. D:\Avaya\IC73\bin   - note that in some IC installations, this file may need to be located in the etc folder – e.g. D:\Avaya\IC73\etc. It is advisable to place the file into both locations to be sure it will be correctly found by the java applet.

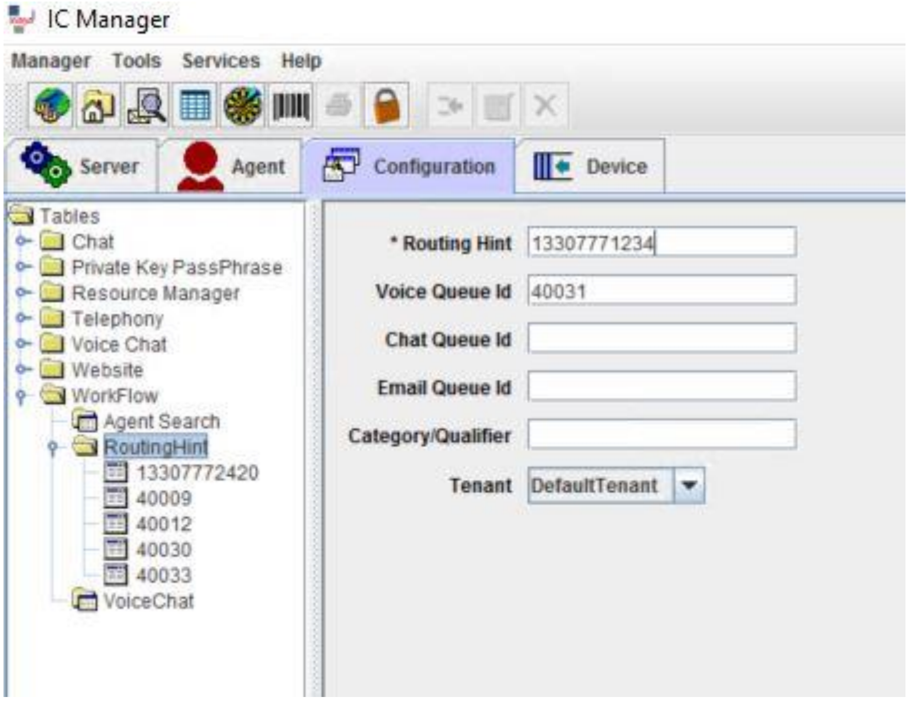
6.2 IC Manager – Additional Configuration

There are some additional configuration steps for Mindful Callback integration – some of these steps may already be complete in your current IC configuration so please skip any steps that would involve duplicating existing config, although take care to make sure any settings are as per the sample configuration below.

6.2.1 Configuration Section

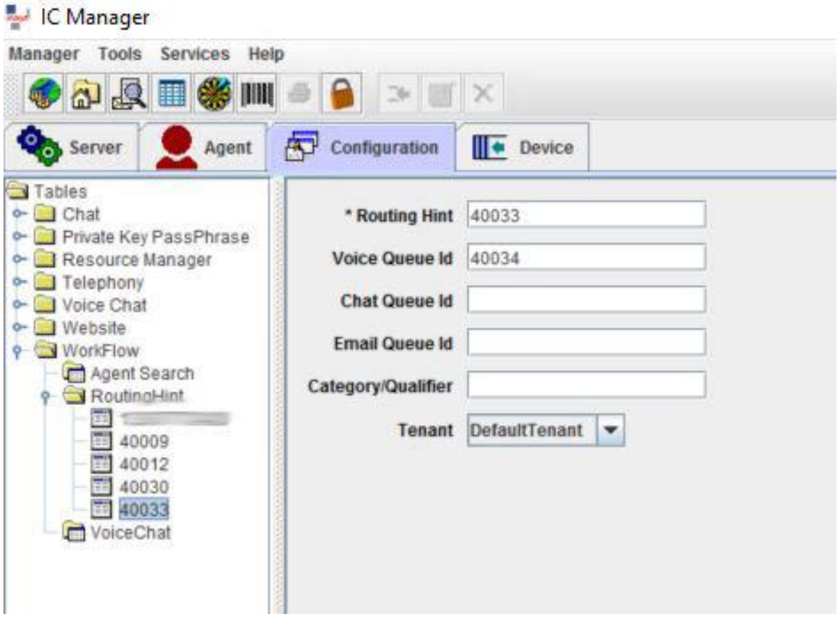
Under the Configuration tab in IC Manager, open the Workflow folder, and then the RoutingHint folder. Add a new Routing Hint by clicking on the New button. In the new Routing Hint page, enter the number of the Inbound Entry VDN as the Routing Hint, and the Inbound DNIS as the Voice Queue Id – in this example the

DID 13307771234 is the DNIS of an inbound call that arrives into Inbound Entry VDN 40030 – so this is mapped to the Mindful Transfer VDN as the transfer destination.



* Routing Hint: 13307771234
Voice Queue Id: 40031
Chat Queue Id:
Email Queue Id:
Category/Qualifier:
Tenant: DefaultTenant

Create another new Routing Hint entry for the Callback, this time entering the Callback Entry VDN number as the Routing Hint, and the Callback DNIS (this would typically be the Callback Queue VDN number) as the Voice Queue Id:

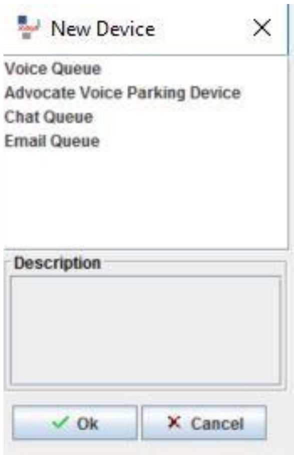


* Routing Hint: 40033
Voice Queue Id: 40034
Chat Queue Id:
Email Queue Id:
Category/Qualifier:
Tenant: DefaultTenant

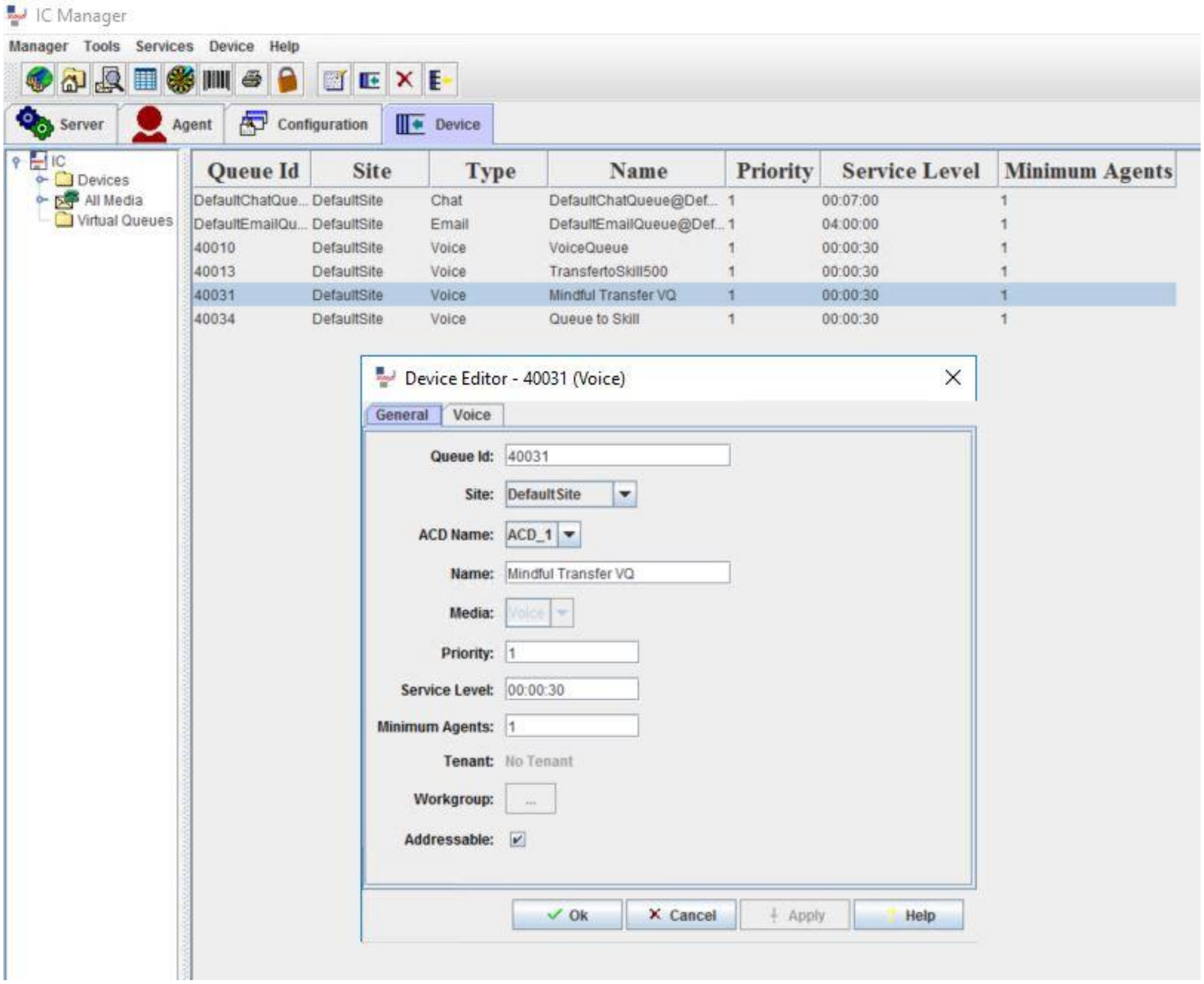
6.2.2 Device Section

Under the Device tab in IC Manager, open the All Media folder and then the Voice section. Click on the create queue button .

A new queue window will pop up – select Voice Queue and then OK:



In the Device Editor window that will now pop up, on the General tab, enter the Mindful Transfer VDN number, select the site (usually the default site), select the pre-configured ACD, and then give the Device a name (e.g. Skill1_Mindful_Xfer) – those are the only parameters that need to be set on this page:



Queue Id	Site	Type	Name	Priority	Service Level	Minimum Agents
DefaultChatQue...	DefaultSite	Chat	DefaultChatQueue@Def...	1	00:07:00	1
DefaultEmailQu...	DefaultSite	Email	DefaultEmailQueue@Def...	1	04:00:00	1
40010	DefaultSite	Voice	VoiceQueue	1	00:00:30	1
40013	DefaultSite	Voice	TransfertoSkill500	1	00:00:30	1
40031	DefaultSite	Voice	Mindful Transfer VQ	1	00:00:30	1
40034	DefaultSite	Voice	Queue to Skill	1	00:00:30	1

Queue Id: 40031

Site: DefaultSite

ACD Name: ACD_1

Name: Mindful Transfer VQ

Media: Voice

Priority: 1

Service Level: 00:00:30

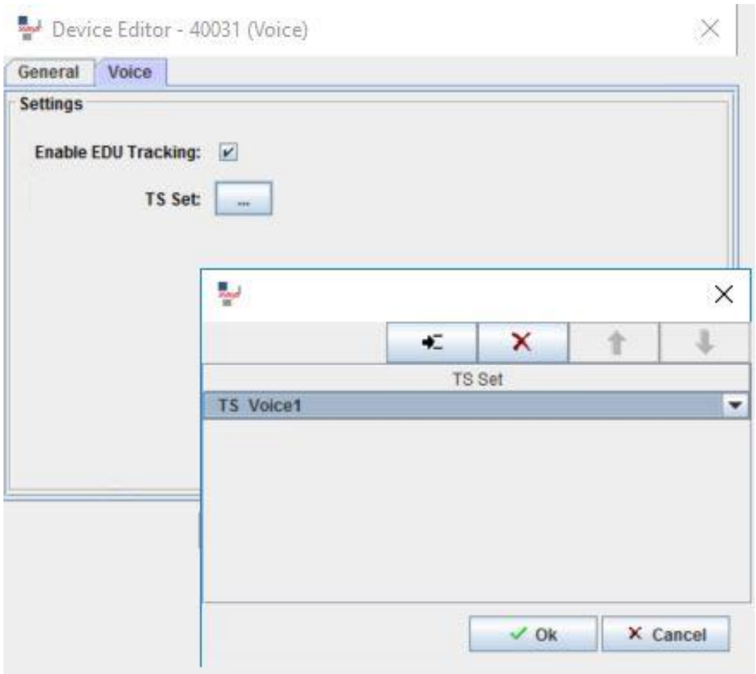
Minimum Agents: 1

Tenant: No Tenant

Workgroup: ...

Addressable: ☑

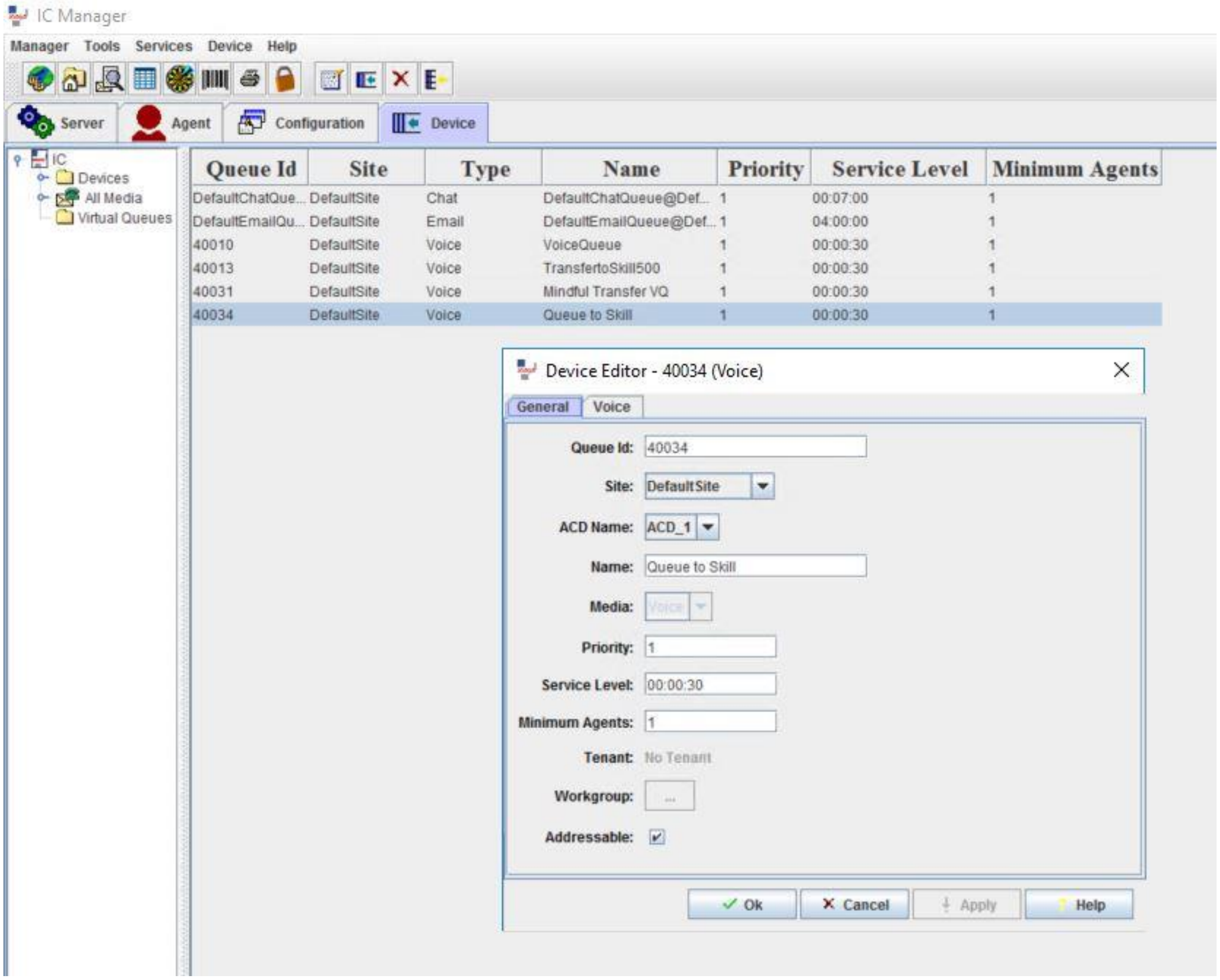
Now click on the Voice tab,



On this page, make sure Enable EDU Tracking is checked, and click on the TS Set button. A new popup will open, in which the Telephony Server (TS) must be selected.

Click OK, and then OK again to close the TS and Device windows.

Repeat the above steps to create another device (new queue button), and this time the Queue Id will be the Callback Queue VDN number:



Queue Id	Site	Type	Name	Priority	Service Level	Minimum Agents
DefaultChatQue...	DefaultSite	Chat	DefaultChatQueue@Def...	1	00:07:00	1
DefaultEmailQu...	DefaultSite	Email	DefaultEmailQueue@Def...	1	04:00:00	1
40010	DefaultSite	Voice	VoiceQueue	1	00:00:30	1
40013	DefaultSite	Voice	TransfertoSkill500	1	00:00:30	1
40031	DefaultSite	Voice	Mindful Transfer VQ	1	00:00:30	1
40034	DefaultSite	Voice	Queue to Skill	1	00:00:30	1

Queue Id: 40034

Site: DefaultSite

ACD Name: ACD_1

Name: Queue to Skill

Media: Voice

Priority: 1

Service Level: 00:00:30

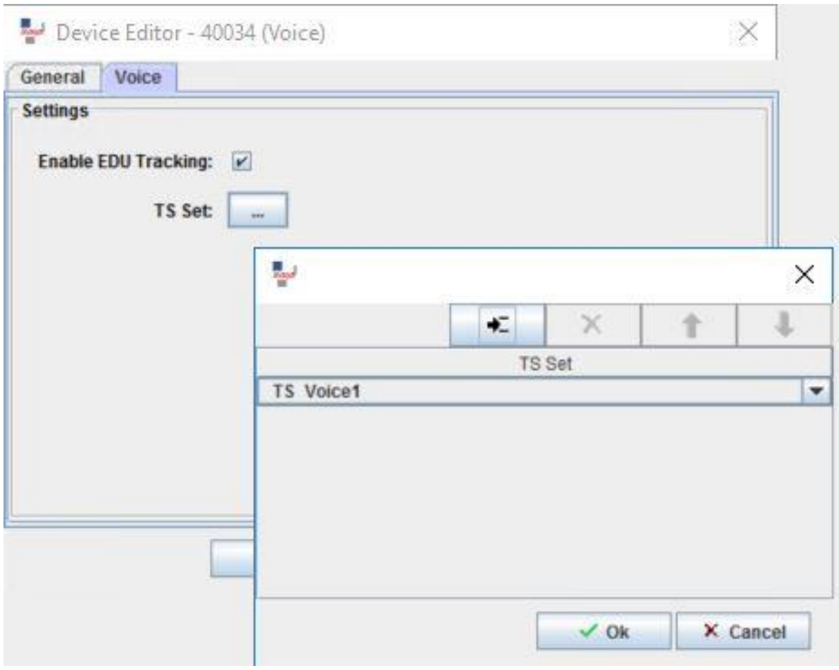
Minimum Agents: 1

Tenant: No Tenant

Workgroup: ...

Addressable: ☑

Again, once this page is configured, click on the voice tab:



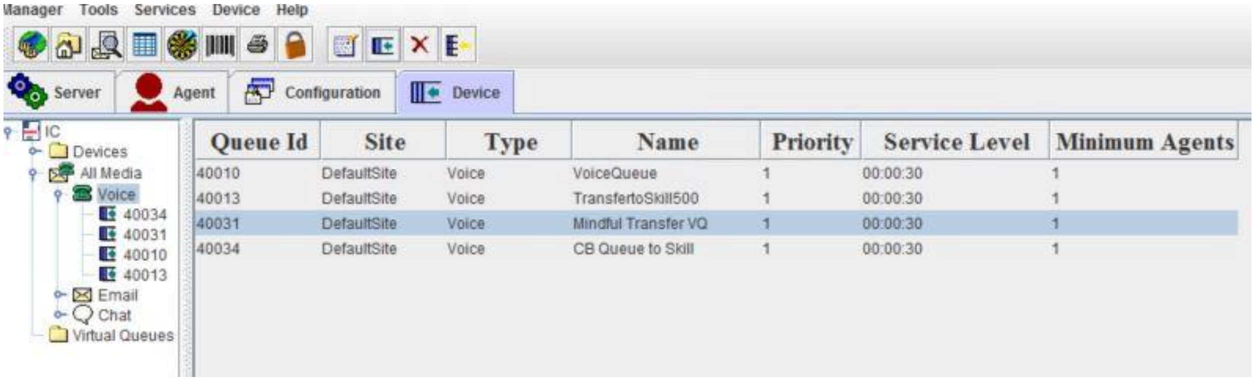
Enable EDU Tracking: ☑

TS Set: ...

TS Set
TS_Voice1

As before, make sure the Enable EDU Tracking setting is checked and open TS Set to set the telephony server (TS) and then OK out from that window, and OK out from the Device window to create the new Queue device.

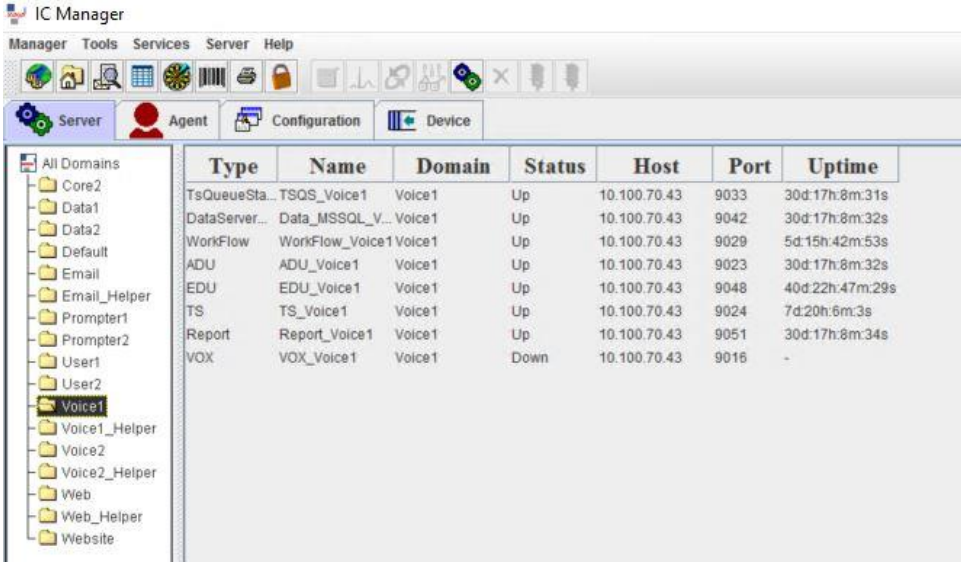
The list of Voice devices will now show listed under the Devices folder on the navigation pane, and listed on the Voice page:



Queue Id	Site	Type	Name	Priority	Service Level	Minimum Agents
40010	DefaultSite	Voice	VoiceQueue	1	00:00:30	1
40013	DefaultSite	Voice	TransfertoSkill500	1	00:00:30	1
40031	DefaultSite	Voice	Mindful Transfer VQ	1	00:00:30	1
40034	DefaultSite	Voice	CB Queue to Skill	1	00:00:30	1

6.2.3 Telephony Server Configuration (Server Tab)

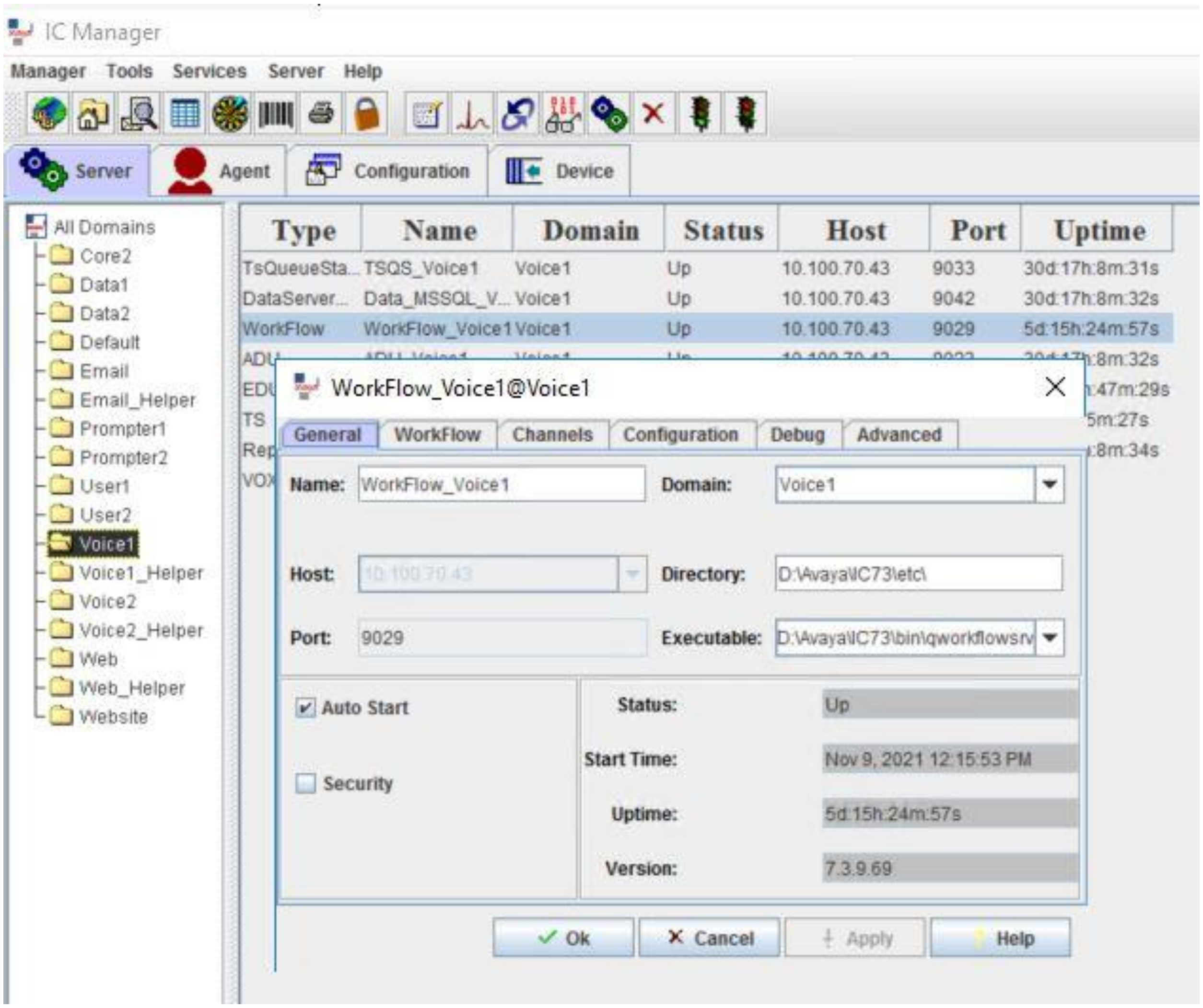
The Telephony Server (TS) should already be configured in the IC environment and comprises of a number of services that will show under the primary Voice server entry. In this example, this is configured as Voice1:



Type	Name	Domain	Status	Host	Port	Uptime
TsQueueSta...	TSQS_Voice1	Voice1	Up	10.100.70.43	9033	30d:17h:8m:31s
DataServer...	Data_MSSQL_V...	Voice1	Up	10.100.70.43	9042	30d:17h:8m:32s
WorkFlow	WorkFlow_Voice1	Voice1	Up	10.100.70.43	9029	5d:15h:42m:53s
ADU	ADU_Voice1	Voice1	Up	10.100.70.43	9023	30d:17h:8m:32s
EDU	EDU_Voice1	Voice1	Up	10.100.70.43	9048	40d:22h:47m:29s
TS	TS_Voice1	Voice1	Up	10.100.70.43	9024	7d:20h:6m:3s
Report	Report_Voice1	Voice1	Up	10.100.70.43	9051	30d:17h:8m:34s
VOX	VOX_Voice1	Voice1	Down	10.100.70.43	9016	-

The two servers that will be configured or modified here are the Telephony (TS) and Workflow servers.

First, open the workflow server (double-click on the Workflow type entry). A new Workflow window will pop up. If no existing Workflow server exists, a new one can be created using the New Server button.



Type	Name	Domain	Status	Host	Port	Uptime
TsQueueSta...	TSQS_Voice1	Voice1	Up	10.100.70.43	9033	30d:17h:8m:31s
DataServer...	Data_MSSQL_V...	Voice1	Up	10.100.70.43	9042	30d:17h:8m:32s
WorkFlow	WorkFlow_Voice1	Voice1	Up	10.100.70.43	9029	5d:15h:24m:57s
ADU						:8m:32s
EDU						:47m:29s
TS						5m:27s
Rep						:8m:34s
VOX						

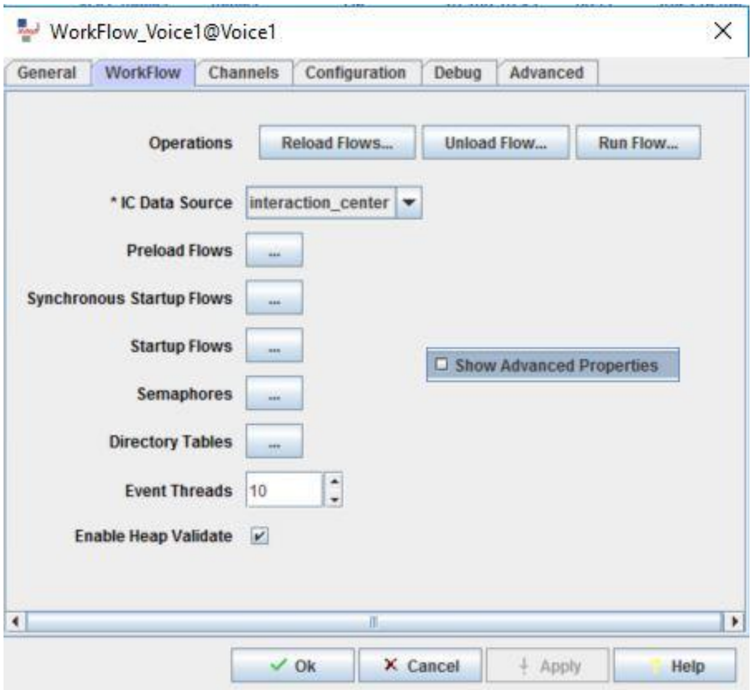
Name: WorkFlow_Voice1
Domain: Voice1
Host: 10.100.70.43
Directory: D:\Avaya\IC73\etc\
Port: 9029
Executable: D:\Avaya\IC73\bin\qworkflowsrv

☑ Auto Start

□ Security

Status: Up
Start Time: Nov 9, 2021 12:15:53 PM
Uptime: 5d:15h:24m:57s
Version: 7.3.9.69

The General tab should not require any changes if opening an existing Workflow server. Click on the Workflow tab:



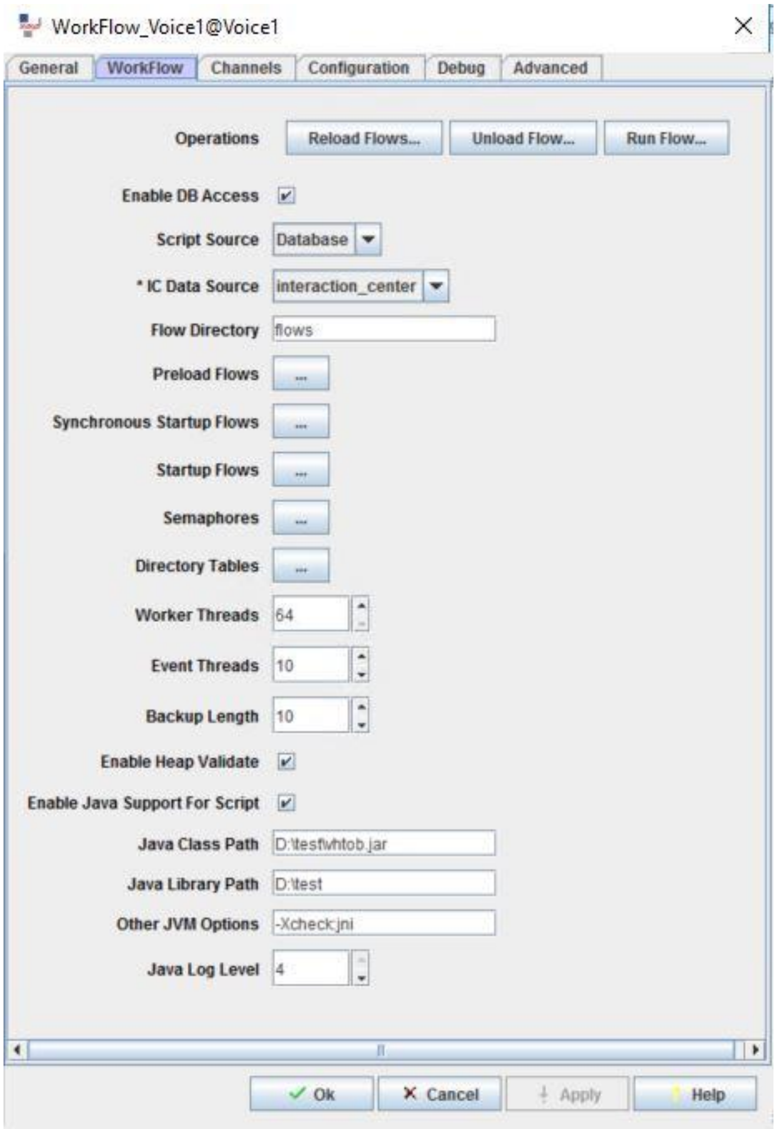
* IC Data Source: interaction_center

Event Threads: 10

Enable Heap Validate ☑

□ Show Advanced Properties

If Mindful Digital is not being used, this tab should require no additional changes and you can skip to the Channel tab configuration. However, where using Mindful Digital, the Advanced Properties of the Workflow tab need to be modified. To view Advanced Properties, click the right mouse button on this window, and a Show Advanced Properties option should appear. Check this to view the Advanced Properties:



Operations

Enable DB Access ☑

Script Source: Database

* IC Data Source: interaction_center

Flow Directory: flows

Preload Flows

Synchronous Startup Flows

Startup Flows

Semaphores

Directory Tables

Worker Threads: 64

Event Threads: 10

Backup Length: 10

Enable Heap Validate ☑

Enable Java Support For Script ☑

Java Class Path: D:\test\vhtob.jar

Java Library Path: D:\test

Other JVM Options: -Xcheck:jni

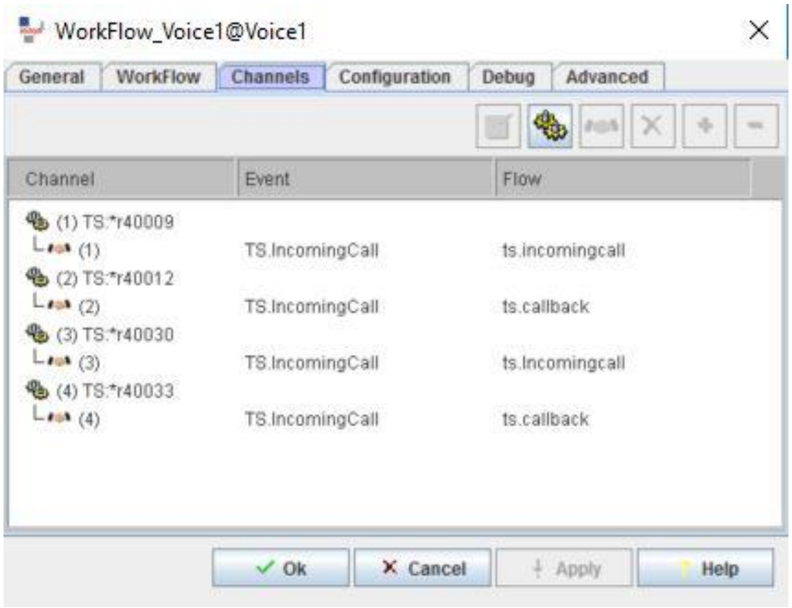
Java Log Level: 4

With the Advanced Properties now being displayed, check the Enable Java Support for Script, then populate the Java Class Path with the path and name of the vhtob.jar file that was copied onto the IC server as described in section Mindful Digital Call flow - Additional Files

Java library path should be set the same path as the vhtob.jar file and ensure that the Other JVM options and Java log level are set as per the example above.

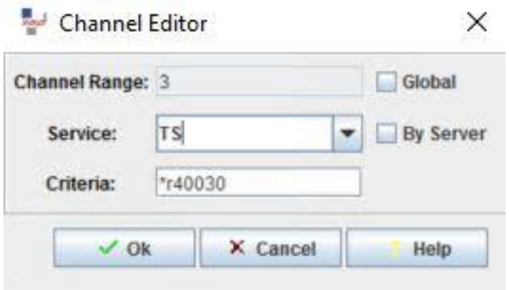
Note that if you are already using Java objects, these fields may already be populated. If this is the case, please contact the VHT Solution Delivery team (a new Workflow server may need to be created).

Now click on the channels tab:



Channel	Event	Flow
(1) TS:*r40009		
└ (1)	TS.IncomingCall	ts.incomingcall
(2) TS:*r40012		
└ (2)	TS.IncomingCall	ts.callback
(3) TS:*r40030		
└ (3)	TS.IncomingCall	ts.incomingcall
(4) TS:*r40033		
└ (4)	TS.IncomingCall	ts.callback

In this page, click on the New Channel button.



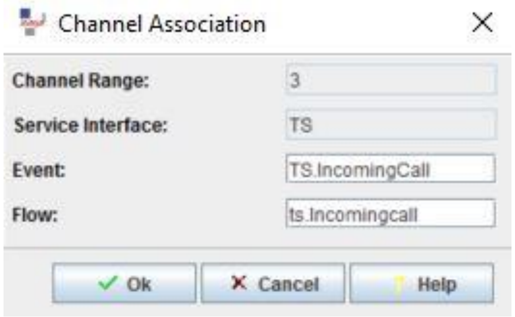
Channel Range: 3 □ Global

Service: TS □ By Server

Criteria: *r40030

In the channel editor window, select the TS service, then for criteria, use *r<vdn number> where vdn number is the VDN number of the Inbound Entry VDN – e.g. for VDN 40030, the Criteria is *r40030.

Click OK, and then whilst the newly created channel is selected, click on the New Association button.



Channel Range: 3

Service Interface: TS

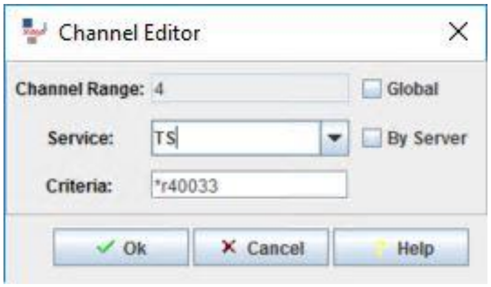
Event: TS.IncomingCall

Flow: ts.incomingcall

In the Channel Association window, the Event should be TS.IncomingCall, and the flow will be ts.incomingcall (to match the project and workflow name of the inbound workflow).

Click OK to save this new association.

Now repeat the above New Channel/New Association steps for the Callback Entry VDN:


Channel Editor

Channel Range: 4

□ Global

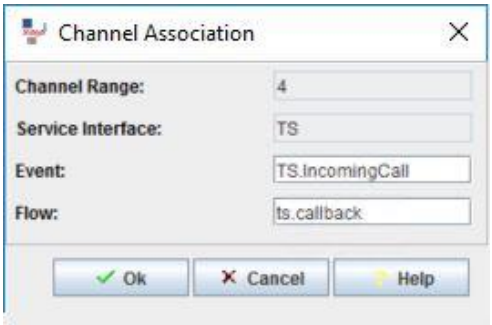
Service: TS

□ By Server

Criteria: *r40033

This time the criteria will be *r<vdn number> where vdn number is the VDN number of the Callback Entry VDN – e.g. for VDN 40033, the Criteria is *r40033

Once this new Channel is created, select that channel and click on the New Association button.


Channel Association

Channel Range: 4

Service Interface: TS

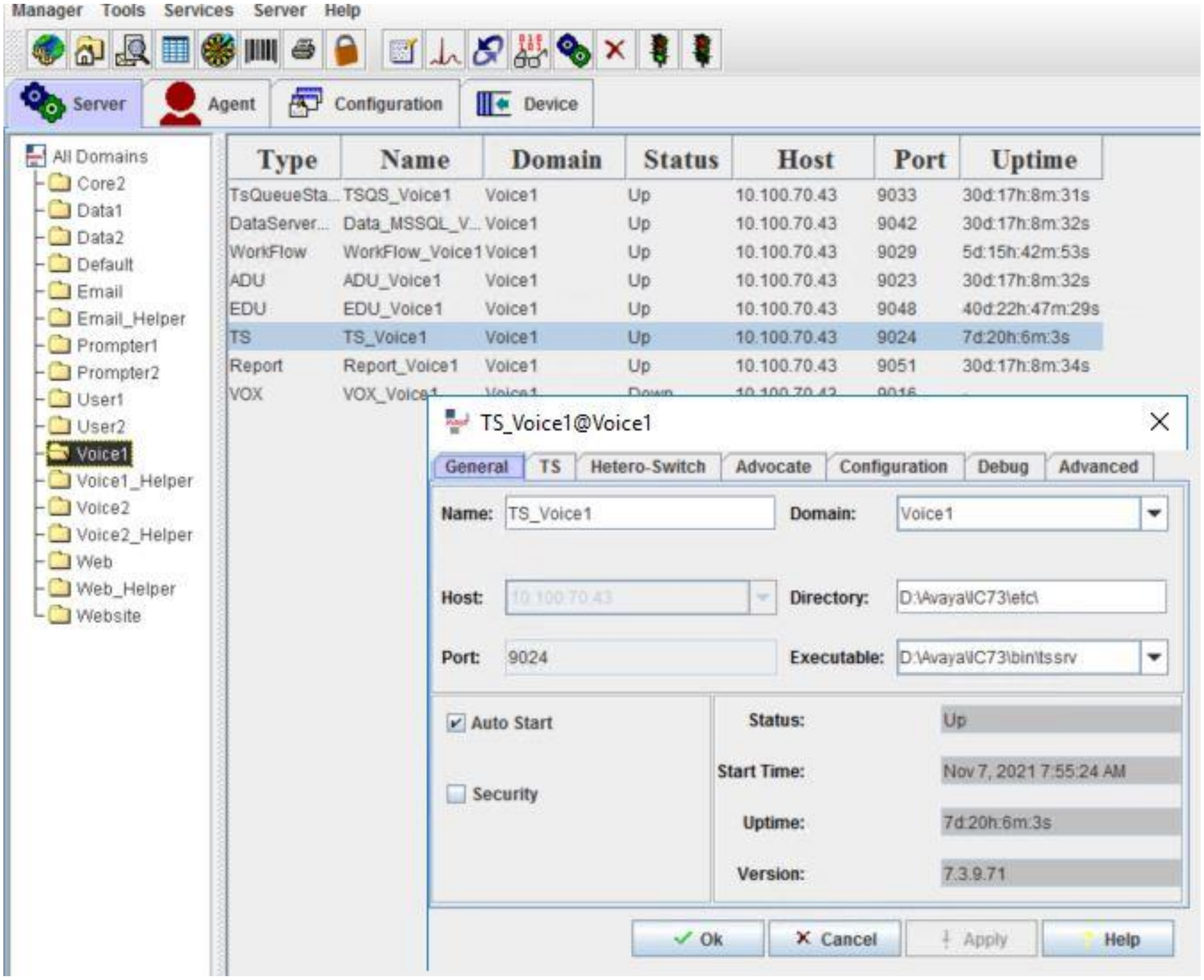
Event: TS.IncomingCall

Flow: ts.callback

In the Channel Association window, the Event should be TS.IncomingCall as before, but this time the flow will be ts.callback (to match the project and workflow name of the callback workflow).

Click OK to save this new association.

Finally, from the list of servers in the primary Voice TS folder, open the TS entry:



Type	Name	Domain	Status	Host	Port	Uptime
TsQueueSta...	TSQS_Voice1	Voice1	Up	10.100.70.43	9033	30d:17h:8m:31s
DataServer...	Data_MSSQL_V...	Voice1	Up	10.100.70.43	9042	30d:17h:8m:32s
WorkFlow	WorkFlow_Voice1	Voice1	Up	10.100.70.43	9029	5d:15h:42m:53s
ADU	ADU_Voice1	Voice1	Up	10.100.70.43	9023	30d:17h:8m:32s
EDU	EDU_Voice1	Voice1	Up	10.100.70.43	9048	40d:22h:47m:29s
TS	TS_Voice1	Voice1	Up	10.100.70.43	9024	7d:20h:6m:3s
Report	Report_Voice1	Voice1	Up	10.100.70.43	9051	30d:17h:8m:34s
VOX	VOX_Voice1	Voice1	Down	10.100.70.43	9016	-

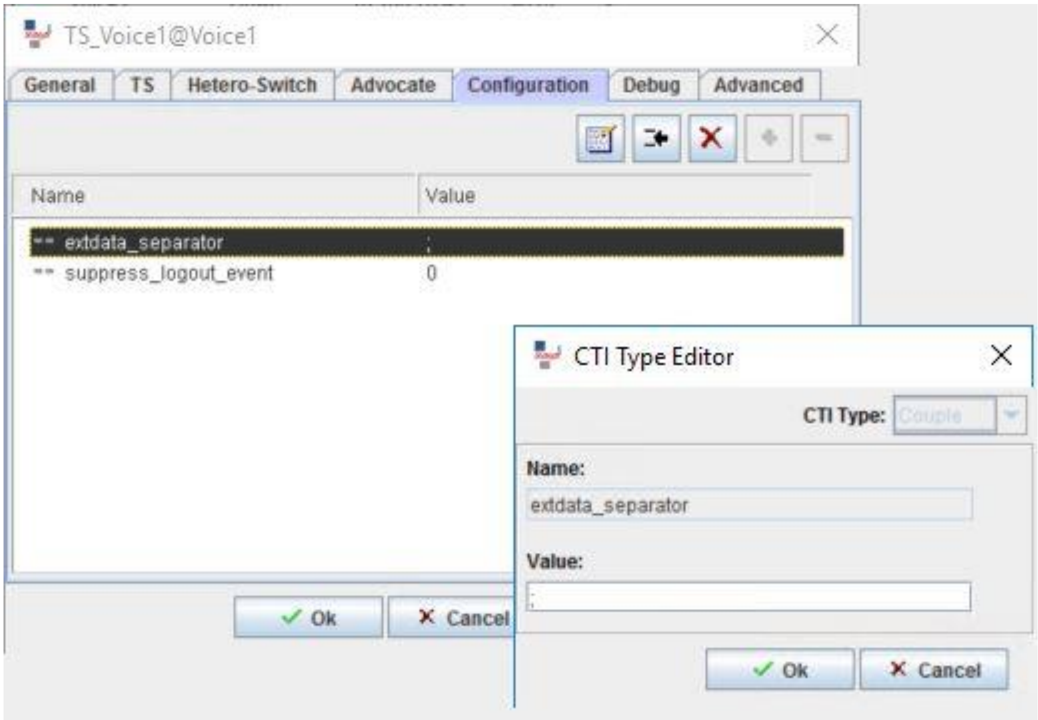
TS_Voice1@Voice1

Name: TS_Voice1
Domain: Voice1
Host: 10.100.70.43
Directory: D:\Avaya\IC73\etc\
Port: 9024
Executable: D:\Avaya\IC73\bin\tssrv

☑ Auto Start
☐ Security

Status: Up
Start Time: Nov 7, 2021 7:55:24 AM
Uptime: 7d:20h:6m:3s
Version: 7.3.9.71

The General tab should require no modification, click on the configuration tab in this TS window:



Name	Value
extdata_separator	;
suppress_logout_event	0

CTI Type: Couple

Name: extdata_separator

Value: ;

Add a new entry using the New button.

In the CTI Type Editor window, add a new entry with exdata_seperator as the name, and semi-colon (;) as the value.

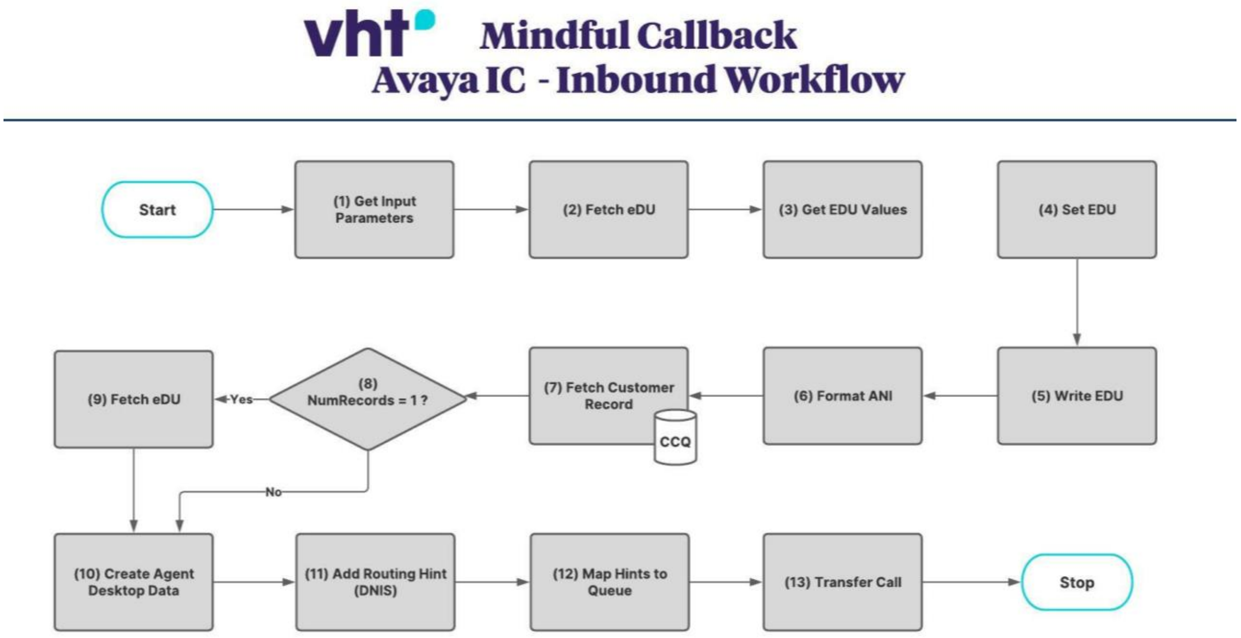
7. Avaya IC Sample Workflows

A project containing two sample workflows will be provided by the VHT Solution Delivery team. The incomingcall workflow, is loaded against the “entry” VDNs in the workflow channel configuration, and the callback workflow is loaded against the Callback VDNs in the workflow channel configuration.

Open the sample project file in IC Workflow Designer, in which two workflows are included – the Inbound call workflow (incomingcall) and the callback workflow (callback).

7.1 Inbound Workflow (incomingcall.qfd)

The Inbound workflow (incomingcall.qfd) is largely based on the standard IC incoming call workflow template, and follows the following logic:

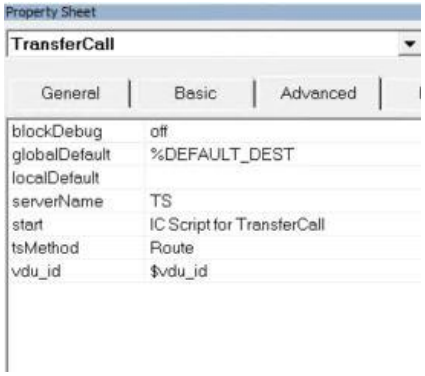


(1) Get Input Parameters – this retrieves values for the input parameters listed on the Basic tab and enters the values into script variables – i.e. ani, dnis, language, vdu_id (eduid), digits and uui.
(2) Fetch EDU – Fetches the EDU (if it exists) based on the EDU ID (variable vdu_id).
(3) Get EDU – Assigns values retrieved from the fetched EDU (if any were returned) to script variables.
(4) Set EDU – Assigns values to EDU couples for later storage in the EDU record. Sets the digits and UUI in the example script.
(5) Write EDU – writes a record to a field in the EDU – this stores the sequence of couples set by the previous Set EDU block into the EDU.

(6) Format ANI – this formats the ANI string into the appropriate number format – e.g. (000)000-0000 for North American numbers.
(7) Fetch Customer Record - Searches the Customer table of the CCQ database for a record that matches the specified value of the specified field – e.g. Phone.
(8) NumRecords = 1? – This Test Data block checks to see how many records are returned in the preceding Fetch Customer Record block. If the number of records returned equals 1, the Fetch eDU block (9) is invoked, otherwise the flow goes straight to the Create Agent Desktop (10) block.
(9) Fetch EDU - Fetches the EDU data again. The workflow can then use both the EDU data and the Customer data retrieved by the Fetch Customer Record block.
(10) Create Agent Desktop Data - Uses the EDU data and Customer data to create the data that Avaya Agent and other agent desktop applications use for screen pops. If the Fetch Customer Record block finds more than one Customer record, or no Customer records, then this block uses only the values in the EDU.
(11) Add Routing Hints - Checks the EDU for the DNIS of the voice contact. Adds the DNIS value to the $scRoutingHints variable that is used by the following Map Hints to Queue block.
(12) Map Hints to Queue - Maps the hints collected in the $scRoutingHints script variable to a routing hint in the RoutingHint table of the Directory server. Uses the routing hints to identify a queue to which the contact routes to. So this will be used in conjunction with the Routing Hint created previously in this document, that maps the Inbound Entry VDN with the Inbound Transfer to Mindful VDN.
(13) Transfer Call - Calls the Telephony server with the routing information collected by the workflow, including the value of the queue set by the Map Hints to Queue block. Please note that the script in this block is slightly modified in the sample Inbound workflow – see the Key changes section below.

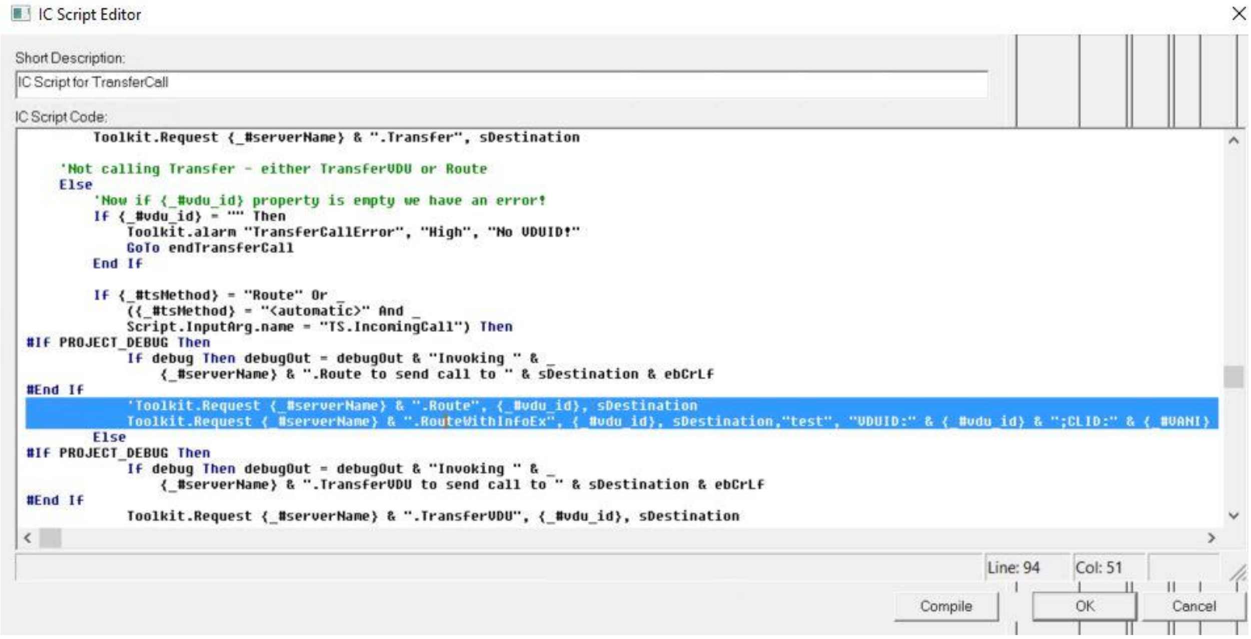
7.1.1 Key changes in the inbound workflow:

Within the TransferCall block, open the script (IC Script for TransferCall):



blockDebug	off
globalDefault	%DEFAULT_DEST
localDefault	
serverName	TS
start	IC Script for TransferCall
tsMethod	Route
vdu_id	$vdu_id

Line 94 has been modified to set the UUI (with the EDU ID).



Short Description:

IC Script for TransferCall

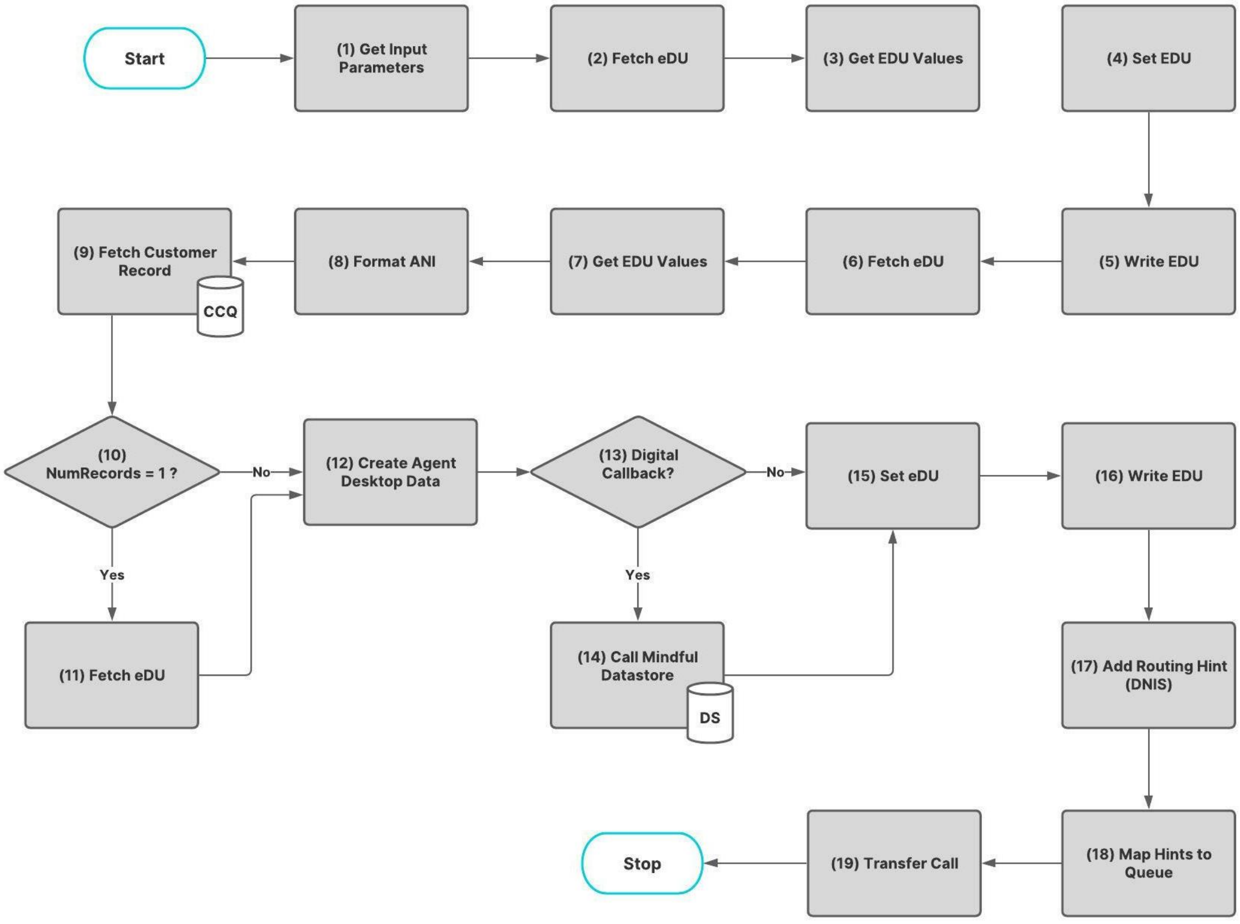
IC Script Code:

```
        Toolkit.Request {_#serverName} & ".Transfer", sDestination

    'Not calling Transfer - either TransferVDU or Route
    Else
        'Now if {_#vdu_id} property is empty we have an error!
        If {_#vdu_id} = "" Then
            Toolkit.alarm "TransferCallError", "High", "No VDUID!"
            GoTo endTransferCall
        End If

        If {_#tsMethod} = "Route" Or _
            ({_#tsMethod} = "<automatic>" And _
            Script.InputArg.name = "TS.IncomingCall") Then
#If PROJECT_DEBUG Then
            If debug Then debugOut = debugOut & "Invoking " & _
                {_#serverName} & ".Route to send call to " & sDestination & ebCrLf
#End If
            'Toolkit.Request {_#serverName} & ".Route", {_#vdu_id}, sDestination
            Toolkit.Request {_#serverName} & ".RouteWithInfoEx", {_#vdu_id}, sDestination,"test", "VDUID:" & {_#vdu_id} & ";CLID:" & {_#VANI}
        Else
#If PROJECT_DEBUG Then
            If debug Then debugOut = debugOut & "Invoking " & _
                {_#serverName} & ".TransferVDU to send call to " & sDestination & ebCrLf
#End If
            Toolkit.Request {_#serverName} & ".TransferVDU", {_#vdu_id}, sDestination
```

7.2 Callback Workflow (callback.qfd)

The Callback workflow (callback.qfd) is similar to the inbound call workflow, and follows the following logic:

Mindful Callback
Avaya IC - Callback Workflow



(1) Get Input Parameters – this retrieves values for the input parameters listed on the Basic tab and enters the values into script variables – i.e. ani, dnis, language, vdu_id (eduid), digits and uui.
(2) Fetch EDU – Fetches the EDU (if it exists) based on the EDU ID (variable vdu_id).
(3) Get EDU – Assigns values retrieved from the fetched EDU (if any were returned) to script variables – this one retrieves the ani in the sample callback workflow.
(4) Set EDU – Assigns values to EDU couples for later storage in the EDU record. Sets the digits and UUI in the example script.
(5) Write EDU – writes a record to a field in the EDU – this stores the sequence of couples set by the previous Set EDU block into the EDU.
(6) Fetch EDU – Fetches the EDU again based on the EDU ID (variable vdu_id).
(7) Get EDU – Assigns values retrieved from the fetched EDU (if any were returned) to script

(8) Format ANI – this formats the ANI string into the appropriate number format – e.g. (000)000-0000 for North American numbers.

(9) Fetch Customer Record - Searches the Customer table of the CCQ database for a record that matches the specified value of the specified field – e.g., Phone.

(10) NumRecords = 1? – This Test Data block checks to see how many records are returned in the preceding Fetch Customer Record block. If the number of records returned equals 1, the Fetch eDU block (9) is invoked, otherwise the flow goes straight to the Create Agent Desktop (10) block.

(11) Fetch EDU - Fetches the EDU data again. The workflow can then use both the EDU data and the Customer data retrieved by the Fetch Customer Record block.

(12) Create Agent Desktop Data - Uses the EDU data and Customer data to create the data that Avaya Agent and other agent desktop applications use for screen pops. If the Fetch Customer Record block finds more than one Customer record, or no Customer records, then this block uses only the values in the EDU.

(13) Digital Callback check – this Test Data block checks the UUI to see if it is empty. If it is empty (which it normally would be for a Mindful Digital callback), then we treat this as a Digital callback and flow to the Call WebServices block (14). If not, the call is treated as a callback generated by an inbound voice call to Mindful Callback (via the IC inbound workflow), and flows straight to the Set EDU block (15)

(14) Call webservices – this block invokes the vhtobpr1 java class in the vhtobj.jar file copied to the IC server and configured in the Telephony Server section in IC Manager previously in this document. This java class will perform a HTTPS GET against the Mindful Datastore using the url and auth token in the config file copied onto the IC server previously in the document, and using the call ANI as the key. The returned values are dynamically added as key-value pairs into the $newVduData variable.

(15) Set EDU – if this call is determined not to be a Digital callback, this block assigns an EDU couple marking the call as a Voice Callback.

(16) Write EDU – writes a record to a field in the EDU – this stores the sequence of couples set by the either the Call Webservices block (14), or the Set EDU block (15) into the EDU.

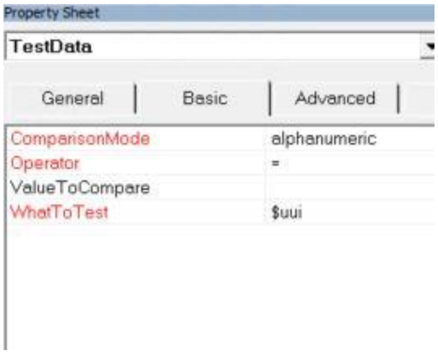
(17) Add Routing Hints - Checks the EDU for the DNIS of the voice contact. Adds the DNIS value to the $scRoutingHints variable that is used by the following Map Hints to Queue block.

(18) Map Hints to Queue - Maps the hints collected in the $scRoutingHints script variable to a routing hint in the RoutingHint table of the Directory server. Uses the routing hints to identify a queue to which the contact routes to. So this will be used in conjunction with the Routing Hint created previously in this document, that maps the Callback Entry VDN with the Callback Queue VDN.

(19) Transfer Call - Calls the Telephony server with the routing information collected by the workflow, including the value of the queue set by the Map Hints to Queue block.

7.2.1 Key changes in the callback workflow:

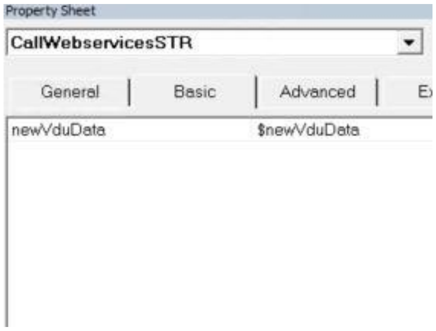
Digital Callback Check (Test Data) block:



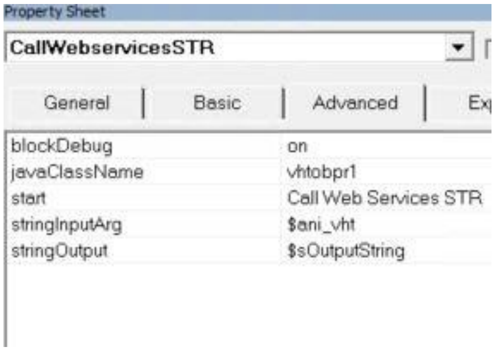
ComparisonMode	alphanumeric
Operator	=
ValueToCompare	
WhatToTest	$uui

A Test Data block does a check against the UUI, if this is blank, it is assumed that the callback is a Digital Callback (as the callbacks generated by Mindful Callback will contain no UUI). If this is not blank, it is treated as a Mindful Callback that was generated via an Inbound Call that passed through the IC Inbound workflow. If Digital Callback is not being used, this block is not required.

Fetch data from Mindful Datastore (CallWebServicesSTR) block:



newVduData	$newVduData



blockDebug	on
javaClassName	vhtobpr1
start	Call Web Services STR
stringInputArg	$ani_vht
stringOutput	$sOutputString

If the Digital Callback check (Test Data) block returns a Yes result (call identified as a Digital Callback), the CallWebServicesSTR block is used to invoke the vhtobpr1 java class (from the vhtob.jar file). The ANI for the call (the number that the customer requested a callback to) is passed in, and used to fetch the Datastore record that matches this ANI. The returned data is stored in the $newVduData variable. A subsequent WriteEDU block then writes this to the EDU record.

8. Troubleshooting

Here are some examples of troubleshooting the following scenarios using the Workflow server logs. These logs may all be found in the Avaya IC Server main logs folder – e.g. C:\Avaya\IC73\logs and the log file name will be *WorkFlow_<name of TS in IC config>.log* - in this example: *Workflow_Voice1.log*

The example log events listed below have been split into Inbound workflow log events and Callback workflow log events.

8.1Inbound Call Flow

Initiation of Voice call (register of voice callback):

```
[TS.IncomingCall.event({0,"TS.IncomingCall",{0,13,{"vdu_id","617d38c0000000000a64462b23580
002"},{"call_ref_id","14071"},{"monitor","40030"},{"call_route_id","85"},{"orig","3307771234"},{"dest
","40030"},{"ani","3307771234"},{"dnis","13307772420"},{"called","13307772420"},{"digits","40030
"},{"calltype","queue"},{"ctype","queue"},{"ucid","00001140711635596470"}}})]
```

Fetching basic vdu data

```
Text:
[VDU.GetValues.response("617d38c0000000000a64462b23580002",{0,22,{"type","voice"},{"ani",
"3307771234"},{"dnis","13307772420"},{"primary_ani","3307771234"},{"primary_dnis","133077724
20"},{"loginid","6065a4f7001b00000a64462b23450002"},{"agent_key",""},{"voice_direction","inbo
und"},{"calltype","queue"},{"ctype","queue"},{"ucid","00001140711635596470"},{"digits","40030"},{
"vdu_id","617d38c0000000000a64462b23580002"},{"owner","617d36d8000000000a64462b234
00002"},{"createtime","2021-10-30
12:21:20"},{"createtimet","1635596480"},{"phone",""},{"dest","40030"},{"orig","3307771234"},{"ext"
,"40030"},{"agent",""},{"agent.1","6065a4f7001b00000a64462b23450002"}})67108881]
```

Setting values (digits and UUI)

```
[VDU.SetValues("617d38c0000000000a64462b23580002",{0,2,{"digits","40030"},{"uui",""}})]
```

Routing hint request and response

Request

```
[DS.GetFewRecords(",","(type=\"RoutingHint\")&(hint=\"40030\")",{0,0})]
```

Response

```
[DS.GetFewRecords.response(",","(type=\"RoutingHint\")&(hint=\"13307772420\")",{0,1,{0,12,{"typ
e","RoutingHint"},{"chatqueue",""},{"creation_timestamp","2021-08-23
11:20:03"},{"customrec","1"},{"emailqueue",""},{"hint","13307772420"},{"key","811"},{"qualifier",""},{"
tenant","1"},{"update_time","2021-08-23
11:20:03"},{"version","2"},{"voicequeue","40031"}}})67108881]
```

Collecting information about queue

Request

```
[ADU.FindByKey("queueid","40031",)]
```

Response

```
[ADU.FindByKey.response("queueid","40031","617d36ce000300000a64462b233f0002")6710888
1]
```

Routing call to Mindful Transfer VDN 40031 with append of UUI Info

Request

```
[TS.RouteWithInfoEx("617d38c0000000000a64462b23580002","40031","test","VDUID:617d38c0
000000000a64462b23580002;CLID:3307771234")]
```

Response

```
[TS.RouteWithInfoEx.response("617d38c0000000000a64462b23580002","40031","test","VDUID:
617d38c0000000000a64462b23580002;CLID:3307771234")67108881]
```

8.2 Callback Workflow logs (Voice callback)

Workflow Log incoming callback from Mindful (on the Callback Entry VDN)

```
[TS.IncomingCall.event({0,"TS.IncomingCall",{0,15,{"vdu_id","617d38c0000000000a64462b23580
002"},{"call_ref_id","14072"},{"monitor","40033"},{"call_route_id","97"},{"orig","3307771234"},{"dest
","40033"},{"ani","3307771234"},{"dnis","40033"},{"called","40033"},{"digits","40030"},{"calltype","
direct"},{"ctype","direct"},{"lai_dnis","MRQA2 IC
Entry"},{"ucid","00001140711635596470"},{"extdata","VDUID:617d38c0000000000a64462b23580
002;CLID:3307771234"}}})]
```

Fetching edu data by workflow

```
[VDU.GetValues.response("617d38c0000000000a64462b23580002",{0,71,{"type","voice"},{"ani","
3307771234"},{"dnis","40033"},{"primary_ani","3307771234"},{"primary_dnis","13307772420"},{"lo
ginid","6065a4f7001b00000a64462b23450002"},{"agent_key",""},{"voice_direction","inbound"},{"c
alltype","queue"},{"ctype","queue"},{"ucid","00001140711635596470"},{"digits","40030"},{"vdu_id",
"617d38c0000000000a64462b23580002"},{"owner","617d36d8000000000a64462b23400002"},
{"createtime","2021-10-30
12:21:20"},{"createtimet","1635596480"},{"phone",""},{"dest","40033"},{"orig","3307771234"},{"ext"
,"40033"},{"agent",""},{"agent.1","6065a4f7001b00000a64462b23450002"},{"uui",""},{"customer_f
etched","true"},{"AgentDesktop",""},{"AgentDesktop.customer_name",""},{"AgentDesktop.customer
_email",""},{"AgentDesktop.contact_label","3307771234"},{"AgentDesktop.ScreenPop",""},{"AgentD
esktop.ScreenPop.type","voice"},{"AgentDesktop.ScreenPop.value","=(561) 316-
7841"},{"AgentDesktop.ScreenPop.CreateContact","True"},{"contactduration","36"},{"contactendti
me","1635596516"},{"last_termination","617d36c4000b00000a64462b23400002"},{"duration","36
"},{"endtime","2021-10-30 12:21:56"},{"lai_dnis","MRQA2 IC
Entry"},{"extdata","VDUID:617d38c0000000000a64462b23580002;CLID:3307771234"},{"agent.2"
,"6065a4f8002300000a64462b23490002"},{"voice",""},{"voice.1","36"},{"voice.1.loginid","6065a4f
8002300000a64462b23490002"},{"voice.1.leg_id","617d38e4000500000a64462b23400002"},{"v
oice.1.destination","40031"},{"voice.1.origin","3307771234"},{"voice.1.direction","inbound"},{"voice.
1.acdname","7e2"},{"voice.1.owner","6065a4f5003e00000a64462b23400002"},{"voice.1.alias","ts_
voice1"},{"voice.1.ucid","00001140711635596470"},{"voice.1.stdstate",""},{"voice.1.stdstate.1","crea
ted"},{"voice.1.stdstate.1.created",""},{"voice.1.stdstate.1.created.reason",""},{"voice.1.stdstate.1.cr
eated.starttime","1635596516"},{"voice.1.stdstate.2","alerting"},{"voice.1.stdstate.2.alerting",""},{"v
oice.1.stdstate.2.alerting.reason",""},{"voice.1.stdstate.2.alerting.starttime","1635596516"},{"voice.1
.queue_key","7"},{"voice.1.queuetime","36"},{"voice.1.exit_reason","abandon"},{"voice.1.abandoned"
,"true"},{"voice.1.abandon","while in
queue"},{"voice.1.abandontime","36"},{"voice.1.stdstate.3","terminated"},{"voice.1.stdstate.3.termin
ated",""},{"voice.1.stdstate.3.terminated.reason","100"},{"voice.1.stdstate.3.terminated.starttime","1
635596516"},{"agent.3","6065a4f7001b00000a64462b23450002"}})67108881]
```

Setting EDU values for screenpop data for potential agent

Following data

```
Text: ======== Debug: SetVdu (ts.incomingcallmf.SetEDU1) ========
 Setting 6 couples
 digits:              40030
 uui:                 VDUID:617d38c0000000000a64462b23580002;CLID:3307771234
 ani:                 3307771234
 orig:                3307771234
 vdu_idold:           617d38c0000000000a64462b23580002
 animfsp:             3307771234
```

Request

```
[VDU.SetValues("617d38c0000000000a64462b23580002",{0,6,{"digits","40030"},{"uui","VDUID:6
17d38c0000000000a64462b23580002;CLID:3307771234"},{"ani","3307771234"},{"orig","330777
1234"},{"vdu_idold","617d38c0000000000a64462b23580002"},{"animfsp","3307771234"}})]
```

Response

```
[VDU.SetValues.response("617d38c0000000000a64462b23580002",{0,6,{"digits","40030"},{"uui"
,"VDUID:617d38c0000000000a64462b23580002;CLID:3307771234"},{"ani","3307771234"},{"orig"
,"3307771234"},{"vdu_idold","617d38c0000000000a64462b23580002"},{"animfsp","3307771234"
}})67108881]
```

Segregating between voice callback or digital callback in block log message

```
Text: ======== Debug: TestData (ts.incomingcallmf.TestData) ========
 Testing expression: VDUID:617d38c0000000000a64462b23580002;CLID:3307771234 =
 Result: No
```

```
 Text: ======== Debug: Set EDU (ts.incomingcallmf.SetEDU2) ========
 Setting 1 couple
 typeofcall:             Voice Callback
```

Setting value in EDU for Agent screen pop display

```
[VDU.SetValues("617d38c0000000000a64462b23580002",{0,1,{"typeofcall","Voice
Callback"}})]
```

Fetching transfer to hunt group as mentioned above in routinghint section

Request

```
 Text: [DS.GetFewRecords(",","(type=\"RoutingHint\")&(hint=\"40033\")",{0,0})]
```

Response

```
[DS.GetFewRecords.response(",","(type=\"RoutingHint\")&(hint=\"40033\")",{0,1,{0,12,{"ty
pe","RoutingHint"},{"chatqueue",""},{"creation_timestamp","2021-08-24
11:03:47"},{"customrec","1"},{"emailqueue",""},{"hint","40033"},{"key","812"},{"qualifier",""},
{"tenant","1"},{"update_time","2021-09-23
13:18:03"},{"version","2"},{"voicequeue","40034"}}})67108881]
```

Route call to final destination (Callback Queue VDN) 40034

Request

```
 Text: [TS.Route("617d38c0000000000a64462b23580002","40034")]
```

Response

```
Text: [TS.Route.response("617d38c0000000000a64462b23580002","40034")67108881]
```

8.3. Callback Workflow logs (Digital callback).

For Java API logs (Digital Callback only) – this is the fetch response to the VHT Datastore CallWebService request in the Callback Workflow – you can see the data returned by the datastore in the bold section of text:

```
@20210928 18:57:12.082 #1416   <WorkflowJavaInterface>
[x:\source\server\qworkflow\javaserver.cpp@2057]
 (CQWFJavaServer::InvokeJMethod) : "String invokeICJava(String)" using classname call -
SUCCESS. ClassName [vhtob], Argument [3307771234]
@20210928 18:57:12.082 #1416   <WorkflowJavaInterface>
[x:\source\server\qworkflow\javaserver.cpp@2057]
 (CQWFJavaServer::releaseJNIEnvCurrentThread) : Start.
@20210928 18:57:12.082 #1416   <WorkflowJavaInterface>
[x:\source\server\qworkflow\javaserver.cpp@2057]
 (CQWFJavaServer::releaseJNIEnvCurrentThread) : End.
@20210928 18:57:12.082 #1416   <WorkflowJavaInterface>
[x:\source\server\qworkflow\javaserver.cpp@2057]
 (CQWFJavaServer::InvokeJMethod) : End.
@20210928 18:57:12.082 #1416   <Print> [x:\source\server\qworkflow\scriptq.cpp@1138]
 Text:
{"customer_contact_number":"3307771234","ani":"3307771234","ecbt":null,"time_to_expire":"2
021-09-
29T01:32:13.238Z","FirstName":"Jane","LastName":"Doe","AccNum":"123654","Company":"VH
T"}
```