



Courtesy Callback

- [Capabilities](#) , on page 1
- [Initial Setup](#) , on page 3
- [Administration and Usage](#), on page 18

Capabilities

Courtesy Callback reduces the time callers have to physically wait on hold or in a queue. The feature enables your system to offer callers (who meet your criteria) the option to receive a courtesy callback by the system instead of waiting on the phone for an agent. The caller who has been queued by Unified CVP can hang up and subsequently be called back when an agent is close to becoming available (preemptive callback).

Preemptive callback does not change the time a customer must wait to be connected to an agent, but rather enables the caller to hang up and not be required to remain in queue listening to music. Callers who have remained in queue or have undergone the callback treatment appears the same to agents answering the call.

If the caller decides to be called back by the system, they leave their name and phone number. Their request remains in the system and when the system determines that an agent will be available soon (or is available), then the system places a call back to the caller. The caller answers the call and confirms that they are the original caller and the system connects the caller to the agent after a brief wait.

In the event that the caller cannot be reached after a configurable max number and frequency of retries, the callback is aborted and the database status is updated appropriately. You can run reports to determine if any manual callbacks are necessary based on your business rules.

Note that you cannot schedule a callback for a specific time.



Note There are a number of prerequisites and design considerations for using this feature. See the Cisco Unified Customer Voice Portal Release Solution Reference Network Design (SRND) guide.



Note The Cisco Unified Customer Voice Portal Release Solution Reference Network Design (SRND) guide also describes how the system determines customer wait time and when to call the customer for the callback.

Callback Criteria

In your callback script, you can establish criteria for offering a caller a courtesy callback. Examples of callback criteria include:

- Number of minutes a customer is expected to be waiting *in queue* that exceeds a maximum number of minutes (based on your average call handling time per customer)



Note The included example scripts use this method for determining callback eligibility.

- Assigned status of a customer (*gold* customers may be offered the opportunity to be called back instead of remaining on the line)
- The service a customer has requested (sales calls, or system upgrades, for example, may be established as callback criteria)

Sample Scripts and Audio Files for Courtesy Callback

The courtesy callback feature is implemented using Unified CCE scripts. The installation provides a set of modifiable example CCE scripts, call studio scripts, and audio files to get you started. You can use these scripts in your implementation after making a few required changes.

Related Topics

[CCE Script for Courtesy Callback](#), on page 13

Typical Use Scenario

If the caller decides to be called back by the system, they leave their name and phone number. Their request remains in the system and the EWT fires when the system places a callback to the caller. The caller answers the call and confirms that they are the original caller, and the system connects the caller to the agent after a short wait.



Note Courtesy Callback is supported for IP originated calls as well.

A typical use of the Courtesy Callback feature follows this pattern:

1. The caller arrives at Unified CVP and the call is treated in the IVR environment.
 2. The Call Studio and Packaged CCE Courtesy Callback scripts determine if the caller is eligible for a callback based on the rules of your organization (such as in the prior list of conditions).
 3. If a courtesy callback can be offered, the system tells the caller the approximate wait time and offers to call the customer back when an agent is available.
 4. If the caller chooses not to use the callback feature, queuing continues as usual.
- Otherwise, the call continues as indicated in the remaining steps.

5. If the caller chooses to receive a callback, the system prompts the caller to record their name and to key in their phone number.
6. The system writes a database record to log the callback information.



Note If the database is not accessible, then the caller is not offered a callback and they are placed in queue.

7. The caller is disconnected from the TDM side of the call. However, the IP side of the call in Unified CVP and Packaged CCEs is still active. This keeps the call in the same queue position. No queue music is played, so Voice XML gateway resources used during this time are less than if the caller had actually been in queue.
8. When an agent in the service/skill category the caller is waiting for is close to being available (as determined by your callback scripts), then the system calls the person back. The recorded name is announced when the callback is made to insure the correct person accepts the call.
9. The system asks the caller, through an IVR session, to confirm that they are the person who was waiting for the call and that they are ready for the callback.

If the system cannot reach the callback number provided by the caller (for example, the line is busy, RNA, network problems, etc.) or if the caller do not confirm they are the caller, then the call is not sent to an agent. The agent is always guaranteed that someone is there waiting when they take the call. The system assumes that the caller is already on the line by the time the agent gets the call.

This feature is called preemptive callback as the system assumes that the caller is already on the line by the time the agent gets the call and that the caller has to wait minimal time in queue before speaking to an agent.

10. The system presents the call context on the agent screen-pop, as usual.

In the event that the caller cannot be reached after a configurable maximum number and frequency of retries, the callback is stopped and the database status is updated appropriately. You can run reports to determine if any manual callbacks are necessary based on your business rules.

See the *Configuration and Administration Guide for Cisco Unified Customer Voice Portal* http://www.cisco.com/en/US/products/sw/custcosw/ps1006/products_installation_and_configuration_guides_list.html guide which provides a call flow description of the function of the scripts that provide the Courtesy Callback feature.

Initial Setup

The Courtesy Callback feature must be configured on the following servers/gateways:

- Ingress Gateway (IOS configuration)
- VXML Gateway (IOS configuration)
- Virtualized Voice Browser (no special configuration is required if you use VVB instead of VXML Gateway)
- Reporting Server (through the PCCE Administration Tool)
- Media Server (upload of Courtesy Callback media files)

- Unified CVP VXML Server (upload of Call Studio Scripts)
- Packaged CCE

Courtesy Callback Design Considerations

The following design considerations apply for Courtesy Callback feature:

- During Courtesy Callback, callback is made using the same Ingress Gateway through which the call arrived.



Note In Courtesy Callback, outbound calls cannot be made using any other Egress Gateway.

- Calls that allow Callback must be queued using a Unified CVP VXML Server.
- The Unified CVP Reporting Server is a prerequisite for Courtesy Callback.
- Answering machine detection is not available for this feature. During the callback, the best that can be done is to prompt the caller with a brief IVR session and acknowledge with DTMF that they are ready to take the call.
- Calls that are transferred to agents using DTMF *8, TBCT, or hookflash cannot use the Courtesy Callback feature.
- Callbacks are a best-effort function. After a limited number of attempts to reach a caller during a callback, the callback is terminated and marked as failed.
- Customers must configure the allowed or blocked numbers that Callback is allowed to place calls through the Unified CVP Operations Console.
- Media inactivity detection feature on the VXML Gateway can impact waiting callback calls. For more information, see the *Configuration Guide for Cisco Unified Customer Voice Portal (CVP)*.
- Courtesy Callback requires an accurate EWT calculation for its optimal behavior.

Consider the following recommendations to optimize the EWT, when using Precision Queues for Courtesy Callback :

- Queue the calls to a single Precision Queue
- Do not include a `Consider If` expression when you configure a step.
- Do not include a wait time between steps or use only one step in the Precision Queue.

Configure the Ingress Gateway for Courtesy Callback

The ingress gateway where the call arrives is the gateway that processes the preemptive callback for the call, if the caller elects to receive a callback.



Note A sip-profile configuration is needed on ISR for the courtesy callback feature, only when deploying an IOS-XE version affected by CSCts00930. For more information on the defect, access the Bug Search Tool at <https://sso.cisco.com/autho/forms/CDClogin.html>.

For more information about sip-profile configuration, see *Design Guide for Cisco Unified Customer Voice Portal*, at <https://www.cisco.com/c/en/us/support/customer-collaboration/unified-customer-voice-portal/products-implementation-design-guides-list.html>.

- Step 1** Download the **survivability.tcl** file from DevNet: <https://developer.cisco.com/site/customer-voice-portal/downloads/courtesy-callback-scripts/>
- Step 2** Copy the survivability.tcl file to the flash memory of the Ingress gateway.
- Step 3** Log in to the ingress gateway.
- Step 4** If survivability is not already configured, configure it as described in the "Call Survivability" section of the *Configuration Guide for Cisco Unified Customer Voice Portal* at <https://www.cisco.com/c/en/us/support/customer-collaboration/unified-customer-voice-portal/products-installation-and-configuration-guides-list.html>.
- Step 5** To add services to the gateway, you must be in enabled-config application mode. Type these commands at the gateway console:

```
GW81#en
GW81#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
GW81(config)#application
GW81(config-app)#
```

- Step 6** Add the following to the survivability service:

```
param ccb id:<host name or ip of this gateway>;loc:<location name>;trunks:<number of callback trunks>
```

Where the definitions of the preceding fields are:

- *id*: A unique identifier for this gateway and is logged to the database to show which gateway processed the original callback request.
- *loc*: An arbitrary location name specifying the location of this gateway.
- *trunks*: The number of DSO's reserved for callbacks on this gateway. Limit the number of T1/E1 trunks to enable the system to limit the resources allowed for callbacks.

The following example shows a basic configuration:

```
service cvp-survivability flash:survivability.tcl
param ccb id:10.86.132.177;loc:doclab;trunks:1
!
```

If you are updating the survivability service, or if this is the first time you created the survivability service, remember to load the application using the command:

```
call application voice load cvp-survivability
```

- Step 7** Create the incoming dial peer, or verify that the survivability service is being used on your incoming dial peer. For example:

```
dial-peer voice 978555 pots
service cvp-survivability
incoming called-number 9785551234
direct-inward-dial
!
```

Note: We support both POTS and VoIP dial peers that point to a service provider.

Step 8 Create outgoing dial peers for the callbacks. These dial peers place the actual callback out to the PSTN. For example:

```
dial-peer voice 978554 pots
destination-pattern 978554...
no digit-strip
port 0/0/1:23
!
```

Step 9 Use the following configuration to ensure that SIP is set up to forward SIP INFO messaging:

```
voice service voip
signaling forward unconditional
```

Step 10 Save your changes.

Configure the VXML Gateway for Courtesy Callback

To configure the VXML gateway for Courtesy Callback:

Step 1 Download the `cvp_ccb_vxml.tcl` file from DevNet: <https://developer.cisco.com/site/customer-voice-portal/downloads/courtesy-callback-scripts/>

Step 2 Copy the `cvp_ccb_vxml.tcl` to the flash memory of the VXML gateway.

Step 3 To add services to the gateway, you must be in enabled-config application mode. Type these commands at the gateway console:

```
GW81#en
GW81#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
GW81(config)#application
GW81(config-app)#
```

Step 4 Add the `cvp_cc` service to the configuration:

```
service cvp_cc flash:cvp_ccb_vxml.tcl
```

The service does not require any parameters.

Load the application with the command:

```
call application voice load cvp_cc
```

Note The media-inactivity detection feature must be turned off in the VXML Gateway to successfully call back the caller. With media-inactivity enabled on the VXML Gateway, the `cvp_cc` service will disconnect the waiting callback calls after 'ip rtp report interval' * 1000-milliseconds interval. This configuration becomes important in a colocated Ingress/VXML setup where media inactivity timers are always enabled. In such scenarios, the 'ip rtp report interval' must be increased to support the maximum allowable waiting for a callback call as defined by the solution requirements.

Step 5 On the VoIP dial-peer that defines the VRU leg from Packaged CCE, verify that the codec can be used for recording. The following example shows that g711ulaw can be used for recording in Courtesy Callback:

```
dial-peer voice 123 voip
service bootstrap
incoming called-number 123T
dtmf-relay rtp-nte
codec g711ulaw
no vad
!
```

In other words, this example shows the g711ulaw codec set on the 123 voip dial-peer. The codec must be specified explicitly. A codec class cannot be used because recording will not work.

Step 6 Use the following configuration to ensure that SIP is set up to forward SIP INFO messaging:

```
voice service voip
signaling forward unconditional
```

Step 7 VXML 2.0 is required to play the beep to prompt the caller to record their name in the BillingQueue example script. Add the following text to the configuration so the VXML Server uses VXML 2.0:

```
vxml version 2.0
```

Note Whenever vxml version 2.0 is enabled on the gateway, vxml audioerror is off by default. When an audio file cannot be played, `error.badfetch` will *not* generate an audio error event. To have the gateway generate an `error.badfetch` event when a file cannot be played, enable vxml audioerror in your gateway configuration. The following example uses config terminal mode to add both commands:

```
config t
vxml version 2.0
vxml audioerror
exit
```

Step 8 Save your changes.

Configure the Reporting Server for Courtesy Callback

Before you begin

A Reporting Server is required for the Courtesy Callback feature. The Reporting Server must be installed prior to completing the following task. For instructions, see the *Cisco Packaged Contact Center Enterprise*

Installation and Upgrade Guide at <https://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-installation-guides-list.html>.

- Step 1** Log in to PCCE Administration Tool.
- Step 2** In **Unified CCE Administration**, choose **Overview > Features > Courtesy Callback**.
- Step 3** From the **Site** drop-down list, choose a site for which you want configure the Courtesy Callback feature. By default, it is 'Main'.
- Step 4** From the **CVP Reporting Server** drop-down list, choose a Reporting Server to use for storing Courtesy Callback data.
 - Note** The list includes all the Reporting Servers configured for the site.
 - If you leave the selection blank, no Reporting Server is associated with the Courtesy Callback deployment.
- Step 5** In the **Dialed Number Configuration** section, complete the following:

Fields	Required?	Description
Maximum Callbacks per Dialed Number	Yes	By default, the Unlimited option is selected, which is equivalent to an unlimited number of callbacks offered per calling number. The maximum value is 1000. To limit the number of calls, from the same calling number that are eligible to receive a callback: <ul style="list-style-type: none"> a. Select the Limited option. b. Enter a positive number in the text field to allow Courtesy Callback to validate and allow the specified number of callbacks per calling number.
Allow unmatched Dialed Numbers	Yes	Check the Allow unmatched Dialed Numbers check box to allow callbacks to the dialed numbers that are not available in the Allowed Dialed Number Patterns list. Note If no dialed numbers are present in the Allowed Dialed Number Patterns list, then Courtesy Callback does not allow any callbacks.
Allowed Dialed Number Patterns	No	The list of allowed dialed numbers to which callbacks can be sent. By default, the list includes preconfigured allowed dialed number patterns. To add a dialed number pattern: <ul style="list-style-type: none"> a. Click the '+' icon and enter a dialed number pattern. b. Click Add. To remove a dialed number pattern, click the 'x' icon associated with the number in the list.

Fields	Required?	Description
Denied Dialed Number Patterns	No	<p>The list of denied dialed numbers to which callbacks are never sent.</p> <p>By default, the list includes preconfigured denied dialed number patterns.</p> <p>To add a dialed number pattern:</p> <ol style="list-style-type: none"> a. Click the '+' icon and enter a dialed number pattern. b. Click Add. <p>To remove a dialed number pattern, click the 'x' icon associated with the number in the list.</p> <p>Denied numbers takes precedence over allowed numbers.</p> <ul style="list-style-type: none"> • Wildcarded DN patterns can contain "." and "X" in any position to match a single wildcard character • Any of the wildcard characters in the set ">!*T" will match multiple characters but can only be used trailing values because they will always match all remaining characters in the string • The highest precedence of pattern matching is an exact match, followed by the most specific wildcard match. • When the number of characters are matched equally by wildcarded patterns in both the Allowed Dialed Numbers and Denied Dialed Numbers lists, precedence is given to the one in the Denied Dialed Numbers list.

Step 6 Click **Save**.

Configure the Media Server for Courtesy Callback

Several Courtesy-Callback-specific media files are included with the sample scripts for Courtesy Callback.

You can download the Courtesy Callback specific sample media files and scripts (CCBAudioFiles.zip) from DevNet **Customer Voice Portal (CVP) > Downloads > Courtesy Callback Sample Scripts** at <https://developer.cisco.com/site/customer-voice-portal/downloads/courtesy-callback-scripts/>

The special audio files should be unzipped and copied to your media server.

CCBAudioFiles.zip has callback-specific application media files under
 C:\inetpub\wwwroot\en-us\app and media files for Say It Smart under
 C:\inetpub\wwwroot\en-us\sys.



Note If you selected the Media File installation option, during the Unified CVP install, the audio files were unzipped and copied to C:\inetpub\wwwroot\en-us\app on the installation server.



Note CCBAudioFiles.zip also contains media files for Say It Smart. During installation, these files are copied to C:\inetpub\wwwroot\en-us\sys. Copy these files to your media server, if you do not have them there already.



Note The sample scripts are set up to use the default location of `http://<server>:<port>/en-us/app` for the audio files. Later in this configuration process you will change the <server> and <port> parameters in the default location of the audio files in the example scripts to be your media server IP address and port number.

Configure Call Studio Scripts for Courtesy Callback

The Courtesy Callback feature is controlled by a combination of Call Studio scripts and ICM scripts. Refer to the *Solution Design Guide for Cisco Unified Contact Center Enterprise* (formerly the *Cisco Unified Customer Voice Portal Solution Reference Network Design*) at <https://www.cisco.com/c/en/us/support/customer-collaboration/unified-customer-voice-portal/products-implementation-design-guides-list.html> for a discussion of the script logic.

To configure the Call Studio scripts, perform the following procedure:



Note This example follows the BillingQueue example application.

Step 1

Extract the example Call Studio Courtesy Callback scripts contained in CourtesyCallbackStudioScripts.zip to a folder of your choice on the computer running Call Studio.

You can access the .zip file from the following two locations:

- From the Unified CVP install media in \CVP\Downloads and Samples\Studio Samples\CourtesyCallbackStudioScripts
- From DevNet (**Customer Voice Portal (CVP) > Downloads > Courtesy Callback Sample Scripts**) at <https://developer.cisco.com/site/customer-voice-portal/downloads/courtesy-callback-scripts/>

Step 2

Each folder contains a Call Studio project having the same name as the folder. The five individual projects comprise the Courtesy Callback feature.

Do *not* modify the following scripts:

- CallbackEngine: Keeps the VoIP leg of the call alive when the caller elects to receive the callback (and ends the call) and when the caller actually receives the callback. Do **not** modify this script.
- CallbackQueue: Handles the keepalive mechanism for the call when callers are in queue and listening to the music played by BillingQueue.

Modify the following scripts to suit your business needs:

- BillingQueue: Determines the queue music played to callers.

- **CallbackEntry:** Modify the initial IVR treatment a caller receives when entering the system and is presented with an opportunity for a callback.
- **CallbackWait:** Modify the IVR treatment a caller receives when they respond to the callback.

Note Do not change the CCB application names.

Step 3 Start Call Studio by selecting **Start > Programs > Cisco > Cisco Unified Call Studio**.

Step 4 In Call Studio, select **File > Import**.

Step 5 In the Import dialog box, expand the Call Studio folder and select **Existing Call Studio Project Into Workspace**.

Step 6 Click **Next**.

Step 7 In the Import Call Studio Project From File System dialog, browse to the location where you extracted the call studio projects. For each of the folders that were unzipped, select the folder (for example BillingQueue) and click **Finish**.

The project is imported into Call Studio. Repeat this action for each of the five folders.

When you are finished importing the five folders, you should see five projects in the *Navigator* window in the upper left.

Step 8 Update the Default Audio Path URI field in Call Studio to contain the IP address and port value for your media server.

For each of the Call Studio projects previously unzipped, complete the following steps:

- a) Select the project in the Navigator window of Call Studio.
- b) Click **Project > Properties > Call Studio > Audio Settings**.
- c) On the Audio Settings window, modify the Default Audio Path URI field by supplying your server IP address and port number for the *<Server>* and *<Port>* placeholders.
- d) Click **Apply**, and then click **OK**.

Step 9 Billing Queue Project: If desired, change the music played to the caller while on hold.

You can also create multiple instances of this project if you want to have different hold music for different clients, for example, BillingQueue with music for people waiting for billing, and SalesQueue with music for people waiting for sales. You also need to point to the proper version (BillingQueue or SalesQueue) in the ICM script. In the ICM script, the parameter `queueapp=BillingQueue` would also have a counterpart, `queueapp=SalesQueue`.

The CallbackEntry Project (in the following step) contains a node called SetQueueDefaults. This node contains the value Keepalive Interval which must be *greater* than the length of the queue music you use. Refer to the Keepalive Interval in the next step for details.

Step 10 Callback Entry Project: If desired, in the CallbackEntry project, modify the caller interaction settings in the SetQueueDefaults node.

This step defines values for the default queue. You can insert multiple SetQueueDefaults elements here for each queue name, if it is necessary to customize configuration values for a particular queue. If you do not have a SetQueueDefaults element for a given queue, the configuration values in the default queue are used.

Note You can define a `Callback_Set_Queue_Defaults` node with **Queue Name** parameter set to default. Configuration defined in this default node will be picked whenever a queue type is encountered for which there are no explicitly defined values.

- a) In the Call Studio Navigator panel, open the CallBackEntry project and double click **app.callflow** to display the application elements in the script window.
- b) Open the Start of Call page of the script using the tab at the bottom of the script display window.
- c) Select the SetQueueDefaults node.

- d) In the **Element Configuration panel**, select the Setting tab and modify the following default settings as desired:
 For the SetQueueDefaults element, the caller interaction values in the Start of Call and the Wants Callback elements, may be edited.

Step 11 Perform the following steps.

- a. Set the path for the storage of recorded caller names.
- b. Select app.callflow.
- c. In the CallbackEntry project, on the Wants Callback page, highlight the Record Name node and click the **Settings** tab in the Element Configuration window of Call Studio.
- d. In the Path setting, change the path to the location where you want to store the recorded names of the callers.

By default, Call Studio saves the path string in your VXML Server audio folder. If you are using the default path, you can create a new folder called recordings in the %CVP_HOME%\VXMLServer\Tomcat\webapps\CVP\audio\ folder on the VXML Server. If you are using IIS as your media server, create a new folder called recordings under C:\Inetpub\wwwroot\en-us\app and set that as the path for recordings.

Step 12 Set the name of the Record name file.

From the CallbackEntry project on the Wants Callback page, highlight the **Add Callback to DB** node and select the **Settings** tab in the Element Configuration window of Call Studio.

Change the **Recorded name file** setting to match the location of the recording folder you created.

This setting references the URL of the recordings folder, whereas the Path setting references the file system path.

The AddCallback element setting in the CallbackEntry project is configured to do automatic recorded file deletions. If automatic recorded file deletion is not desired, then remove the value of the Recorded name path setting in the AddCallback element. This removal action assumes that you will be doing the deletion or management of the recorded file yourself.

Step 13 In the CallbackEntry project on the Callback_Set_Queue_Defaults node, be sure the keepalive value (in seconds) is greater than the length of the queue music being played. The default is 120 seconds.

Step 14 Save the **CallbackEntry** project.

Step 15 CallbackWait Project: Modifying values in the CallbackWait application.

In this application, you can change the IVR interaction that the caller receives at the time of the actual callback. The caller interaction elements in **CallbackWait > AskIfCallerReady (page)** may be modified. Save the project after you modify it. The WaitLoop retry count can also be modified from the default of six retries in the Check Retry element. This will allow a larger window of time to pass before the call is dropped from the application. It is used in a failure scenario when the CallbackServlet on the reporting server cannot be reached. For instance, in a reboot or a service restart, this allows more time for the reporting server to reload the entry from the database when it is initializing. If the reporting server is not online within the retry window, then the entry will not be called back.

Step 16 Validate each of the five projects associated with the Courtesy Callback feature by right-clicking each Courtesy Callback project in the Navigator window and selecting **Validate**.

Deploy VXML Application to VXML Server

You can deploy a VXML application to the VXML Server using the File Transfer feature in Packaged CCE Administration web application.

-
- Step 1** After validating and saving your applications, in the navigator panel of Call Studio (top left), right-click and select all the applications you want to deploy.
- Step 2** Click **Deploy**.
- Step 3** In the Deploy Destination area, select **Archive File** and click **Browse**.
- Step 4** Navigate to the archive folder that you have set up; for example, `C:\Users\Administrator\Desktop\Sample`.
- Step 5** Enter the name of the file; for example, **Samplefile.zip**.
- Step 6** Click **Save**.
- Step 7** In the Deploy Destination area, click **Finish**.
- Step 8** Log in to the Packaged CCE Administration web application from the principal AW machine.
- Step 9** Choose **Overview > Call Settings > IVR Settings > File Transfers**.
- Step 10** Click **New** to open the **New File Transfer** page.
- Step 11** Click **Add to Server** to open the **Upload File** pop-up.
- Note** You can upload one file at a time.
- Step 12** Click **Click to select** and select a zip file to upload.
- Step 13** Click **Upload**.
- The file is uploaded to AW and listed under **Available Files in the Server**.
- Note** You can hover over a row and click the **x** icon to delete a file from the server.
- Step 14** Select one or more sites for the file transfer.
- Step 15** On the **Available Files in the Server** list, select the files that you want to transfer and click **Save**. This initiates the transfer of the selected files to VXML Server of the selected sites.
-

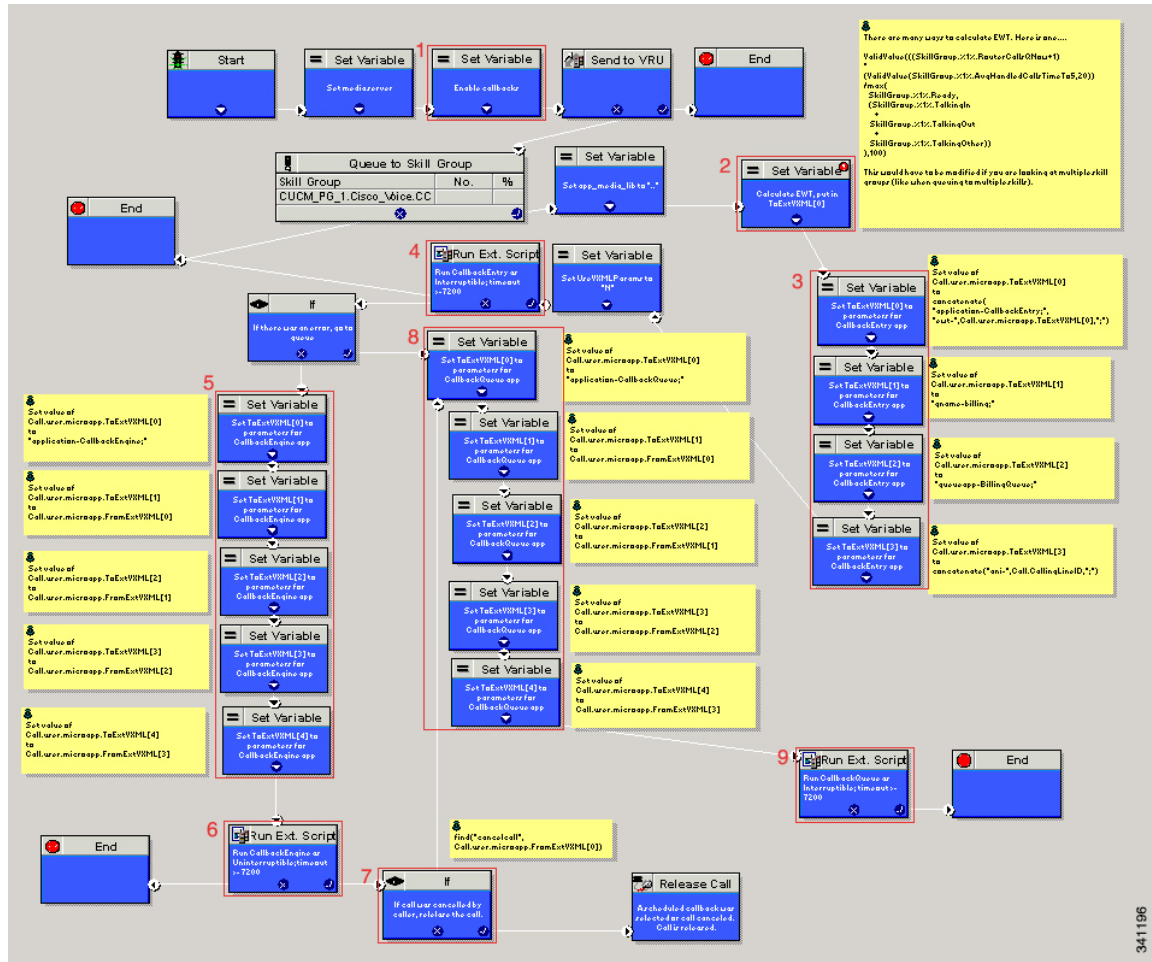
CCE Script for Courtesy Callback

The following discussion provides an overview of the scripts used for the courtesy callback feature. There are nine numbered blocks, or sets of blocks, identified in the following figure.



Note In the following example, the yellow comment blocks describe first the value being set and then the place where the value is being sent.

Figure 1: Setting Value for Courtesy Callback



The following bullets provide descriptions for the numbered blocks in the preceding graphic:

- Block 1: Enable callback or shut it off.
- Block 2: Compute average wait time. Once the caller is *in queue*, calculate the Estimated Wait Time (EWT) for that queue and place the value in ToExtVXML[0].

If there is poor statistical sampling because of sparse queues and the wait time cannot be calculated in the VXML Server, use the ICM-calculated estimated wait time.

One method of calculating EWT (the method used in this example) is:

```
ValidValue(((SkillGroup.%1%.RouterCallsQNow+1)
*
(ValidValue(SkillGroup.%1%.AvgHandledCallsTimeTo5,20)
/max(
SkillGroup.%1%.Ready,
(SkillGroup.%1%.TalkingIn
+
SkillGroup.%1%.TalkingOut
+
SkillGroup.%1%.TalkingOther))
),100)
```

Modify this method if you are looking at multiple skill groups (when queuing to multiple skills).

- Block 3: Set up parameters to be passed.
- Block 4: Run this block and prompt the caller. If the caller does not accept the offer for a callback, keep the caller in the queue and provide queue music.
- Block 5: Set up variables. Call flow returns to this block if the caller elects to receive a callback. Otherwise, the call remains queuing in the queuing application (BillingQueue in this example) on the VXML Server.
- Block 6: Run external to Callback engine to keep the call alive. If the agent becomes available and there is no caller, then agent can't interrupt (do not want an agent to pick up and have no one there).
- Block 7: Has the caller rejected the callback call? If no, then go to block 8.
- Block 8: Compute average wait time, as in block 2.
- Block 9: Set up variables.
- Block 10: Put caller briefly into queue (after caller accepts the actual callback call).

Modifiable Example Scripts and Sample Audio Files

The courtesy callback feature is implemented using Unified CCE scripts. Modifiable example scripts are provided. These scripts determine whether or not to offer the caller a callback, depending on the callback criteria (previously described). Sample audio files are also provided.

The example scripts and audio files are located on the CVP installation media in the `\CVP\Downloads` and `Samples\` folder.

The files provided are:

- `CourtesyCallback.ICMS`, the ICM script, in the `ICMDownloads` subfolder
- `CourtesyCallbackStudioScripts.zip`, a collection of Call Studio scripts, in the `helloStudioSamples` subfolder.

The following example scripts are provided:

- `BillingQueue`: Plays queue music to callers. Can be customized.
 - `Callback Engine`: Keeps the VoIP leg of the call alive when the caller elects to receive the callback (and ends the call) and when the caller actually receives the callback. *Do not* modify this script.
 - `CallbackEntry`: Initial IVR when caller enters the system and is presented with opportunity for a callback. Can be customized.
 - `CallbackQueue`: Handles the keepalive mechanism for the call when callers are in queue. *Do not* modify this script.
 - `CallbackWait`: Handles IVR portion of call when caller is called back. Can be customized.
- `CCBAudioFiles.zip` contains sample audio files that accompany the sample studio scripts.

If you use `CCBAudioFiles.zip`, unzip the contents onto the media server. `CCBAudioFiles.zip` has Courtesy Callback-specific application media files under `en-us\app` and media files for **Say It**

Smart under `en-us\sys`. If you already have media files for **Say It Smart** on your media server, then you only require the media files under `en-us\app`.



Note The Courtesy Callback sample files and scripts are also available on DevNet (**Customer Voice Portal (CVP) > Downloads > Courtesy Callback Sample Scripts**) at <https://developer.cisco.com/site/customer-voice-portal/downloads/courtesy-callback-scripts/>.

Overview of CCE Script Configuration for Courtesy Callback

The provided CCE script for Courtesy Callback contains the necessary sample elements for the Courtesy Callback feature. However, you must merge this script into your existing CCE scripts.

As a starting point and to run a simple test, import the script into the CCE script editor, validate it with the CCE script editor validation tool to locate nodes that need extra configuration (such as for Network VRU scripts and expanded call variables), and then modify the script according to your existing CCE environment.

The general process is as follows:

1. Locate each queue point in every CCE script. For example: Queue To Skill Group, Queue to Enterprise Skill Group, Queue to Scheduled Target or Queue to Agent.
2. Categorize each queue point according to the pool of resources that it is queuing for. Each unique pool of resources will ultimately require a queue in VXML Server if Courtesy Callback is going to be offered for that resource pool. For example, using the following example, QueueToSkill X and QueueToSkill Z are queuing for the exact same resource pool (despite the different queuing order). Queue to Skill Y, however, is queuing to a different pool because it includes Skill Group D.
 - QueueToSkillGroup X is queuing for Skill Group A, B, C in that order.
 - QueueToSkillGroup Y is queuing for Skill Group A, C and D in that order.
 - QueueToSkillGroup Z is queuing for Skill Group C, B, A in that order.
3. Assign a unique name to each unique resource pool. In the above example, we can use names ABC and ACD as example names.
4. For each resource pool, decide whether callbacks will be allowed in that resource pool. If yes, then every occurrence of that resource pool in all ICM scripts must be set up to use VXML Server for queuing. This is to ensure that the Courtesy Callback mechanism in the VXML Server gets a full, accurate picture of each resource pool's queue.
5. For any queue point where Courtesy Callback will be offered, modify all CCE scripts that contain this queue point according to the guidelines in the following CCE script examples.

Configure the CCE Script for Courtesy Callback

Many of the following configuration items relate to the numbered blocks in the diagram and provide understanding for CCE Script for Courtesy Callback (for more information, see [CCE Script for Courtesy Callback, on page 13](#)). Steps that refer to specific blocks are noted at the beginning of each step.

To configure CCE to use the sample Courtesy Callback CCE script, perform the following steps:

Step 1 Copy the CCE example script, **CourtesyCallback.ICMS** to the CCE Admin Workstation.

The example CCE script is available in the following locations:

- On the CVP install media in `\CVP\Downloads and Samples\`.
- On DevNet (**Customer Voice Portal (CVP) > Downloads > Courtesy Callback Sample Scripts**) at <https://developer.cisco.com/site/customer-voice-portal/downloads/courtesy-callback-scripts/>

Step 2 Perform these steps:

- a) Enable the user.CourtesyCallbackEnabled ECC variable.
- b) Enable the user.microapp.ToExtVXML ECC variable and verify that it is set up as an array with a maximum array size of 5 elements.
- c) Enable the user.microapp.FromExtVXML ECC variable and verify that it is set up as an array with a maximum array size of 4 elements.

Step 3 Make sure the VXML_Server_Interruptible and the VXML_Server_Noninterruptible Network VRU scripts exist.

Step 4 Once the script is open in Script Editor, open the **Set media server** node and specify the URL for your VXML Server.

For example: **http://10.86.132.139:7000/CVP**

With the current implementation of CVP, you do not have to specify the VXML Server URL. You do, however, have to enter *some* numeric value; for example "1" (with quotes).

Step 5 Map the route and skill group to the route and skill group available for courtesy callback.

- a) In Script Editor, select **File > Import Script...**
- b) In the script location dialog, select the **CourtesyCallback.ICMS** script and click **Open**.
- c) In the Import Script - Manual Object Mapping window, map the route and skill group to the route and skill group available for courtesy callback (identified previously).

In Packaged CCE deployments, the route tool does not exist. Routes have one-on-one mappings with skill groups, so when you create a skill group, a route is created with the same name.

Step 6 **Block #2:** If you wish to use a different estimated wait time (EWT), modify the calculation in block #2. You must do this if you use a different method for calculating EWT or if you are queuing to multiple skill groups.

Step 7 **Block #3:** Set up the parameters to be passed to CallbackEntry (VXML application).

Note This step assumes that you have already configured the CCE and expanded call variables not related to Courtesy Callback.

Variable values specific to Courtesy callback include:

```
ToExtVXML[0] = concatenate("application=CallbackEntry",";ewt=",Call.user.microapp.ToExtVXML[0])
```

```
ToExtVXML[1] = "qname=billing";
```

```
ToExtVXML[2] = "queueapp=BillingQueue;"
```

```
ToExtVXML[3] = concatenate("ani=",Call.CallingLineID,"");
```

Definitions related to these variables are:

- CallbackEntry is the name of the VXML Server application that is run.
- ewt is calculated in **Block #2**.

- `qname` is the name of the VXML Server queue into which the call is placed. There must be a unique `qname` for each unique resource pool queue.
- `queueapp` is the name of the VXML Server queuing application that is run for this queue.
- `ani` is the caller's calling Line Identifier.

Step 8 Verify that you have at least one available skill group to map to the skill group in the example script.

Step 9 Save the script, then associate the call type and schedule the script.

For information about scheduling scripts, see the *Cisco Packaged Contact Center Enterprise Administration and Configuration Guide* at <https://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-maintenance-guides-list.html>.

Administration and Usage

Element Specifications for Courtesy Callback

The example IVR scripts provided for Courtesy Callback work as installed. To change how Courtesy Callback works, you can change the configuration of Courtesy Callback elements. This section lists the elements associated with Courtesy Callback and briefly describes the purpose of each one.

Callback_Add

The `Callback_Add` element is used to add a callback object to the database after all the callback information has been collected from the caller. In addition, it can be optionally configured to automatically delete old recorded files at specified intervals. These recorded files are the files produced by the Record element when the user records their name if they want a call back in the CallbackEntry application.

Callback_Disconnect_Caller

The `Callback_Disconnect_Caller` element is responsible for disconnecting the caller's leg of the call. The IP leg of the call for Unified CVP is preserved to hold the caller's *place in line* until the callback is made back to the caller.

Callback_Enter_Queue

The `Callback_Enter_Queue` element is responsible for adding a new caller to the queue. This element must be run for all callers even if the caller may not be offered a callback.

Callback_Get_Status

The `Callback_Get_Status` element is responsible for retrieving all information about the callback related to the current call (if a callback exists).

Callback_Reconnect

The `Callback_Reconnect` element is responsible for reconnecting the caller's leg of the call.

Callback_Set_Queue_Defaults

The `Callback_Set_Queue_Defaults` element is responsible for updating the DBServlet with the values that should be used for each queue. There is always a *default* queue type. The values are used whenever a queue type is encountered for which there are no explicitly defined values. For example, if an administrator has defined values for a *billing* and *default* queues, but the caller is queued for *mortgages*. In that case, the application uses the values from `Callback_Set_Queue_Defaults`.

Note When the DBServlet is not reachable to check the callback status for the duration of keepalive interval, the callback entry in the Reporting Server gets marked as a stale cached entry and subsequently gets cleared. As a result, a callback is not initiated.

Callback_Update_Status

The `Callback_Update_Status` element is responsible for updating the database after a callback disconnect or reconnect.

Callback_Validate

The `Callback_Validate` element is responsible for verifying whether or not a callback can be offered to the caller during this call. Depending on the outcome of the validation, the Validate element exits with one of four states.

Callback_Wait

The `Callback_Wait` element is responsible for *sleeping* the application for X seconds. The application hands control back to `cvp_ccb_vxml.tcl` with the parameter `wait=X`.

