



Writing Scripts for Unified CVP

This chapter discusses using Unified ICME configuration and script editing to access the Unified CVP solution. It includes information about how to:

- Set up Unified ICME to interact with Unified CVP
- Write applications for Unified CVP



Note This chapter contains important information for IVR application developers. It also may be of interest to Call Center Managers, Unified CVP System Managers, and Unified ICME System Managers.

- [Before You Begin](#), on page 1
- [Making Unified ICME Work with Unified CVP](#), on page 2
- [Scripting for Unified CVP with Unified ICME](#), on page 4
- [Unified ICME Setup](#), on page 13
- [Writing Unified ICME Applications for Unified CVP](#), on page 13
- [Unified CVP Micro-Applications](#), on page 15
- [Scripting for Unified CVP with Call Studio](#), on page 67
- [Scripting for Unified CVP with Context Service](#), on page 75
- [System Media Files](#), on page 76

Before You Begin

This chapter makes the following assumptions:

- The information in this chapter assumes that you are already familiar with using Unified ICME software's ICM Configuration Manager and Script Editor tools for call center operations and management.



Note You should have a copy of the following Unified ICME documentation available in addition to this manual in order to successfully configure Unified ICME and use its features in conjunction with Unified CVP: [ICM Configuration Guide for Cisco ICM Enterprise Edition](#) and *ICM Scripting and Media Routing Guide for Cisco ICM/IPCC Enterprise & Hosted Editions*.

- When creating applications that interact with Unified CVP, only use alphanumeric characters for application, element, and field names; *do not* use special characters such as periods, asterisks or brackets. Follow this to avoid potential issues with data transfer between different systems.

Making Unified ICME Work with Unified CVP

Unified ICME determines where to route calls - whether to ACDs, specific agents, or to VRUs. However, the routing services themselves must be provided by an external routing client.

Traditionally, Unified ICME's routing clients were PSTN network switches, or in some cases, customer provided ACDs. Unified CVP provides VoIP routing capability for the Unified ICME and Unified CCE products. Unified CVP makes it possible for Unified ICME to use VoIP gateways as routing clients, as well traditional routing services.

Unified ICME and Unified CVP work together to perform such tasks as:

- Playing media, such as a recording stating office hours, to a caller.
- Playing streaming audio, such as a radio broadcast, to a caller.
- Retrieving caller-entered data, DTMF, or speech.
- Playing back different types of data, such as an account number or balance, to a caller.
- Moving calls to other destinations. For example, forwarding calls to an agent.

Unified ICME uses IVR messaging technology to direct Unified CVP and to receive the responses from Unified CVP.

Scripts to Access Unified CVP From Unified ICME

Both Unified ICME and Unified CVP use scripts to invoke their features. In fact, Unified ICME references Unified CVP scripts from *within* its own scripts. This method of invoking Unified CVP from within Unified ICME enables Unified ICME to take advantage of the features of Unified CVP.

The two products (Unified ICME and Unified CVP) provide two service creation (scripting) environments. Each environment is used for different purposes:

- **ICM Script Editor.** Use this scripting tool to develop agent routing scripts and to invoke the Unified CVP **micro-applications**: Play Media, Get Speech, Get Digits, Menu, Play Data, and Capture. These applications are the basic building blocks of a voice interaction design.
- **Call Studio.** Use Call Studio to develop sophisticated IVR applications.



Note For more information, refer to "[Scripting for Unified CVP with Call Studio.](#)"

Unified ICME Scripting

The Unified ICME Script Editor is used to develop agent routing scripts, and to invoke Unified CVP micro-applications - basic building blocks of a voice interaction design. The Unified CVP micro-applications

are: Play Media, Get Speech, Get Digits, Menu, Play Data, and Capture. These applications are combined and customized in the Unified ICME routing script to produce a viable voice interaction with the caller.

While it is possible to develop full scale IVR applications using micro-applications, it is not supported. Micro-application-based scripts are primarily used for initial prompt and collection operations, as well as for directing the playing of .wav files while calls are in queue. Instead, use the IVR scripts developed using Call Studio to create the IVR applications.

In an environment where Unified ICME script works with VXML script (the 2-script implementation for Unified ICME-integrated models described here), the Unified ICME script remains in control (and receives control back), even while it *delegates* the more complex self-service activity to the VXML Server script. Data can be passed from one script to the other and back through ECC variables.



Note The capability of using Unified ICME scripting for anything other than simple functions has been kept in support of legacy deployments. New customers are strongly advised to use the VoiceXML scripting environment of Unified CVP for creating IVR applications.

Unified CVP VoiceXML Scripting

Sophisticated IVR applications can be developed using Call Studio which is an Eclipse-based service creation environment whose output is an intermediary file describing the application flow. That file gets loaded onto the VXML Server machines for execution. To invoke a VXML Server application, the script writer includes a special micro-application in his Unified ICME routing script. This micro-application instructs the VoiceXML Gateway to interact with the VXML Server directly to execute the application. The final results are passed back to Unified ICME.

Some of the VoiceXML scripting environment features include:

- A drag-and-drop interface with a palette of IVR functions
- The ability to do database queries
- Extensibility with Java code written to perform any task a Java application can perform



Note Unified CVP does not support using the *MicroApp* nodes that are available in the ICM Script Editor. All MicroApp implementation must be done using the *Run External Script* node in ICM Script Editor. Refer to *ICM Scripting and Media Routing Guide for Cisco ICM/IPCC Enterprise & Hosted Editions* for detailed information about the Run External Script node. Refer to [Writing Unified ICME Applications for Unified CVP](#) for detailed information about setting Unified CVP-specific parameters in this node for each Unified CVP micro-application.



Note For more information about creating scripts, refer to "[Scripting for Unified CVP with Unified ICME.](#)"



Note When you invoke an application on VXML Server by providing a custom URL, ensure that the URL does not contain any space. Application invocation will not work if the URL contains a space.

Micro-Application Use versus VXML Server Use

The same special micro-application that is used to invoke VXML Server applications can also be used to invoke arbitrary "External VXML" pages from a Media Server or other customer-provided source. However, only use this capability for very simple VoiceXML needs, because Cisco has no way to verify that *customer-provided* VoiceXML documents are compatible with the IOS Voice Browser. (As opposed to VoiceXML documents that are generated by VXML Server, which *are* guaranteed by Cisco to be compatible with the IOS Voice Browser.) **Although the capability to use the micro-application has not been removed from the Unified CVP 4.0 and later offerings, customers are discouraged from using it directly.**

Additionally, all the VoiceXML Gateway sizing metrics that Cisco provides are based on the specific VoiceXML documents that are generated using either micro-applications or VXML Server applications. Using VoiceXML from another source will require you to perform your own empirical performance and capacity testing in order to determine how to size the VoiceXML Gateways.

Scripting for Unified CVP with Unified ICME

The sections that follow include:

- A discussion of micro-applications.
- A sample Unified ICME script.
- A discussion of how Unified ICME and Unified CVP exchange information.

Micro-Applications

Micro-applications are a set of specific IVR functions that can be invoked by Unified ICME, enabling communication with the caller.

There are six Unified CVP micro-applications:

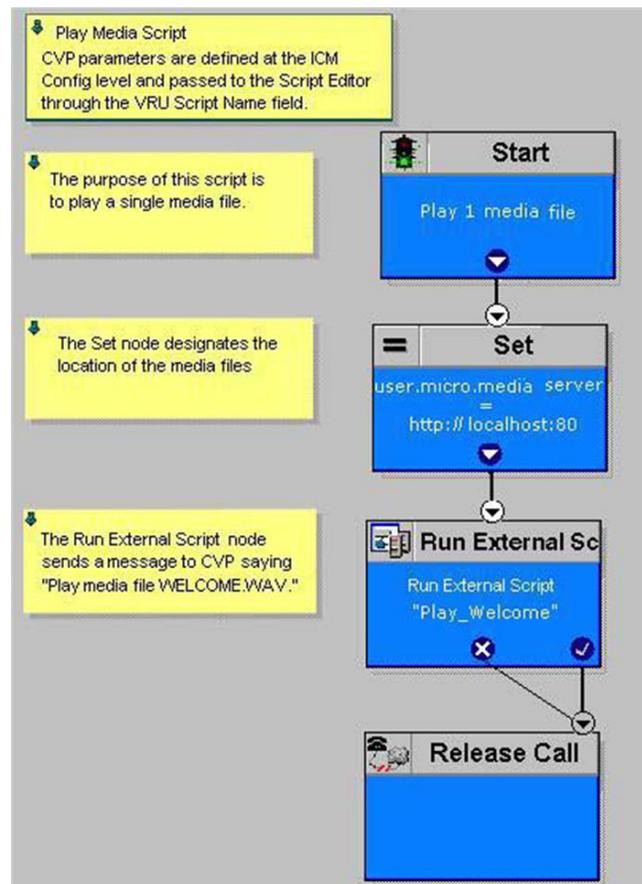
- **Play Media.** Plays a message to the caller.
- **Play Data .** Retrieves data from a storage area and plays it to the caller in a specific format called a data play back type.
- **Get Digits.** Plays a TTS or media file and retrieves digits from the caller.
- **Menu.** Plays a TTS or media menu file and retrieves a single telephone keypad entry from the caller.
- **Get Speech.** Collects ASR or DTMF input after prompting a caller.
- **Capture.** The Capture (CAP) micro-application enables you to trigger the storage of current call data at multiple points in the Unified ICME routing script.

Micro-applications are interpreted by the IVR Service, which resides on the VXML Server. The IVR Service sends VoiceXML code to the VoiceXML Gateway Voice Browser.

The IVR Service also accepts HTTP requests from the VoiceXML Gateway's Voice Browser, and communicates those requests to Unified ICME's Service Control Interface using the ICM Service.

Simple Example Script: Welcome to XYZ Corporation

Suppose you want to create a Unified ICME script that simply plays a message, "Welcome to XYZ Corporation." From the Unified ICME's perspective, there is no difference between a script written for a standard "black box" IVR or the Unified CVP, so you can create a script such as the one shown in the following figure.



This simple script performs three functions:

- Sends the Run External Script request to Unified CVP.
- Indicates the location of the "Welcome" media file.
- Releases the call.



Note In a “real life” application, any Unified ICME script you create would include error checking to ensure that micro-applications instructions are properly executed.

Unified ICME Unified CVP Micro-app Connection

Before the Unified CVP IVR solution can be accessible through the Script Editor’s Run External Script node, you must first set up Unified ICME with special Unified CVP parameters using the ICM Configuration Manager tool.

Begin by using the ICM Configuration Manager’s Network VRU Script window to define Unified CVP parameters:

- **PM,Welcome.** (VRU Script Name field.) This means: "Use the instructions in the Play Media micro-application to play the Welcome.wav media file."
- **N.** (Configuration Param field.) This means: "Do not allow barge-in." (Barge-in is when the caller can interrupt message play by entering a digit, causing the script to move to the next prompt.)
- You *must* check the **Interruptible** checkbox as shown in the figure above. This specification allows the script to be interrupted by the Unified CVP script functions.



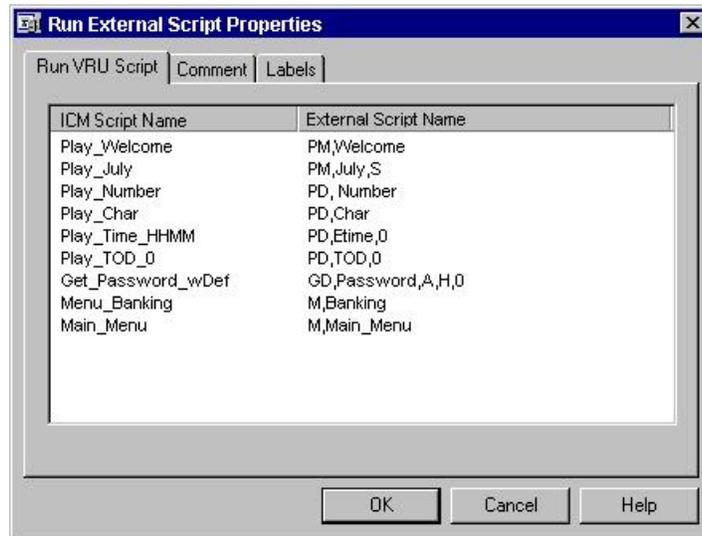
Note As shown in the two columns of the following table, certain entries for the VRU Script Name and Configuration Param fields are case-sensitive.

Network VRU Script Field Attributes That Are Case-Sensitive	Network VRU Script Field Attributes That Are Not Case-Sensitive
<p>Applies to: All micro-applications.</p> <p>Attribute: Media File Name (for example, media or .vxml)</p>	<p>Applies to: All micro-applications.</p> <p>Attribute: VRU Script Name (for example, PM, GD).</p>
<p>Applies to: Get Speech (GS).</p> <p>Attribute: External Grammar File name.</p>	<p>Applies to: All micro-applications.</p> <p>Attribute: Media Library Type (A, S, V)</p>
	<p>Applies to: All micro-applications.</p> <p>Attribute: Barge-in Allowed (Y/N),</p>
	<p>Applies to: PlayData (PD).</p> <p>Attribute: Data playback type (for example, Number, Char).</p>
	<p>Applies to: PlayData (PD).</p> <p>Attribute: Time Format (HHMM, HHMMSS, HHMMAP),</p>

Network VRU Script Field Attributes That <i>Are</i> Case-Sensitive	Network VRU Script Field Attributes That <i>Are Not</i> Case-Sensitive
	<p>Applies to: Get Digits (GD), Get Speech (GS), Menu (M).</p> <p>Attribute: Timeout Message Override (Y/N)</p>
	<p>Applies to: Get Digits (GD), Get Speech (GS), Menu (M).</p> <p>Attribute: Invalid Entry Message Override (Y/N).</p>
	<p>Applies to: All micro-applications.</p> <p>Attribute: DTMF Termination Key (only N)</p>
	<p>Applies to: Get Speech (GS).</p> <p>Attribute: Type of Data to Collect (for example, boolean, date).</p>

Once the ICM Configuration Manager’s settings have been saved, the information is available to the Script Editor. When you place a Run External Script node in the Script Editor workspace and open the Properties dialog box, it displays all the script names defined in the system.

The Run External Script node below shows that the ICM Script Name Play_Welcome was selected.



Information Exchange Between Unified ICME and Unified CVP

When Unified ICME processes a Run External Script node, parameters are sent to Unified CVP.

These parameters contain instructions about how to interact with a caller, such as:

- What micro-application to use.
- The location of the media files to be played to the caller.

- Timeout settings to be used during caller digit entry.

Some IVR parameters are passed to Unified CVP through Expanded Call Context (ECC) variables and/or Call.Peripheral variables. Other parameters are sent in the normal VRU messaging interface (Unified ICME/IVR Service Control Interface).

Unified ICME Data Handling

In defining scripts, you might specify strings, numbers, or formulas to be sent to Unified CVP. When passing numbers to Unified CVP, always enclose them in quotes so that they will be processed as a string.

This is especially important if:

- Leading 0's are significant to the data type (times, character), enter the number as a quoted string (example: "031524").
- Trailing 0's after a decimal point are significant to the data type (number, character, currency), enter the number as a quoted string (examples: "42.00" or "42.10").
- The number is very large (example: a number normally expressed through exponential notation).

Unified CVP Script Error Checking

Unified CVP uses the **user.microapp.error_code** ECC variable to return information regarding problems encountered while running a script.

Unified CVP software tests for the following conditions when processing Unified ICME scripts:

ASR Error

Failure of an Advanced Speech Recognition component.

General error

General error occurred.

Invalid Configuration Param

Data passed from Unified ICME to the IVR Service is not consistent with what the micro-application requires for processing.

Invalid variable data

The variable data passed was not valid for the script type being processed.

Invalid VRU Script Name format

VRU Script Name data passed from Unified ICME to the IVR Service does not contain the expected components (micro-application name, media file name, media file type, uniqueness value).

Locale

Locale was not supported. (Only applies to Play Data micro-applications that use .wav files. Does not apply to Play Data micro-applications that use TTS, or to Play Media, Get Digits, Menu, Get Speech, or Capture micro-applications.)

Misconfigured ECC variable

An ECC variable was set to a value the IVR Service did not recognize. ECC variable definitions must be the same in Unified ICME and Unified CVP.

Network Error

Failure of an IP network connection.

Reached Maximum Invalid Tries

Caller was unsuccessful in entering digits during each of the tries allowed by the micro-application. (Only applies to Get Digits, Menu, and Get Speech micro-applications.)

Reached Maximum Number Entry Tries

Caller did not enter digits in response to the prompt for each of the tries allowed by the micro-application. (Only applies to Get Digits and Get Speech micro-applications.)

Semantic-Runtime

Semantic error occurred while running a micro-application.

System Error

Unexpected failure of a Unified CVP component.

Timed Out

Caller did not enter digits in response to the prompt in the time allowed by the micro-application.

TTS Error

Failure of a Text-to-Speech component.

Unavailable Media file

Media file name passed from Unified ICME to the IVR Service did not exist on the Media Server.

Unknown micro-application

Micro-application name passed from Unified ICME to the IVR Service did not exist on the IVR Service.

Unsupported locale

The VoiceXML Interpreter (that is, Gateway) did not recognize the locale passed from the IVR Service.

Unsupported VoiceXML element

The VoiceXML Interpreter (that is, Gateway) did not recognize a VoiceXML element passed from the IVR Service, VXML Server, or media server (using External VoiceXML).

Unsupported VoiceXML format

The VoiceXML Interpreter (that is, Gateway) did not recognize a VoiceXML format passed from the IVR Service, VXML Server, or media server (using External VoiceXML).

Each Unified CVP micro-application has individualized settings for **user.microapp.error_code** (non-video and video), as shown in the following table.

Table 1: Possible User.Microapp.error_code ECC Variable Settings for Non-Video

Error Code	Play Media	Play Data	Get Digits	Menu	Get Speech	Capture
0	No error	No error	No error	No error	No error	No error

Error Code	Play Media	Play Data	Get Digits	Menu	Get Speech	Capture
1	Caller Hangup	Caller Hangup	Caller Hangup	Caller Hangup	Caller Hangup	N/A
2	Network Error	Network Error	Network Error	Network Error	Network Error	N/A
3	System Error	System Error	System Error	System Error	System Error	System Error
5	Unknown micro-application	Unknown micro-application	Unknown micro-application	Unknown micro-application	Unknown micro-application	Unknown micro-application
6	Invalid VRU Script Name format	Invalid VRU Script Name format	Invalid VRU Script Name format	Invalid VRU Script Name format	Invalid VRU Script Name format	N/A
7	Invalid Configuration Param	Invalid Configuration Param	Invalid Configuration Param	Invalid Configuration Param	Invalid Configuration Param	N/A
8	Misconfigured ECC variable	Misconfigured ECC variable	Misconfigured ECC variable	Misconfigured ECC variable	Misconfigured ECC variable	N/A
9	One of the following: <ul style="list-style-type: none"> Media or external VXML file does not exist Invalid URL for Media or external VXML file External VXML is in an invalid format 	One of the following: <ul style="list-style-type: none"> Media file does not exist Invalid URL for Media L file 	One of the following: <ul style="list-style-type: none"> Media file does not exist Invalid URL for Media L file 	One of the following: <ul style="list-style-type: none"> Media file does not exist Invalid URL for Media L file 	One of the following: <ul style="list-style-type: none"> Media or external VXML file does not exist Invalid URL for Media or external VXML file External VXML is in an invalid format 	N/A
10	Semantic-Runtime Error	Semantic-Runtime Error	Semantic-Runtime Error	Semantic-Runtime Error	Semantic-Runtime Error	N/A
11	Unsupported VoiceXML format	Unsupported VoiceXML format	Unsupported VoiceXML format	Unsupported VoiceXML format	Unsupported VoiceXML format	N/A
12	Unsupported VoiceXML element	Unsupported VoiceXML element	Unsupported VoiceXML element	Unsupported VoiceXML element	Unsupported VoiceXML element	N/A
13	N/A	Variable data is invalid	N/A	N/A	N/A	N/A

Error Code	Play Media	Play Data	Get Digits	Menu	Get Speech	Capture
14	N/A	Location of variable data is empty	N/A	N/A	N/A	N/A
15	N/A	Time format is invalid	N/A	N/A	N/A	N/A
16	N/A	N/A	Reached Maximum Invalid Tries	Reached Maximum Invalid Tries	Reached Maximum Invalid Tries	N/A
17	N/A	N/A	Reached Maximum No Entry Tries	Reached Maximum No Entry Tries	Reached Maximum No Entry Tries	N/A
20	N/A	Data value out of range	N/A	N/A	N/A	N/A
23	No answer	No answer	No answer	No answer	No answer	N/A
24	Busy	Busy	Busy	Busy	Busy	N/A
25	General transfer error	General transfer error	General transfer error	General transfer error	General transfer error	N/A
26	Invalid extension	Invalid extension	Invalid extension	Invalid extension	Invalid extension	N/A
27	Called party hung up	Called party hung up	Called party hung up	Called party hung up	Called party hung up	N/A
28	Error after transfer established	Error after transfer established	Error after transfer established	Error after transfer established	Error after transfer established	N/A
30	Unsupported locale	Unsupported locale	Unsupported locale	Unsupported locale	Unsupported locale	N/A
31	ASR error	ASR error	ASR error	ASR error	ASR error	N/A
32	TTS error	TTS error	TTS error	TTS error	TTS error	N/A
33	General ASR/TTS error	General ASR/TTS error	General ASR/TTS error	General ASR/TTS error	General ASR/TTS error	N/A
34	Unknown error	Unknown error	Unknown error	Unknown error	Unknown error	N/A
40	VXML Server system unavailable	N/A	N/A	N/A	VXML Server system unavailable	N/A
41	VXML Server application error	N/A	N/A	N/A	VXML Server application error	N/A

Error Code	Play Media	Play Data	Get Digits	Menu	Get Speech	Capture
42	VXML Server application used hangup element instead of subdialog return element	N/A	N/A	N/A	VXML Server application used hangup element instead of subdialog return element	N/A
43	VXML Server application is suspended	N/A	N/A	N/A	VXML Server application is suspended	N/A
44	VXML Server session error (for example, application has not yet been loaded)	N/A	N/A	N/A	VXML Server session error (for example, application has not yet been loaded)	N/A
45	VXML Server encounters a bad fetch error (for example, media or grammar file not found)	N/A	N/A	N/A	VXML Server encounters a bad fetch error (for example, media or grammar file not found)	N/A
46	Audio streaming error	N/A	N/A	N/A	N/A	N/A

**Note**

- **user.microapp.error_code** is always zero, indicating success, if control proceeds out the Checkmark (success) branch of the Run External Script node. Usually, if control proceeds out the X (failure) branch, Unified CVP sets this variable to one of the codes listed here. (Set up your routing script to always test the error code after an X branch is taken.)
- However, if a configuration error, or a network or component failure of some sort, prevents the micro-application from being executed at all, then Unified CVP does not get a chance to set this variable at all. Such cases can be identified by using a Set node to pre-set **user.microapp.error_code** to some known invalid value such as -1, and then to test for that value using an If node, following the X branch of the Run External Script node.
- These problems can prevent the execution of the micro-application:
 - Inability to route the call to an appropriate VoiceXML-enabled gateway and IVR Service (VRU-Only call flow model only). Temporary network outage or other component failure can cause this error.
 - Mismatch between Network VRU associated with the configured VRU script and Network VRU associated with Unified CVP that is handling the call. ICM configuration error can cause this error.

Unified ICME Setup

Before you can use Unified ICME features to access the Unified CVP solution, you must perform some initial setup tasks to enable communication between Unified ICME and Unified CVP. These setup tasks are determined by Unified CVP call flow model; see the *Configuration Guide for Cisco Unified Customer Voice Portal*, for setup procedure for each model.



Note For more information about the supported Unified CVP call flow models, see the *Configuration Guide for Cisco Unified Customer Voice Portal*.

Writing Unified ICME Applications for Unified CVP

Once Unified ICME-to-Unified CVP initial setup is complete, you can create Unified ICME applications to access Unified CVP micro-applications.

You do this using two Unified ICME software tools:

- Configuration Manager
- Script Editor

The sections that follow give a brief overview of how to use these tools to access Unified CVP functionality.

Configure a Unified CVP Network VRU Script

Procedure

Step 1 Within the ICM Configuration Manager, select **Tools > List Tools > Network VRU Script List**.

Step 2 In the Network VRU Script List window, enable the **Add** button by clicking **Retrieve**.

Step 3 Click **Add**.

The Attributes property tab is enabled.

Step 4 Complete the Attributes tab as described below.

Warning The format of the strings for the **VRU Script Name** and **Configuration Param** fields are *very specific* and vary for different micro-applications (Play Media, Play Data, Get Digits, Menu, and Get Speech).

- **Network VRU**. (Drop-down list.) The name of the Network VRU to be associated with the Network VRU script.
- **VRU Script Name**. A 39-character, comma-delimited string used by Unified CVP to pass parameters to the IVR Service. The content of string depends on the micro-application to be accessed. For more information on what to specify in this field, refer to the following sections:

- [Play Media Micro-Application](#)
 - [Play Data Micro-Application](#)
 - [Get Digits Micro-Application](#)
 - [Menu Micro-Application](#)
 - [Get Speech Micro-Application](#)
- **Name.** A unique name for the VRU script. Unified ICME generates a name based on the Network VRU and script names.
 - **Timeout.** The number of seconds Unified ICME waits for a response from the VRU after invoking the script before assuming that the Unified CVP script has failed.
 - Warning** This setting is designed to detect VRU failures only; attempting to use it as a technique for interrupting script processing can lead to unexpected results. Use the 180-second default or lengthen the setting to a duration that is longer than the longest time the script is expected to take.
 - **Configuration Param.** A string used by Unified CVP to pass additional parameters to the IVR Service. The content of string depends on the micro-application to be accessed. For more information on what to specify in this field, refer to the following sections:
 - [Play Media Micro-Application](#)
 - [Play Data Micro-Application](#)
 - [Get Digits Micro-Application](#)
 - [Menu Micro-Application](#)
 - [Get Speech Micro-Application](#)
 - **Description.** Any additional information about the script.
 - **Customer.** (Optional.) A customer associated with the script. For Service Provider solutions, this field is mandatory, due to multiple tenancy solutions (customer-specific data needs to be separated).
 - **Interruptible.** (Checkbox.) Whether Unified ICME can interrupt the script (for example, if a routing target becomes available).
 - **Overridable.** (Checkbox.) Indicates whether the script can override its own Interruptible attribute. Options: This setting does not apply to Unified CVP micro-applications.

Step 5 When finished, click **Save** to apply your changes.

Run External Script Node That Accesses a Unified CVP Micro-Application

Procedure

-
- Step 1** Within Script Editor, place the Run External Script object in the workspace, right-click, and open the Properties dialog box.
- The Run External Script Properties dialog box lists all Network VRU scripts currently configured
- Note** The ICM Script Name column reflects the values defined through the Name field in ICM Configuration Manager's Network VRU Script List tool.
- Step 2** Select the **ICM Script/VRU Script Name** you want to execute.
- Step 3** Modify the Comments tab, if required.
- Step 4** Modify the Labels tab, if required.
- Step 5** When finished, click **OK** to submit the changes and close the dialog box.
-

Unified CVP Micro-Applications



Note Not all third-party Automatic Speech Recognition (ASR) servers use Unified CVP micro-application parameters the same way. This affects how third-party ASR servers interact with the Unified CVP micro-applications. For example, although Unified CVP allows timeout parameters to be set to a value from 1 to 99 seconds, an ASR server may only support a range of 1 to 32 seconds. Another ASR server requires a "#" to indicate that digits are to be collected before the inter-digit timeout is reached. Be sure to follow the instructions provided by your third-party vendor. Test your micro-applications before deploying them.

The sections that follow describe the parameters that can be defined through ICM Configuration Manager for each of the six Unified CVP micro-applications.

Keep the following in mind as you configure each Network VRU Script to be used with Unified CVP:

- Each micro-application parameter in fields of the Network VRU Script List's Attributes tab must be separated by a comma.
- If a parameter value is not specified, the micro-application uses its default.

Each section concludes with sample ICM Configuration Manager and Script Editor screen captures for the micro-application.

Micro-Applications Automatic Speech Recognition and Text-to-Speech

Unified CVP micro-applications can use ASR in two ways:

- in Get Digits and Menu micro-applications, to recognize data for built-in data types, such as numbers, dates or currency, using digits and/or voice. The `user.microapp.input_type` ECC variable specifies the collection type. The script writer uses this variable in a Script Editor Set node to allow the caller to input

DTMF only (**D**) or both DTMF and Voice (**B**, the default). If you are not using an ASR, you need to set this variable to **D**. If you are using an ASR, you can set the variable to either **D** or **B**. Regardless of the value of **user.microapp.input_type**, the recognized digit(s) are always returned to ICM in the CED variable.



Note With **input_mode** set to "B" (both), either DTMF or speech is accepted, but mixed mode input is not. When you enter with one mode, input by the other mode is ignored and has no effect.

- in Get Speech micro-applications, to collect voice input according to a specified grammar. The grammar is specified either as inline grammar (through the setting in the **user.microapp.grammar_choices** ECC variable) or as an external grammar file (through a text file, the name of which is given in the Network VRU Script's Configuration Param field). The recognized result is returned to ICM in the **user.microapp.caller_input** ECC variable.

Unified CVP micro-applications can use TTS for two purposes:

- As an alternative for playing recorded announcement prompts with the Play Media, Get Digits, Menu, and Get Speech micro-applications, using either the contents of the **user.microapp.inline_tts** or an external .vxml file. See "[Micro-Applications External Voice XML](#)." The ECC variable is useful if the amount of text is short and simple. The external .vxml file is useful for more lengthy text or text that needs to be changed frequently using tools other than the ICM Script Editor.
- As a method of playing data using the Play Data micro-application. If the **user.microapp.pd_tts** ECC variable contains **Y**, Unified CVP uses TTS to speak the data (depending on the TTS locale support and capabilities); if **N**, Unified CVP uses the system recorded announcements to speak the data (depending on IVR Service locale support and capabilities).



Note These ECC variables must be set in the Unified ICME script prior to executing the micro-application that they modify.

Micro-Applications External Voice XML

The Play Media and Get Speech micro-applications render external .vxml; that is, text Voice-XML files. To access the external file, the Media File Component of the Network VRU Script's VRU Script Name field must point to a .vxml file and specify **V** as the Media Library Type parameter.

The external VoiceXML file must contain particular call control catch blocks and must not execute call control, as Unified CVP and Unified ICME must be responsible for all call control. See [External VoiceXML File Contents](#).

Dynamic Audio File Support for Micro-Applications

In ISN 2.0 (an earlier release of Unified CVP), audio files needed to be specified in the VRU Script Name of the Play Media, Menu, Get Digits and Get Speech micro-applications. Unified CVP lets you use a single micro-application and specify the prompt using call variables and the ICM formula editor.

To provide dynamic audio file capability, set the second VRU script parameter to a numeric value, 1-10, prefixed by a dash. You then set the Media Library to either “A”, “S”, or “V”. Unified CVP looks in the corresponding Call.PeripheralVariable for the name of the audio file to play.

When you set the Media Library to “A” or “S”, Unified CVP plays the audio file specified by the Call Variable after the “-(number)”. For example, if the second VRU Script Parameter is set to “-4”, it plays the audio file specified in Call.PeripheralVariable4. This functionality is added for Play Media, Menu, Get Digits, and Get Speech micro-applications.

If you set the Media Library to “V”, Unified CVP calls the external VoiceXML file specified by the Call Variable after the “-(number)”. If the Script Parameter is set to “-7”, for example, it calls the external VoiceXML file specified in Call.PeripheralVariable7.



Note The “V” option is only supported for the Play Media and Get Speech micro-applications.

Second VRU Script Parameter	Corresponding Call Variable
-1 to -10	Call.PeripheralVariable (1 to 10)

For an example of how to use a dynamic audio file, see the following table.

VRU Script Parameter Example	Definition
PM, -3,V	<p>PM - Uses the Play Media micro-application.</p> <p>-3 - Plays the file specified in Call.PeripheralVariable3.</p> <p>V - Acquires the file from the external VoiceXML Media Library.</p>

Example of Using the Dynamic Prompt

Use the dynamic prompt as follows:

- In the Set node in a Unified ICME script, set the value of ToExtVXML[0] to:

```
prompt=http://152.217.34.252/en-us/app/Welcome.wav
```

- In the external VoiceXML file specify the following configuration:

```
<?xml version="1.0"?>
<vxml version="2.0">
<form id="BilingualMenu" scope="dialog">
<var name="prompt"/>
<field name="caller_input">
<prompt bargein="true" timeout="3s">
<audio expr="prompt"/>
</prompt>
```



Note A specific hostname, wav filename, and form ID, is used in this example. Replace these elements with your own configuration settings.

Notes

- If you do not specify a file extension for the file name in the `Call.PeripheralVariable`, the default media file extension is applied.
- If you set the second VRU script parameter to a value prefixed with a dash and don't specify a file name in the corresponding `Call.PeripheralVariable`, the IVR Service creates a VoiceXML that does not contain a media prompt.
- If you set the second VRU Script Parameter to a value prefixed with a dash and set the “App Media Library” to `V`, signifying external VoiceXML, you must specify a VoiceXML file in the corresponding `Call.PeripheralVariable`. If you do not, an “Invalid VRU Script Name” error is returned to ICM. If the specified VoiceXML filename does not contain an extension, and `user.microapp.UseVXMLParams` is not set to `N`, the default extension of `.vxml` is automatically added.
- You can only specify the name of a single file in the Peripheral Variable. You cannot set this value to a name/value pair.

For more information, refer to the sections on individual micro-applications in this chapter.

Default Media Server for Micro-Applications

In prior releases, the only way to specify a media server for a micro-application was to use the ECC variable `user.microapp.media_server`. You can now use the Operations Console to designate a default media server for the entire deployment.

The global default media server can be specified in the Operations Console by selecting **Device Management > Media Server**. The default media server is used by the micro-applications if the ECC variable `user.microapp.media_server` is missing or empty in the Unified ICM script.

The following list specifies the order in which the micro-application tries to resolve which media server to use:

1. Media server is specified by the ECC variable: `user.microapp.media_server`
2. Global default media server is specified

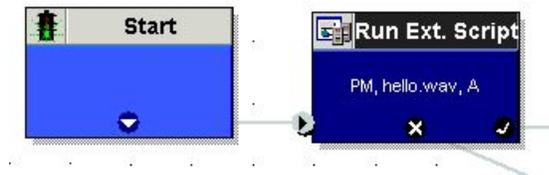
The first non-empty media server value encountered in the above order is used by the micro-application. This applies to all micro-applications including

- Play Media (PM)
- Play Data (PD)
- Get Digits (PD)
- Menu (M)
- Get Speech (GS) only if the Media Library Type in the VRU Script is set to `A` (Application) or `s` (System) but not `v` (ExternalVXML)

The following screen shot shows the Unified ICM script where Play Media micro-application plays a media file using the ECC variable `user.microapp.media_server`.



The following screen shot shows the Unified ICM script where Play Media micro-application plays a media file using a default media server configured in the Operations Console.



Capture Micro-Application

The Capture (CAP) micro-application allows you to trigger the storage of current call data at multiple points in the ICM routing script. The CAP micro-application must be configured as a VRU script, and it is executed using a RunExternalScript node, just as with any other Unified CVP micro-application. The VRU Script Name value is "CAP" or "CAP,xxx," where "xxx" is any arbitrary string to be used if necessary for uniqueness purposes. There is no VRU Script Config string.

Executing a Capture micro-application causes the ICM PG to produce an intermediate termination record. Specifically, it writes a record in the Termination_Call_Detail (TCD) table which includes all current call variables (not the VRUProgress variable), router call keys, date and time, and caller entered digits. Together with the TCD record, the Capture micro-application writes a set of records to the Termination_Call_Variable (TCV) table which includes the current values of all ECC variables.



Note To capture the ECC variables in the TCV table, enable the **persistent** check box in the **Expanded Variable List** window in the **ICM configuration manager**.

Unified ICME provides no standard reporting templates for TCD and TCV records. These tables are large and minimally indexed, and are optimized for writing rather than querying, to minimally impact call handling throughput. If you plan to report on this data, create off-hours extract processes which copy rows in their raw format into a database which is external to ICM. From there you can organize the tables in the way that best supports your querying requirements.

Information you need about these records includes:

- TCD records for a given call may be identified because they contain the same RouterCallKeyDay and RouterCallKey. Successive TCD records are ordered by incrementing RouterCallKeySequenceNumber.
- Intermediate TCD records may be identified because they contain a CallDisposition of 53, "PartialCall". Only the last TCD record for the call contains the actual disposition.

- TCV records corresponding to a particular TCD record may be obtained by joining on TCV.TCDRecoveryKey. This key matches the RecoveryKey value in the TCD record.
- As of Unified ICME 6.0(0), the TCD record's CallTypeId is also populated for VRU peripherals. This means you can determine the call's current CallType at each Capture micro-application invocation, and at the end of the call.
- In Unified CVP Comprehensive call flow models, these records are associated with the VRU leg peripheral. If you are doing VRU application reporting, you can filter for TCD records which contain the PeripheralID of the ISN VRU leg.

The Capture micro-application, places a heavy demand on ICM resources. Each time you use it, ICM writes one TCD record and multiple TCV records. Though it can conveniently capture the information you need, it can also capture extra information which you do not require. If you overuse this micro-application, it can place a heavy load on ICM in terms of processing time and disk space, which despite the minimal indexing, may impact ICM's ability to handle the expected call load. Carefully choose where you need to capture information in your scripts. Spread data items into as many call variables as possible to maximize the usefulness of each invocation.

Play Media Micro-Application

The Play Media (PM) micro-application can be configured to play a message that is contained in a media file or streaming audio file.

Configure Network VRU Script for Play Media

Use the ICM Configuration Manager's Network VRU Script List tool's Attributes tab to specify parameters.



Note DTMF digit type-ahead is not supported by Play Media and Play Data micro-apps when executing in Comprehensive mode (Type 7). However, this feature is supported for Type 5 calls.

Procedure

Step 1 Configure VRU Script field parameters:

- **Micro-application type.** For Play Media, valid options are: **PM** or **pm**.
- **Media File Name.** Name of the media file to be played (that is, the prompt file) or the name of the external VoiceXML file.

The valid options are:

- A file name (for instance, a .wav file).

Note Media file names are case-sensitive.

- **null** - (default) If this field is empty, Unified CVP examines the contents of the **user.microapp.inline_tts** ECC variable. If this ECC variable contains a value, Unified CVP prompts using TTS. If the ECC is empty, no prompt is played.

- **-(number 1-10)** - Unified CVP plays the file in the corresponding Call.PeripheralVariable file. For example, a value of 2 instructs Unified CVP to look at Call.PeripheralVariable2.

Note If you use the - (number 1-10) option and set the Media Library Type to "V," Unified CVP plays the external VoiceXML file specified in the corresponding Call.PeripheralVariable. If you set the value to - (no value) and set the Media Library Type to "A" or "S", the IVR Service creates VoiceXML without a media prompt.

- **-a** - Unified CVP automatically generates the media file name for agent greeting when this option is specified. The file name is based on GED-125 parameters received from Unified ICM. This option is only valid if the Media Library Type is not set to V.

- **Media Library Type.** Flag indicating the location of the media files to be played.

The valid options are:

- **A** - (default) Application
- **S** - System
- **V** - External VoiceXML

- **Uniqueness value.** Optional. A string identifying a VRU Script Name as unique.

Step 2 Configure the Configuration Param field parameters:

- **Barge-in Allowed.** Specifies whether barge-in (digit entry to interrupt media playback) is allowed.

The valid options are:

- **Y** - (default) barge-in allowed
- **N** - barge-in not allowed

Note Voice barge-in is not supported by Play Media and Play Data micro-applications. However, Dual Tone Multifrequency (DTMF) barge-in is supported for these micro-applications.

Unified CVP handles barge-in as follows: If barge-in *is not* allowed, the SIP Service/Gateway continues prompt play when a caller starts entering digits, and the entered digits are discarded. If barge-in *is* allowed, the Gateway discontinues prompt play when the caller starts entering digits. See [Get Speech and External VoiceXML](#).

- **RTSP Timeout.** Specifies the Real-time Streaming Protocol (RTSP) timeout - in seconds - when RTSP is used.

The valid range is 0 - 43200 seconds (default is 10 seconds). If the value is set to 0 or a timeout value is not provided, the stream does not end.

See [Configure Play Media Micro-Application to Use Streaming Audio](#) for more details.

- **Type-ahead Buffer Flush.** The Cisco VoiceXML implementation includes a type-ahead buffer that holds DTMF digits collected from the caller. When the VoiceXML form interpretation algorithm collects user DTMF input, it uses the digits from this buffer before waiting for further input. This parameter controls whether the type-ahead buffer is flushed after the prompt plays out. A false value (default) means that the type-ahead buffer is not flushed after the prompt plays out. If the prompt allows barge-in, the digit that barges in is not flushed.

The valid options are

- **Y** - flush the type-ahead buffer
- **N** - (default) do not flush the type-ahead buffer

Note This parameter is only applicable when using the Cisco IOS gateway with DTMF barge-in. This parameter is not applicable when using external VXML. This parameter is normally used when two or more PM and/or PD microapps are used in a loop in the ICM script (such as while in queue for an agent). If the PM and/or PD microapps are enabled for barge-in, one would set this parameter to **Y** to prevent an uncontrolled looping in the ICM script when the user barges in.

Configure Play Media Micro-Application to Use Streaming Audio

Use the ICM Script Editor to configure Play Media (PM) micro-application to play .wav files from a streaming audio server.

Cisco does not sell, OEM, or support any Media Servers. The IOS gateway only supports μ -law wav files in 8-bit format. Media Servers such as RealNetwork's Helix™ Server will serve RTSP broadcast audio streams in the μ -Law format. See *Configuration Guide for Cisco Voice Portal* to configure Helix Server for use with CVP.



Note Only RealNetwork's Helix™ server has been tested with CVP for RTSP broadcast audio streams in the μ -Law format, and none of the other streaming servers are supported.

Procedure

Step 1 Add a Set Node in the script to configure the media_server ECC variable.

- On the Set Variable tab of the Set Properties dialog box, select **Call** from the Object Type drop down and then set the Variable to user.microapp.media.server.

The screenshot shows the 'Set Properties (Read Only)' dialog box with the following configuration:

- Object type:** Call
- Object:** (No selection)
- Variable:** user.microapp.media_server
- Array index:** (empty)
- Value:** "rtsp://10.86.129.250:554/broadcast"

- In the Value field, specify the URL up to, but not including, the stream name.

Note The URL must begin with an *rtsp://* prefix (Real-time Streaming Protocol) to stream audio over the network. A trailing forward slash is not permitted in the URL.

- Click **OK**.

Step 2 Add another Set Node in the script to configure the stream name.

- On the Set Variable tab of the Set Properties dialog box, select Call from the Object Type drop down and set the Variable to **PeripheralVariable<1>**.

The range for standard ICM Peripheral Variables is PeripheralVariable1 through PeripheralVariables10.

The screenshot shows the 'Set Properties (Read Only)' dialog box with the following configuration:

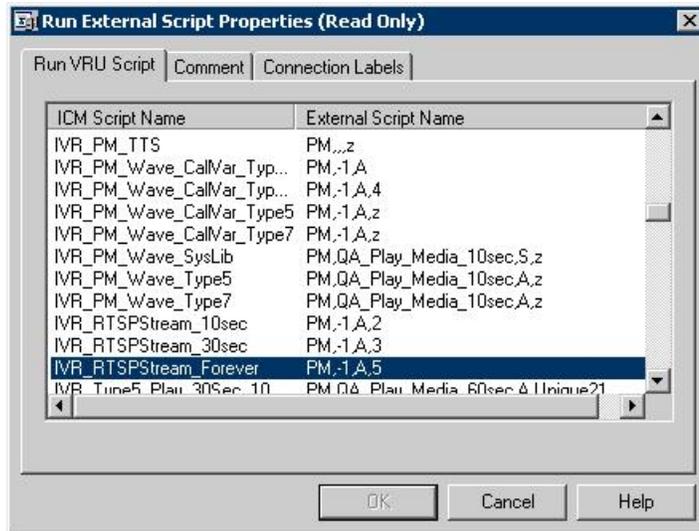
- Object type:** Call
- Object:** (No selection)
- Variable:** PeripheralVariable1
- Array index:** (empty)
- Value:** "african.rm"

- In the Value field, specify the stream name and click **OK**.

Note Stream names are case-sensitive.

Step 3 Add a Run External Script node to the workspace and double-click Run External Script.

The Run External Script Properties dialog box lists all of the Network VRU scripts that are currently configured.



Note In the example above, the IVR_RTSPStream_Forever script's external script name contains four parameters: PM, -1, A, 5. The second parameter, **-1**, instructs CVP to play the stream name declared in **PeripheralVariable1** (shown in Step 2). You must configure streaming audio following the steps outlined so that you may easily change the stream name within the Script Editor, if necessary.

You can also use the Run External Script node in the ICM Script Editor to configure ICM to failover to a new streaming server. For example, if you want to point to an alternate streaming server (IP address), use the X-path out of the Run External Script node to redefine the media_server ECC variable. In a failover situation, the script executes and the stream plays from the targeted streaming server and proceeds normally.

Step 4 From the Run VRU Script tab, select the ICM Script Name desired and click **OK**.

Step 5 Optionally, you can use the ICM Configuration Manager's Network VRU Script List tool's Attributes tab to configure the timeout value for the stream.

Configure the Configuration Param field parameter:

- In the RTSP Timeout field, enter a timeout value (in seconds).
 - The valid range is 0 - 43200 seconds.
 - If the value is set to 0 or a timeout value is not provided the stream does not end.

Step 6 Access the IOS device in global configuration mode and use the **rtsp client timeout connect** command to set the number of seconds the router waits before it reports an error to the Real-time Streaming Protocol (RTSP) server.

The range is 1 to 20. The recommended value is 10 seconds.

If the SIP Call with IVR Service is Terminated with **Reason Code: Q.850;Cause=38** then be sure that the network interface configuration is as follows:

```

ip route-cache same-interface
ip route-cache cef
ip route-cache
ip mroute-cache
no cdp enable

```

If specified, remove the following line from the network interface:

```
keepalive 1800
```

This issue arises if the Unified CVP loses network connectivity, then the VXML Server Gateway is not able to get information from the IVR Service, and as a result a code 38 rejection is generated in the Gateway logs.

Play Media Examples: Play Welcome Message

The following table shows some Network VRU Script configuration examples for Play Media.

Table 2: Network VRU Script Configuration Examples

Example	Field Name	Field Contents	Tells Unified CVP..
1	VRU Script Name	PM,Welcome	To use the Play Media (PM) micro-application to play the "Welcome.wav" Media file and accept the defaults for remaining settings. Note If no file extension is specified, .wav is assumed.
	Configuration Param	N	That Barge-in <i>is not</i> allowed.
2	VRU Script Name	pm,July,S	To use the Play Media (PM) micro-application to play the "July.wav" Media file, using the System (S) Media library.
	Configuration Param	Null (Accept default.)	That Barge-in <i>is</i> allowed.
3	VRU Script Name	PM,WebSite,,0	To use the Play Media (PM) micro-application to play the "Website.wav" Media file, using the default Media Type (Application library), and setting 0 as the Uniqueness value. Note A , (comma) indicates a skipped parameter. When a parameter is skipped, Unified CVP applies its default.
	Configuration Param	Null (Accept default.)	That Barge-in <i>is</i> allowed.

Example	Field Name	Field Contents	Tells Unified CVP...
4	VRU Script Name	PM,WebSite,,1	To use the Play Media (PM) micro-application to play the "Website.wav" Media file, using the default Media Type (Application library), and setting 1 as the Uniqueness value.
	Configuration Param	N	That Barge-in <i>is not</i> allowed.
5	VRU Script Name	PM,customer.vxml,V,1	To use the Play Media (PM) micro-application to the external VoiceXML file "customer.vxml", using the VoiceXML Media library, and setting 1 as the Uniqueness value.
	Configuration Param	Note Any barge-in setting is ignored when using external VoiceXML.	
6	VRU Script Name	PM	To use the Play Media (PM) micro-application and accept the defaults for remaining settings. Note If the user.microapp.inline_tts ECC contains a value, the PM micro-application will play its contents (for example, "Hello world").
	Configuration Param	N	That Barge-in <i>is not</i> allowed.
7	VRU Script Name	PM, -3, A	To use the Play Media (PM) micro-application, using the file listed in Call.PeripheralVariable3, acquiring the file from the Application (A) media library.
	Configuration Param	N	That Barge-in <i>is not</i> allowed.
8	VRU Script Name	PM, -7, V	To use the Play Media (PM) micro-application, Calls the external VoiceXML listed in Call.PeripheralVariable7, acquiring external VoiceXML (V) media library.
	Configuration Param	Note Any barge-in setting is ignored when using external VoiceXML.	

Example	Field Name	Field Contents	Tells Unified CVP..
9	VRU Script Name	PM, stream.rm	To use the Play Media (PM) micro-application to play "stream.rm" from a streaming audio server and accept the defaults for remaining settings.
	Configuration Param	N, 30	That Barge-in <i>is not</i> allowed, and the stream is configured to stop playing in 30 seconds.



Note Play Media sets the ECC variable **user.microapp.error_code** to zero, indicating success, if control proceeds out the Checkmark (success) branch of the Run External Script node. If control proceeds out the X (failure) branch, Play Media typically sets this variable to one of the codes listed in [Unified CVP Script Error Checking](#).

Play Data Micro-Application

The Play Data micro-application retrieves data from a storage area and plays it to the caller in a specific format, called a data play back type.

Some possible sources of the data to be played back:

- Information retrieved from a database look-up
- Information entered by the caller

Play Data and Data Storage

Before this micro-application can be called, you must specify the location of the play back data. You do this with a Script Editor Set node that points to one of the following storage areas:

- One of the standard ICM Peripheral Variables (PeripheralVariable1 through PeripheralVariables10).
- The **user.microapp.play_data** elements.

Configure Network VRU Script Settings for Play Data Micro-Application

Use the ICM Configuration Manager's Network VRU Script List tool's Attributes tab to specify parameters.



Note DTMF digit type-ahead is not supported by Play Media and Play Data micro-apps when executing in Comprehensive mode (Type 7). This feature is supported for Type 5 calls.

Voice barge-in is not supported by Play Media and Play Data micro-applications. However, DTMF barge-in is supported for these micro-applications.

If you are using integers that are larger than nine digits, enclose the value in quotation marks, so it will be treated as a string.

Procedure

Step 1

Configure VRU Script field parameters:

- **Micro-application type.** For Play Data, valid options are: **PD** or **pd**.
- **Data Playback Type.** The type of the data to be returned (“played”) to the caller. The valid options are:
 - **Number**
 - **Char** (character)
 - **Date**
 - **Etime** (elapsed time)
 - **TOD** (Time of Day)
 - **24TOD** (24-hour Time of Day)
 - **DOW** (Day of Week)
 - **Currency**

Note 24TOD and DOW data play back types are not supported when using TTS. Currency other than US dollar (USD) is not supported.

For more information about each of these playback types, including input format and output examples, see [Play Back Types for Voice Data](#).

- **Uniqueness value.** Optional. A string identifying a VRU Script Name as unique.

Step 2

Configure the Configuration Param field parameters:

- **Location of the data to be played** . The valid options are:
 - *null* (default) - If you leave this option empty, uses the ECC variable **user.microapp.play_data**.
 - A **number** representing a Call Peripheral Variable number (for example, a 1 to represent Call.PeripheralVariable1).

Note For more information on data location, see [Play Data and Data Storage](#).

- **Barge-in Allowed.** Specifies whether barge-in (digit entry to interrupt media playback) is allowed.

The valid options are:

- **Y** - (default) barge-in allowed
- **N** - barge-in not allowed

Note Voice barge-in is not supported by Play Media and Play Data micro-applications. However, DTMF barge-in is supported for these micro-applications.

Unified CVP deals with barge-in as follows: If barge-in *is not* allowed, the SIP/Gateway continues prompt play when a caller starts entering digits and the entered digits are discarded. If barge-in *is* allowed, the Gateway discontinues prompt play when the caller starts entering digits. See [Get Speech and External VoiceXML](#).

Barge-in works the same for ASR as DTMF. If the caller speaks during prompt play, the prompt play stops. Unlike DTMF input, ASR caller input is checked against the grammar that is defined. If a match is not found, an Invalid Entry error is generated and the caller input is deleted. Voice barge-in is not supported during a Play Media or Play Data script because there is not a grammar specified for these micro-applications.

• Time Format

Valid only for the time Data Playback types (Etime, TOD, 24TOD).

The available formats are:

- *null* - leave this option empty for non-time formats
- **HHMM** - default for time formats
- **HHMMSS** - includes seconds
- **HHMMAP** - includes am or pm; valid only for TOD

- **Type-ahead Buffer Flush** . The Cisco VoiceXML implementation includes a type-ahead buffer that holds DTMF digits collected from the caller. When the VoiceXML form interpretation algorithm collects user DTMF input, it uses the digits from this buffer before waiting for further input. This parameter controls whether the type-ahead buffer is flushed after the prompt plays out. A false value (default) means that the type-ahead buffer is not flushed after the prompt plays out. If the prompt allows barge-in, the digit that barges in is not flushed.

The valid options are:

- **Y** - flush the type-ahead buffer
- **N** - (default) do not flush the type-ahead buffer

Note This parameter is only applicable when using the Cisco IOS gateway with DTMF barge-in. This parameter is not applicable when using external VXML. This parameter is normally used when two or more PM and/or PD microapps are used in a loop in the ICM script (such as while in queue for an agent). If the PM and/or PD microapps are enabled for barge-in, one would set this parameter to **Y** to prevent an uncontrolled looping in the ICM script when the user barges in.

Play Back Types for Voice Data

Configuring how voice data is presented to a caller is an important part of setting up your Unified CVP IVR. The "Data Play Back Types" table below describes each type, along with sample valid values and formats for the supported locales when **not** using TTS:

- **en-us**. English (United States)
- **en-gb**. English (Great Britain)
- **es-mx**. Spanish (Mexico)
- **es-es**. Spanish (Spain)

Locale is selected by setting the **user.microapp.locale** variable.



Note For information about locale support when using TTS, check with your third-party vendor.

Any string of characters typically used in the language may need to be spoken back character by character (this includes special keyboard symbols and numbers). If a particular symbol is not used by a particular language, a string containing that symbol may be spelled out with a Play Data with Char data type.

For example, assume that an IVR application in the US (a locale of **en-us**) queries a database for an account owner's name and spells the name back to the caller. If the name pulled from the database was "Hänschen Walther," the media files that would need to be pulled from the Media Server would have been derived from a URL including the **en-us** locale. The symbol **ä** has a decimal value of 228, which is different than the symbol **a** which has a value of 97. It is the translator's task to record the proper word(s) for each symbol to be supported. For detailed information on character translation, refer to "[System Media Files](#)."



Note When using TTS, some output format may vary between Unified CVP playback types and TTS playback types. For example, TTS may pronounce a Play Data number "1234" as "twelve thirty four". When not using TTS, the output is "one thousand, two hundred, thirty four". Please check with your third-party vendor on TTS outputs for playback types.

Table 3: Data Play Back Types

Data Play Back Type	Description	Input Format	Output Examples (When Not Using TTS)
Number	Play the stored data as a number.	<p>#####.#####</p> <p>The leading minus (-) is optional and is played as “minus.”</p> <p>The whole number portion of the string can contain a maximum of 15 digits (for a maximum value of 999 trillion, 999 billion and so on).</p> <p>The decimal point is represented as a period (.) and played as “point.” It is optional if there is no floating portion.</p> <p>The floating point portion of the number is optional and can contain a maximum of six digits.</p> <p>Trailing zeros are played.</p>	<p>en-us and en-gb typical spoken form:</p> <ul style="list-style-type: none"> • -123 = “minus one hundred twenty three” • 35.67 = “thirty five point six seven” • 1234.0 = “one thousand, two hundred, thirty four point zero” <p>es-mx and es-es typical spoken form:</p> <ul style="list-style-type: none"> • -120 = “menos ciento veinte” • 10.60 = “diez coma seis cero” • 1,100 = “mil cien”
Char	Play the stored data as individual characters.	<p>All printable American National Standards Institute (ANSI) characters are supported.</p> <p>Note Code Page 1252 is ANSI standard. It contains ASCII (characters 0-127) and extended characters from 128 to 255</p>	<p>en-us and en-gb typical spoken form:</p> <ul style="list-style-type: none"> • abc123= “A, B, C, one, two, three” <p>es-mx and es-es typical spoken form:</p> <ul style="list-style-type: none"> • abc123 = “A, B, C, uno, dos, tres”

Data Play Back Type	Description	Input Format	Output Examples (When Not Using TTS)
Date	Play the stored data as a date.	YYYYMMDD, regardless of locale. YYYY options: the range of 1800 through 9999. MM options: the range of 01 through 12. DD options: the range of 01 through 31. Note The software does not validate the date (for example, 20000231 is valid and played accordingly). However, a failure occurs if any bounds are broken (for example, 34 for month).	en-us typical spoken form: <ul style="list-style-type: none"> • MMDDYYYY format: 20000114 = “January fourteenth, two thousand” en-gb typical spoken form: <ul style="list-style-type: none"> • DDMMYYYY format: 20000114 = “Fourteenth of January, two thousand” es-mx and es-es typical spoken form: <ul style="list-style-type: none"> • DDMMYYYY format: 20001012 = “doce octubre dos mil” Note All spoken forms use the proper grammar for the locale.
Etime (elapsed time)	Play the stored data as an amount of elapsed time.	HHMM or HHMMSS Maximum 99 hours, 59 minutes, 59 seconds Leading zeros are ignored.	en-us and en-gb typical spoken form: <ul style="list-style-type: none"> • HHMM format: 0830 = “eight hours thirty minutes” • HHMMSS format: 083020 = “eight hours, thirty minutes, twenty seconds” es-mx and es-es typical spoken form: <ul style="list-style-type: none"> • HHMM format: 0205 = “dos horas cinco minutos” • HHMMSSS format: 020101 = “dos horas un minuto un segundo”

Data Play Back Type	Description	Input Format	Output Examples (When Not Using TTS)
TOD (Time of Day)	Play the stored data as a time of day.	HHMM or HHMMSS 24 hour time HH options: 00 - 24 MM options: 00 - 59 SS options: 00 - 59	<p>en-us and en-gb typical spoken form:</p> <ul style="list-style-type: none"> • HHMM format: 0800 = “eight o’clock” 0830 = “eight thirty” 1430 = “two thirty” • HHMMSS format: 083020 = “eight thirty and twenty seconds” • HHMMAP format: 1430 = “two thirty p.m.” <p>es-mx and es-es typical spoken form:</p> <ul style="list-style-type: none"> • HHMM format: 0100 = “una a.m.” • HHMMAP format: 1203 = “doce y tres p.m.” • HHMMSS format: 242124 = “doce veintiuno a.m.”

Data Play Back Type	Description	Input Format	Output Examples (When Not Using TTS)
24TOD (24-hour Time of Day)	Play the stored data as military time.	<p>HHMM or HHMMSS 24 hour time.</p> <p>HH options: 00 - 24</p> <p>Note 24-hour time and military time may have a discrepancy as to valid hours. Unified CVP plays back the value 00 or 24 “as is.” The application developer is free to make alterations to the data passed to the micro-application, if so desired.</p> <p>MM options: 00 - 59</p> <p>SS options: 00 - 59</p> <p>Unified CVP validates the ranges as stated above. For example, if a time ends in 00 minutes (that is, 2300), one would say “hundred hours” (that is, “twenty-three hundred hours”). The range is 0000 (12 a.m.) through 2459 (after midnight) or 0059. 1300 equals 1 o’clock in the afternoon.</p> <p>Note The 24TOD play back type is not supported when using TTS.</p>	<p>en-us and en-gb typical spoken form:</p> <ul style="list-style-type: none"> • HHMM format: 0815 = “eight fifteen” 2330 = “twenty three thirty” 2300 = “twenty three hundred hours” • HHMMSS format: 233029 = “twenty three thirty and twenty nine seconds” <p>es-mx and es-es typical spoken form:</p> <ul style="list-style-type: none"> • HHMM format: 2121 = “veintiuno veintiuno” 2100 = “veintiún horas” <p>Note In Spanish, when a time ends in 00 minutes the spoken form is “hours,” not “hundred hours.”</p> <ul style="list-style-type: none"> • HHMMSS format: 050505 = “cinco y cinco y cinco segundos”

Data Play Back Type	Description	Input Format	Output Examples (When Not Using TTS)
DOW (Day of Week)	Play the stored data as a day of week.	An integer from 1 through 7 (1 = Sunday, 2 = Monday, et cetera). Note The DOW data play back type is not supported when using TTS.	en-us and en-gb typical spoken form: <ul style="list-style-type: none"> • 7 = “Saturday” es-mx and es-es typical spoken form: <ul style="list-style-type: none"> • 7 = “Sabado”

Data Play Back Type	Description	Input Format	Output Examples (When Not Using TTS)
Currency	Play the stored data as currency.	<p>Format is [-]15(X)[.2(Y)] where the minus sign is optional as well as the decimal point and the two digits after the decimal point. The whole number portion of the string can contain a maximum of 15 digits (for a maximum value of 999 trillion, 999 billion).</p> <p>Note No comma delimiters or currency symbols are recognized.</p> <p>Leading and trailing zeros are played. If a number does not have a decimal point, the “cents” portion of the amount will not be spoken. For example, the spoken form for the input 100 is “one hundred dollars.”</p> <p>The grammar rules apply to the currency, not the locale.</p> <p>Note The <code>user.microapp.currency</code> ECC variable contains the currency indicator (USD, CAD, EUR, et cetera).</p>	<p>USD (US dollar) typical spoken form:</p> <ul style="list-style-type: none"> • 15.05 = “fifteen dollars and five cents” • 3.00 = “three dollars and zero cents” <p>Note Unified CVP uses the <code>USD_dollar.wav</code> and <code>USD_dollars.wav</code> media files; the <code>dollar.wav</code> and <code>dollars.wav</code> used by ISN Version 1.0 are no longer installed.</p> <p>CAD (Canadian dollar) typical spoken form:</p> <ul style="list-style-type: none"> • 15.05 = “fifteen dollars and five cents” • 3.00 = “three dollars and zero cents” <p>EUR (Euro dollar) typical spoken form:</p> <ul style="list-style-type: none"> • 1.10 = “one point one zero euro” <p>GBP (Great Britain pound) typical spoken form:</p> <ul style="list-style-type: none"> • 1.10 = “one pound ten pence” <p>MXN (Mexican pesos) typical spoken form:</p> <ul style="list-style-type: none"> • 1.10 = “one peso and 10 centavos” <p>Note The default spoken form for a negative amount (for all currency types) is “minus <amount>.”</p>



Note 24TOD and DOW data play back types are not supported when using TTS. Also, currency other than US dollar (USD) is not supported.

Play Data Configuration Examples

The following table shows several configuration examples for Play Data.

Table 4: Play Data Configuration Examples

If the VRU Script Name field setting is...	It means...	If the Configuration Param field is...	It means...
PD,Number Note If you are using integers that are larger than nine digits, enclose the value in quotation marks, so it will be treated as a string.	PD - Use the Play Data micro-app. Number - Play back the data as a number.	empty	Play the data in the default ECC, user.microapp.play_data , as a number.
PD, Char	pd - Use the Play Data micro-app. Char - Play back the data as individual characters.	1	1 - Play the data in Call PeripheralVariable 1 as a character.
PD,Etime,0 Note If you are using integers that are larger than 9 digits, enclose the value in quotation marks, so it will be treated as a string.	PD - Use the Play Data micro-app. Etime - Play back the data as a Time.	1,,HHMM	1 - Play the data in Call PeripheralVariable 1 as an elapsed time. , - (Skipped parameter) Accept default setting (Y) HHMM - Play the time in HHMM format (for example, 8 hours, 30 minutes).
PD,Date	PD - Use the Play Data micro-app. Date - Play back the data as a Date.	1,N	1 - Play the data in Call Variable 1 as a date. N - No barge-in allowed.
PD,Currency	PD - Use the Play Data micro-app. Currency - Play back the data as a Currency.	4,N	4 - Play the data in Call Variable 4 s currency. N - No barge-in allowed.



Note Play Data sets the ECC variable **user.microapp.error_code** to zero, indicating success, if control proceeds out the Checkmark (success) branch of the Run External Script node. If control proceeds out the X (failure) branch, Play Data typically sets this variable to one of the codes listed in [Unified CVP Script Error Checking](#).

To enable Text-to-Speech (TTS) as the data play back source, you need to insert a Set node in the script prior to the Run External Script node setting **user.microapp.pd_tts** to "Y".

Get Digits Micro-Application

The Get Digits (GD) micro-application plays a media file and retrieves digits. For example, you could use Get Digits in an application that prompts a caller to enter a password.

Unified Customer Voice Portal passes the retrieved digits back to Unified ICME for further processing using the Caller-Entered Digits (CED) field in the ICM/IVR Messaging interface. (This is available in the Unified ICME script through the variable **Call.CallerEnteredDigits**).



Note In Customer Voice Portal, the Collected digits will only be cached for a CAPTURE microapp following the digit collection node. If a non-digit collect nodes follows a digit collect node, Customer Voice Portal's CED variable will get nulled out and **Call.CallerEnteredDigits** also get nulled out .

For example, if the script is **START > GD (Get Digit node) > CAPTURE NODE > PM (Play Media) > CAPTURE NODE**. In this case, user has to cache the collected data via ICM peripheral (or call) variables.

Here collected digits will be available only for first CAPTURE NODE. After PM node (non-digit collect node), the CED value is set to null. So second CAPTURE NODE or any node which require collected digits to process, will not be able access the collected digits through **Call.CallerEnteredDigits**.

Configure Network VRU Script Settings for Get Digits Micro-Application

Use the ICM Configuration Manager's Network VRU Script List tool's Attribute tab to specify parameters.

Procedure

Step 1 Configure VRU Script field parameters:

- **Micro-application type.** For Get Digits, valid options are: **GD** or **gd**.
- **Media File Name.** Name of the media file or external VoiceXML to be played (that is, the prompt file). The valid options are:
 - A file name (for instance, a .wav file).

Note The file name is case-sensitive.
- **null** - (default) If this field is empty, Unified CVP examines the contents of the **user.microapp.inline_tts** ECC variable. If this ECC variable contains a value, Unified CVP prompts using TTS. If the ECC is empty, no prompt is played.

- **-(number 1-10)** - Unified CVP plays the file in the corresponding Call.PeripheralVariable file. For example, entering -2 causes Unified CVP to look at Call.PeripheralVariable2.
- **Media Library Type** . Flag indicating the location of the media files to be played. The valid options are:
 - **A** - (default) Application
 - **S** - System
- Note** This value is ignored if using TTS.
- **Uniqueness value**. Optional. A string identifying a VRU Script Name as unique.

Step 2 Configure the Configuration Param field parameters:

- **Minimum Field Length**. Minimum number of digits expected from the caller. The valid options are: **1-32** (the default is **1**)
- **Maximum Field Length**. Maximum number of digits expected from the caller. The valid options are: **1-32** (the default is **1**).
- Note** For information about Maximum Field Length and the DTMF Termination Key, see [Get Digits and Digit Entry Completion](#).
- **Barge-in Allowed** . Specifies whether barge-in (digit entry to interrupt media playback) is allowed. The valid options are:
 - **Y** - (default) barge-in allowed
 - **N** - barge-in not allowed
- Note** Unified CVP deals with barge-in as follows: If barge-in *is* not allowed, the SIP/Gateway continues prompt play when a caller starts entering digits. If barge-in *is* allowed, the Gateway discontinues prompt play when the caller starts entering digits. See [Get Speech and External VoiceXML](#).
- **Inter-digit Timeout** . The number of seconds the caller is allowed between entering digits. If exceeded, the system times-out. The valid options are: **1-99** (the default is **3**).
- Note** This value is ignored if using ASR.
- **No Entry Timeout** . The number of seconds a caller is allowed to begin entering digits. If exceeded, the system times-out. The valid options are: **0-99** (the default is **5**).
- **Number of No Entry Tries**. Unified CVP repeats the “Get Digits” cycle when the caller does not enter any data after the prompt has been played. (Total includes the first cycle.) The valid options are: **1-9** (the default is **3**).
- **Number of Invalid Tries**. Unified CVP repeats the “Get digits” cycle when the caller enters invalid data (total includes the first cycle). The valid options are: **1-9** (default is **3**).
- **Timeout Message Override** . The valid options are:
 - **Y** - override the system default with a pre-recorded Application Media Library file

- **N** - (default) do not override the system default

Note This value is ignored if using TTS.

- **Invalid Entry Message Override.** The valid options are:

- **Y** - override the system default with a pre-recorded Application Media Library file.
- **N** - (default) do not override the system default

Note This value is ignored if using TTS.

For more information about Timeout and Invalid Entry Messages, see [System Media Files](#).

- **DTMF Termination Key.** A single character that, when entered by the caller, indicates that the digit entry is complete. The valid options are:

- **0-9**
- ***** (asterisk)
- **#** (pound sign, the default)
- **N** (No termination key)

Note For information about Maximum Field Length and the DTMF Termination Key, see [Get Digits and Digit Entry Completion](#).

This value is ignored if using ASR.

- **Incomplete Timeout.** The amount of time after a caller stops speaking to generate an invalid entry error because the caller input does not match the defined grammar. The valid options are: **0-99** (the default is 3).

Note This value is ignored when not using ASR. If the value is set to 0, the IVR Service treats the NoEntry Timeout as NoError.

Get Digits Configuration Examples

The following table shows several configuration examples for Get Digits for an application that prompts using .wav files and retrieves input through DTMF.

Table 5: Get Digits Configuration Examples

If the VRU Script Name field setting is...	It means...	If the Configuration Param field setting is...	It means...
GD>Password,A,0	GD - Use the Get Digits micro-app. Password - Play the Media file named "Password.wav." A - Application Media Library. 0 - Uniqueness value.	6,12	6 - Minimum field length 12 - Maximum field length Accept defaults for all other settings.
GD>Password,A,1	gd - Use the Get Digits micro-app. Password - Play the Media file named "Password.wav." A - Application Media Library. 1 - Uniqueness value.	6,12,N,3,5,2,2,N,Y,#	6 - Minimum field length 12 - Maximum field length N - No barge-in allowed 3 - Inter-digit Timeout (seconds) 5 - No Entry Timeout (seconds) 2 - Number of no entry tries 2 - Number of invalid tries N - Timeout Msg Override Y - Invalid Entry Msg Override # - DTMF Termination key
Note The two examples above both play the Password.wav file ("Please enter your password followed by the pound sign.") and collect digits. They differ in that the first example accepts most of the default settings available through the Configuration Param field; the second field does not.			
GD,ssn	GD - Use the Get Digits micro-app. ssn - Play the Media file named "ssn.wav."	9,9,	9 - Minimum field length 9 - Maximum field length Accept defaults for all other settings.

If the VRU Script Name field setting is...	It means...	If the Configuration Param field setting is...	It means...
GD	<p>GD - Use the Get Digits micro-app.</p> <p>Since no Media field settings appear after GD, Unified CVP examines the contents of the user.microapp.inline_tts ECC variable. If this ECC variable contains a value - for example, "What is your account number?" - Unified CVP prompts using TTS.</p> <p>Note If the user.microapp.inline_tts is empty, no prompt is played.</p> <p>In turn, if the user.microapp.input_type ECC variable is D, Unified CVP is set to process any DTMF input the customer supplies.</p>	6,12,N	<p>6 - Minimum field length</p> <p>12 - Maximum field length</p> <p>N - No barge-in allowed</p> <p>Accept defaults for all other settings.</p>
<p>Note Type-ahead can only be used with the Get Digits micro-application when user.microapp.input_type is set to D. See Get Speech and External VoiceXML.</p>			
GD, -4, S	<p>gd - Use the Get Digits micro-app</p> <p>-4 - Calls the file specified in Call.PeripheralVariable4</p> <p>S - Acquires the file from the System media library</p>	6,12,	<p>6 - Minimum field length</p> <p>12 - Maximum field length</p> <p>Accept defaults for all other settings</p>

The following table shows several configuration examples for Get Digits for an ASR/TTS application.

Table 6: Get Digits Configuration Examples

If ...	It means ...	If the Configuration Param field setting is...	It means ...
<p>The user.microapp.inline_tts ECC variable contains “What is your account number?”</p> <p><i>and</i></p> <p>user.microapp.input_type contains: D (DTMF)</p> <p><i>and</i></p> <p>The VRU Script Name field contains: GD</p>	<p>Use the Get Digits micro-app to play the contents of the ECC variable and collect DTMF input.</p>	<p>6, 12, N,3,5,2,2,N,Y,#</p>	<p>6 - Minimum field length 12 - Maximum field length N - No barge-in allowed 3 - Inter-digit Timeout (seconds) 5 - No Entry Timeout (seconds) 2 - Number of no entry tries 2 - Number of invalid tries N - Timeout Msg Override Y - Invalid Entry Msg Override # - DTMF Termination key</p>
<p>The user.microapp.inline_tts ECC variable contains “What is your account number?”</p> <p><i>and</i></p> <p>user.microapp.input_type contains: B (the default, both DTMF and voice)</p> <p><i>and</i></p> <p>The VRU Script Name field contains: GD</p>	<p>Use the Get Digits micro-app to play the contents of the ECC variable and collect either voice or DTMF input.</p>	<p>6,12,N,,,,4</p>	<p>6 - Minimum field length 12 - Maximum field length N - No barge-in allowed ,,,, - Accept defaults for Inter-digit Timeout (seconds), No Entry Timeout (seconds), Number of no entry tries, Number of invalid tries, Timeout Msg Override, Invalid Entry Msg Override, DTMF Termination key 4 - Incomplete timeout</p>
<p>The user.microapp.inline_tts ECC variable contains “What is your account number?”</p> <p><i>and</i></p> <p>user.microapp.input_type contains: B (the default, both DTMF and voice)</p> <p><i>and</i></p> <p>The VRU Script Name field contains: GD</p>	<p>Use the Get Digits micro-app to play the contents of the ECC variable and collect DTMF or voice input.</p>	<p>6,12,N</p>	<p>6 - Minimum field length 12 - Maximum field length N - No barge-in allowed Accept defaults for all other settings.</p>
<p>Note Type-ahead can only be used with the Get Digits micro-application when user.microapp.input_type is set to D. See Get Speech and External VoiceXML.</p>			



Note Get Digits sets the ECC variable `user.microapp.error_code` to zero, indicating success, if control proceeds out the Checkmark (success) branch of the Run External Script node. If control proceeds out the X (failure) branch, Get Digits typically sets this variable to one of the codes listed in [Unified CVP Script Error Checking](#).

Get Digits and Digit Entry Completion

Unified CVP tests GD digit entry input against several conditions to determine whether digit entry is complete.

Unified CVP considers digit entry to be complete if the caller enters any of the following:

- The maximum allowable number of digits (when terminator key is not used).
- The maximum number of digits, excluding a terminator key.
- Less than the maximum number of digits, followed by the terminator key.
- Less than the maximum number of digits and exceeding the inter-digit timeout.
- Nothing and reaching the no entry timeout.



Caution It is important that you set up your Unified ICME script to test for all the scenarios mentioned below.

If Digit Entry Input Is Complete

After digit-entry input is complete, Unified CVP validates the digit string to determine if it is \geq (greater than or equal to) the minimum length and \leq (less than or equal to) the maximum length.

In variable-length data entry, the Maximum Field Length value does not accommodate the termination key. For example, if a GD micro-application is configured to accept a password that is between 6 and 12 digits long and digit-entry completion is indicated through a termination key (or a timeout), the Minimum Field Length setting would be 6, the Maximum Field Length setting would be 12, and the DTMF Termination Key is defined as a single character.

Before passing the result back to the IVR Service, SIP Service discards the termination key (only the password digits are included in the CED returned to Unified ICME).



Note In this example, if the 13th digit is entered without reaching the interdigit timeout and the 13th digit is not the terminator key, the extra digits are buffered by the gateway VXML browser and will be consumed by the next digit collecting node (for example: GD or Menu micro-app).

This type-ahead behavior is described online in the Type-ahead Support section of the [Cisco VoiceXML Programmer's Guide](#).

After validating the digit string, Unified CVP does the following:

- If the string is valid, Unified CVP stores the digit string (not including the terminator key) in the `Call CallerEnteredDigits` variable, exits the node through the Checkmark (success) branch, and returns control to Unified ICME software.

- If the string is not valid, Unified CVP considers it an invalid entry and does the following:
 - If the Number of Invalid Entry Tries value is not reached, Unified CVP plays an error message and re-plays the original prompt.
 - If the Number of Invalid Entry Tries value is reached, Unified CVP stores the last-entered digit string in the Call.CallerEnteredDigits variable, exits the node through the X (failure) branch, sets the **user.microapp.error_code** ECC variable to **16** (Reached Maximum Invalid Tries), and returns control to Unified ICME.

If No Entry Timeout Occurs

If the caller does not enter input and No Entry Timeout period is exceeded, the following happens:

- If the Number of No Entry Tries value has not been reached, Unified CVP plays the “no entry” error message and re-plays the original prompt.
- If the Number of No Entry Tries value has been reached, Unified CVP exits the node through the X (failure) branch, sets the Call.CallerEnteredDigits variable to NULL, the **user.microapp.error_code** ECC variable to **17** (Reached Maximum No Entry Tries), and returns control to Unified ICME.

Menu Micro-Application

This micro-application plays a menu media file and retrieves a defined digit. (Menu is similar to the Get Digit micro-application except that it only accepts one digit, which it checks for validity.)

Unified CVP passes the retrieved digit back to Unified ICME for further processing using the Caller-Entered Digits (CED) field in the ICM/IVR Messaging interface.

Configure Network VRU Script Settings for the Menu Micro-Application

Use the ICM Configuration Manager’s Network VRU Script List tool’s Attribute tab to specify parameters.

Procedure

-
- Step 1** Configure VRU Script field parameters:
- **Micro-application type** . For Menu, valid options are: **M** or **m**.
 - **Media File Name**. Name of the media file or external VoiceXML to be played (that is, the prompt file). The valid options are
 - A file name (for instance, a .wav file)

Note The file name is case-sensitive.
 - **null** - (default) If this field is empty, Unified CVP examines the contents of the **user.microapp.inline_tts** ECC variable. If this ECC variable contains a value, Unified CVP prompts using TTS. If the ECC is empty, no prompt is played.
 - **-(number 1-10)** - Unified CVP plays the file in the corresponding Call.PeripheralVariable file. For example, entering -2 causes Unified CVP to look at Call.PeripheralVariable2.

- **Media Library Type** . Flag indicating the location of the media files to be played. The valid options are:

- **A** - (default) Application
- **S** - System

Note This value is ignored if using TTS.

- **Uniqueness value**. Optional. A string identifying a VRU Script Name as unique.

Step 2 Configure the Configuration Param field parameters:

- A list of **menu choices** . The valid options are:

- **0-9**
- * (asterisk)
- # (pound sign)

Formats allowed include:

- Individual options delimited by a / (forward slash)
- Ranges delimited by a - (hyphen) with no space

- **Barge-in Allowed** . Specifies whether barge-in (digit entry to interrupt media playback) is allowed.

The valid options are:

- **Y** - (default) barge-in allowed
- **N** - barge-in not allowed

Note Unified CVP deals with barge-in as follows: If barge-in *is not* allowed, the Gateway continues prompt play when a caller starts entering digits. If barge-in *is* allowed, the Gateway discontinues prompt play when the caller starts entering digits. See [Get Speech and External VoiceXML](#).

- **No Entry Timeout** . The number of seconds a caller is allowed to begin entering digits. If exceeded, the system times-out. The valid options are: **0-99** (the default is **5**).
- **Number of No Entry Tries**. Unified CVP repeats the "Menu" cycle when the caller does not enter any data after the prompt has been played. (Total includes the first cycle.) The valid options are: **1-9** (the default is **3**).
- **Number of Invalid Tries** . Unified CVP repeats the prompt cycle when the caller enters invalid data. (Total includes the first cycle.) The valid options are: **1-9** (the default is **3**).
- **Timeout Message Override**. The valid options are:
 - **Y** - override the system default with a pre-recorded Application Media Library file
 - **N** - (default) do not override the system default
- **Invalid Entry Message Override** . The valid options are:
 - **Y** - override the system default with a pre-recorded Application Media Library file

- **N** - (default) do not override the system default

Note For more information about Timeout and Invalid Entry Messages, refer to "[System Media Files](#)."

Menu Configuration Examples

The following table shows several configuration examples for Menu for use in an ASR/TTS application:

Table 7: Menu Configuration Examples - ASR/TTS Application

If...	It means...	And, if the Configuration Param field contains...	It means...
<p>The user.microapp.inline_tts ECC variable contains "Press 1 for Sales and 2 for Support."</p> <p>and</p> <p>user.microapp.input_type contains: D (DTMF)</p> <p>and</p> <p>The VRU Script Name field contains: M</p>	<p>Use the Menu micro-app to play the contents of the user.microapp.inline_tts ECC variable and collect DTMF input.</p>	<p>1-2,Y,4,3</p>	<p>1-2 - Accept the DTMF digits 1 and 2.</p> <p>Y - Barge-in allowed.</p> <p>4 - No Entry Timeout value (in seconds).</p> <p>3 - Number of no entry tries allowed.</p>
<p>The user.microapp.input_type ECC variable contains: D (DTMF)</p> <p>and</p> <p>The VRU Script Name field contains: M,SalesService,A</p>	<p>Use the Menu micro-app to play the media file named "SalesService.wav" (which is located in the Application Media library) and collect DTMF input.</p>	<p>1-2,N,4,3,2,Y,Y</p>	<p>1-2 - Accept the numbers 1 and 2.</p> <p>N - No barge-in allowed.</p> <p>4 - No Entry Timeout value (in seconds).</p> <p>3 - Number of no entry tries allowed.</p> <p>2 - Number of invalid tries allowed.</p> <p>Y - Allow Timeout Msg Override.</p> <p>Y - Allow Invalid Entry Msg Override).</p>

If...	It means...	And, if the Configuration Param field contains...	It means...
<p>The user.microapp.inline_tts ECC variable contains "Press or Say 1 for Sales and 2 for Support."</p> <p>and</p> <p>user.microapp.input_type contains: B (the default, both DTMF and voice)</p> <p>and</p> <p>The VRU Script Name field contains: M</p>	<p>Use the Menu micro-app to play the contents of the user.microapp.inline_tts ECC variable and collect either DTMF or voice input.</p>	<p>1-2,Y,4,3</p>	<p>1-2 - Accept the DTMF digits 1 and 2.</p> <p>Y - Barge-in allowed.</p> <p>4 - No Entry Timeout value (in seconds).</p> <p>3 - Number of no entry tries allowed.</p>
<p>The user.microapp.inline_tts ECC variable contains "Press 1 for Sales and 2 for Support."</p> <p>and</p> <p>user.microapp.input_type contains: B (the default, both DTMF and voice)</p> <p>and</p> <p>The VRU Script Name field contains: M</p>	<p>Use the Menu micro-app to play the contents of the user.microapp.inline_tts ECC variable and collect DTMF or voice input.</p>	<p>The Configuration Param field contains: 1-2,Y,4,3</p>	<p>1-2 - Accept the DTMF digits 1 and 2.</p> <p>Y - Barge-in allowed.</p> <p>4 - No Entry Timeout value (in seconds).</p> <p>3 - Number of no entry tries allowed.</p>
<p>Note Type-ahead can <i>only</i> be used with the Menu micro-application when user.microapp.input_type is set to D. See Get Speech and External VoiceXML.</p>			

The following table shows several configuration examples for Menu for use in an application where input type is DTMF.

Table 8: Menu Configuration Example - DTMF Application

If the VRU Script Name field setting is...	It means...	If the Config Param setting is...	It means...
M,Banking	<p>M - Use the Menu micro-app.</p> <p>Banking - Play the Media file named "Banking.wav."</p> <p>Note This file may contain a message such as: "For Checking, press 1. For Savings, press 2. For Money Market, press 3."</p>	1-3	<p>1-3 - Accept numbers 1, 2, 3. Accept all other defaults (No Entry Timeout, Number of no entry tries, Number of invalid tries, Timeout Msg Override, Invalid Entry Msg Override).</p>
M,Main_Menu	<p>M - Use the Menu micro-app.</p> <p>Main_Menu - Play the Media file called "Main_Menu.wav."</p> <p>Note This file may contain a message such as: "For information or transactions on checking, press 1. For savings or club accounts, press 2. For other information, press 0. If you know your party's extension, press 9."</p>	0-2/9,,4,2,2	<p>0-2/9 - Accept numbers 0, 1, 2, and 9.</p> <p>, (Skipped parameter) - Accept the default barge-in setting (Y).</p> <p>4 - No Entry Timeout value (in seconds).</p> <p>2 - Number of no entry tries allowed.</p> <p>2 - Number of invalid tries allowed.</p> <p>Accept all other defaults (Timeout Msg Override, Invalid Entry Msg Override).</p>

If the VRU Script Name field setting is...	It means...	If the Config Param setting is...	It means...
M,-2,S	<p>M - Use the Menu micro-app.</p> <p>-2 - Plays the file specified in Call.PeripheralVariable2.</p> <p>S - Acquires the file from the System media library.</p>	1-3	1-3 - Accept numbers 1, 2, 3. Accept all other defaults (No Entry Timeout, Number of no entry tries, Number of invalid tries, Timeout Msg Override, Invalid Entry Msg Override).



Note Menu sets the ECC variable **user.microapp.error_code** to zero, indicating success, if control proceeds out the Checkmark (success) branch of the Run External Script node. If control proceeds out the X (failure) branch, Menu typically sets this variable to one of the codes listed in [Unified CVP Script Error Checking](#).

Menu and Digit Entry Completion

Unified CVP tests Menu digit entry input against two conditions to determine whether digit entry is complete:

- If a caller enters a digit, Unified CVP checks whether the digit is within the set of valid digits for this menu.
- If a caller does not enter a digit, Unified CVP checks whether the No Entry Timeout value has been reached.



Caution It is important that you set up your Unified ICME script to test for all the scenarios mentioned below.

Digit Entry Completion

After a caller enters a digit, Unified CVP validates the digit against the list of valid menu options that were defined through ICM Configuration Manager. Then Unified CVP does the following:

- If the digit is valid, Unified CVP stores the digit in the Call.CallerEnteredDigits variable, exits the node through the Checkmark (success) branch, and returns control to Unified ICME.
- If the digit is not valid, Unified CVP considers it an invalid entry and does the following:
 - If the Number of Invalid Entry Tries value *has not* been reached, Unified CVP plays the "invalid message" file and re-plays the menu prompt.
 - If the Number of Invalid Entry Tries value has been reached, Unified CVP stores the last-entered invalid digit in the **user.microapp.caller_input** variable, exits the node through the X (failure) branch, sets the **user.microapp.error_code** ECC variable to **16** (Reached Maximum Invalid Tries), and returns control to Unified ICME.

If No Entry Timeout Occurs

If the caller does not enter a digit within the No Entry Timeout period:

- If the Number of No Entry Tries value is reached, Unified CVP plays the "no entry" error message and re-plays the menu prompt.
- If the Number of No Entry Tries value has been reached, Unified CVP exits the node through the X (failure) branch, sets the Call.CallerEnteredDigits variable to NULL, the **user.microapp.error_code** ECC variable to **17** (Reached Maximum No Entry Tries), and returns control to Unified ICME.

Get Speech Micro-Application

The Get Speech (GS) micro-application collects input that can be DTMF-only, Speech, or both input modes, after prompting a caller. The prompt can be generated by a media file or a TTS source.



Note The Get Speech (GS) micro-application collects voice and DTMF input from the caller. Get Speech supports SRGS and built-in grammars with the exception of the "Digit" grammar which is handled by GetDigit. Use the ICM Configuration Manager's Network VRU Script List tool's Attribute tab to specify parameters. The prompt can be generated by a media file or a TTS source.

Unified CVP passes the input back to Unified ICME for further processing using the **user.microapp.caller_input** ECC variable.

Get Speech and Grammar Specification

There are three ways to specify a grammar in the Get Speech micro-application:

- Include a **Type of Data to Collect** setting in the Get Speech Configuration Param field for built-in grammars such as dates and numbers. If the "Type of Data to Collect" setting is specified, the other grammar options are not used by the IVR Service. Conversely, if you do not specify a "Type of Data to Collect" setting, then you must include either an inline or external grammar.
- Include an external grammar file name in the Get Speech Configuration Param field's "External Grammar File Name" setting.
- Include a list of inline grammar choices in the **user.microapp.grammar_choices** ECC variable. These grammar choices only used if a "Type of Data to Collect" or "External Grammar File Name" setting is not specified.



Note One of these grammar options must be used for each micro-application. If no grammar option is specified, an Invalid Config Param error is sent back to Unified ICME.



Note If you are using an external grammar, be sure to follow the instructions provided by your third-party vendor.

To write an external grammar file, see [External VoiceXML File Contents](#).



Note For the following table, the Configuration Param field is not used if you are using external VoiceXML.

Configure Network VRU Script Settings for the Get Speech Micro-Application

Use the ICM Configuration Manager's Network VRU Script List tool's Attribute tab to specify parameters.

Procedure

Step 1 Configure VRU Script field parameters:

- **Micro-application type.** For Get Speech, valid options are: **GS** or **gs**.
- **Media File Name.** Name of the media file or external VoiceXML to be played (that is, the prompt file). The valid options are:
 - A file name (for instance, a .wav file)
 - Note** The file name is case-sensitive.
 - **null** - (default) If this field is empty, Unified CVP examines the contents of the **user.microapp.inline_tts** ECC variable. If this ECC variable contains a value, Unified CVP prompts using TTS. If the ECC is empty, no prompt is played.
 - **-(number 1-10)** - Unified CVP plays the file in the corresponding Call.PeripheralVariable file. For example, entering -2 causes Unified CVP to look at Call.PeripheralVariable2.

Note If you use the -(number 1-10) option and set the Media Library Type to "V," Unified CVP plays the external VoiceXML file specified in the corresponding Call.PeripheralVariable. If you set the value to - (no value) and set the Media Library Type to "A" or "S", the IVR Service creates VoiceXML without a media prompt.

- **Media Library Type.** Flag indicating the location of the media files to be played. The valid options are:
 - **A** - (default) Application
 - **S** - System
 - **V** - External VoiceXML. Refer to ["Get Speech and External VoiceXML."](#)

Note This value is ignored if using TTS.

- **Uniqueness value.** Optional. A string identifying a VRU Script Name as unique.

Step 2 Configure the Configuration Param field parameters:

Note This field does not apply if you are using external VoiceXML. For example, if you are using external VoiceXML, all Configuration Param settings (such as barge-in) will be allowed whether they're set to Y or N.

- **Type of Data to Collect.** A flag indicating the location of the media files to be played. The valid options are:

- **null** - (default) Leave this option empty if you will be specifying an External Grammar File Name setting.
 - **boolean** - Affirmative and negative phrases appropriate to the current locale.
 - **date** - Phrases that specify a date, including a month, days and year.
 - **currency** - Phrases that specify a currency amount.
 - Note** Nuance 8.5 ASR does not support negative currencies in its built-in grammar of datatype "currency."
 - **number** - Phrases that specify numbers. (For example, "one hundred twenty-three.")
 - **time** - Phrases that specify a time, including hours and minutes.
 - Note** For information about the format of the currency data returned to Unified ICME in the `user.microapp.caller_input` ECC variable, refer to ["Get Speech Data Format."](#)
- **External Grammar File Name.** The name of the grammar file that holds the grammar definition for the ASR. The valid options are:
- **null** - (default) Leaving this option empty implies that an inline grammar, as given in the Type of Data to Collect setting, is used.
 - A grammar file name. The Gateway retrieves the grammar file from a Web Server using HTTP.
 - Note** The file name is case-sensitive.
- Note** For more information about the "Type of Data to Collect" and "External Grammar" settings, see [Get Speech and Grammar Specification](#).
- **Barge-in Allowed.** Specifies whether barge-in (digit entry to interrupt media playback) is allowed. The valid options are:
- **Y** - (default) barge-in allowed
 - **N** - barge-in not allowed
- Note** Unified CVP deals with barge-in as follows: If barge-in *is not* allowed, the Gateway continues prompt play when a caller starts entering input. If barge-in *is* allowed, the Gateway discontinues prompt play when the caller starts entering input. See [Get Speech and External VoiceXML](#).
- **No Entry Timeout** . The number of seconds a caller is allowed to begin entering digits. If exceeded, the system times-out. The valid options are: **0-99** (the default is **5**).
 - **Number of No Entry Tries** Unified CVP repeats the "Get Digits" cycle when the caller does not enter any data after the prompt is played. (Total includes the first cycle.) The valid options are: **1-9** (default is **3**).
 - **Number of Invalid Tries** Unified CVP repeats the prompt cycle when the caller enters invalid data. (Total includes the first cycle.) The valid options are: **1-9** (default is **3**).
 - **Timeout Message Override** . The valid options are:
 - **Y** - override the system default with a pre-recorded Application Media Library file
 - **N** - (default) do not override the system default

Note This value is ignored if using TTS.

• **Invalid Entry Message Override.** The valid options are:

- **Y** - override the system default with a pre-recorded Application Media Library file.
- **N** - (default) do not override the system default

Note This value is ignored if using TTS.

Note For more information about Timeout and Invalid Entry Messages, see [System Media Files](#).

• **Incomplete Timeout.** The amount of time after a caller stops speaking to generate an invalid entry error because the caller input does not match the defined grammar. The valid options are: **0-99** (default is 3).

Note This value is ignored when not using ASR. If the value is set to 0, the IVR Service treats the NoEntry Timeout as NoError

• **Inter-digit Timeout** . The number of seconds the caller is allowed between entering DTMF key presses. If exceeded, the system times-out. The valid options are: **1-99** (the default is 3).

Note This value is ignored if using ASR.

• **Pass FTP Information** Specifies whether to pass FTP server information to the VXML Server. This option is only useful if the VXML Server application uses the FTP_Client Element and the FTP server information is already configured using the Operations Console. Valid options are:

- **Y** - Pass FTP server information to the VXML Server as VXML Server session variables.
- **N** - (default) Do not pass FTP server information.

If the **Pass FTP Information** parameter is set, the following information is passed:

- **ftpServer** - A space separated string of FTP servers. For example, `ftp_host1|21|username|password ftp_host2`. Everything is optional except the host name. See FTP_Client Element settings located in the *Element Specifications for Cisco Unified CVP VXML Server and Cisco Unified Call Studio* guide for more information.
- **ftpPath** - path on the FTP server. By default, this path is formed from the content of the ECC variable `user.microapp.locale` concatenated with path separator (/) and the content of the ECC variable `user.microapp.app_media_lib`. One exception is if the value of `user.microapp.app_media_lib` is `..`, then `app` is used instead. An example of a path is: `en-us/app`

Get Speech Configuration Examples

The following table shows several configuration examples for Get Speech.

Table 9: Get Speech Configuration Examples

If...	It means...	And, if...	It means...
<p>The user.microapp.inline_tts ECC variable contains “What is your account value”</p> <p>and</p> <p>user.microapp.input_type contains: B (the default, both DTMF and voice)</p> <p>and</p> <p>The VRU Script Name field contains: GS</p>	<p>Use the Get Speech micro-app to play the contents of the user.microapp.inline_tts ECC variable and collect account balance in voice or DTMF input, which it passes in the user.microapp.caller_input ECC variable.</p>	<p>The Configuration Param field contains: Currency,,N,5,2,1</p>	<p>Currency - Collect a string of data in currency format</p> <p>, - Accept the default External Grammar File Name setting (empty).</p> <p>Note You accept the default because you are specifying a Type of Data to Collect parameter (Currency).</p> <p>N - No barge-in allowed.</p> <p>5 - No Entry Timeout (seconds)</p> <p>2 - Number of no entry tries</p> <p>1 - Number of invalid tries</p>
<p>The user.microapp.inline_tts ECC variable contains “What department do you wish to speak to”</p> <p>and</p> <p>user.microapp.input_type contains: B (the default, both DTMF and voice)</p> <p>and</p> <p>user.microapp.grammar_choices contains Sales/ Customer Service/ Help Desk</p> <p>and</p> <p>The VRU Script Name field contains: GS,Department,A</p>	<p>Use the Get Speech micro-app play the contents of the user.microapp.inline_tts ECC variable and collect the answer, which will be passed in either voice or DTMF format in the user.microapp.caller_input ECC variable</p>	<p>The Configuration Param field contains: ,, N</p>	<p>, - Accept the default Type of Data to Collect parameter (empty)</p> <p>Note You accept the default Type of Data to Collect parameter because you are specifying a value in the External Grammar File Name parameter.</p> <p>, - Accept the default External Grammar File Name setting (empty) and use the grammar in the user.microapp.grammar_choices ECC variable.</p> <p>Note This is an inline grammar example. Each option in the list of choices specified in the user.grammar_choices ECC must be delimited by a forward slash (/).</p> <p>N - No barge-in allowed.</p>

If...	It means...	And, if...	It means...
<p>The user.microapp.inline_tts ECC variable contains “What is your name”</p> <p>and</p> <p>user.microapp.input_type contains: B (the default, both DTMF and voice)</p> <p>and</p> <p>user.microapp.media_server contains <code>http://grammars.com</code></p> <p>and</p> <p>user.microapp.app_media_lib contains Boston</p> <p>and</p> <p>The VRU Script Name field contains: GS,YourName,A</p>	<p>Use the Get Speech micro-app play the contents of the user.microapp.inline_tts ECC variable and collect the answer, which will be passed in either voice or DTMF format in the user.microapp.caller_input ECC variable.</p>	<p>The Configuration Param field contains:</p> <p>,customers.grxml,N</p>	<p>, - Accept the default Type of Data to Collect parameter (empty)</p> <p>Note You accept the default Type of Data to Collect parameter because you are specifying a value in the External Grammar File Name setting parameter.</p> <p>customers.grxml - Use the grammar in this file</p> <p>Note This is an external grammar file example. For details on writing an external grammar file, refer to External VoiceXML File Contents</p> <p>N - No barge-in allowed.</p>



Note Get Speech sets the ECC variable **user.microapp.error_code** to zero, indicating success, if control proceeds out the Checkmark (success) branch of the Run External Script node. If control proceeds out the X (failure) branch, Get Speech typically sets this variable to one of the codes listed in [Unified CVP Script Error Checking](#).

Get Speech and DTMF Input Collection

Contrary to its name, the Get Speech micro-application can also be used to collect DTMF input. For certain grammars, the caller could type a number, time, or currency rather than saying it.

Although the Get Digits micro-application is capable of providing the same type of functionality, it does not allow for validation at collection time. If a caller inputs 2 5 0 0 in response to a Get Speech prompt prompting the caller to enter a time, the Get Speech micro-application detects that “twenty-five hundred hours” is an invalid entry. With the Get Digits micro-application, this kind of validation is done using additional Script Editor nodes.



Note The caller cannot mix DTMF and speech in a single input, even if both are enabled. Once the caller starts talking, they cannot key-in characters, and vice versa.

The following table lists the rules associated with using DTMF collection in the Get Speech micro-application.

Table 10: DTMF Rules for Get Speech

Type of Data To Collect (as specified in the Config Params)	Allows DTMF Input?	DTMF Rules
boolean	Yes	Valid DTMF inputs are: 1 (Yes) and 2 (No).
date	Yes	Valid DTMF inputs are: four digits for the year, followed by two digits for the month, and two digits for the day. *
currency	Yes	For DTMF input, the * (asterisk) key represents the decimal point.*
number	Yes	Valid DTMF input includes positive numbers entered using digits and the * (asterisk) key to represent a decimal point.
time	Yes	Since there is no DTMF convention for specifying AM/PM, in the case of DTMF input, the result will always end with h or ?
External Grammars	No	None
Inline Grammars	No	None



Note Regardless of the collection type ("voice" or "DTMF"), caller input from Get Speech is always written to the `user.microapp.caller_input` ECC variable.

Get Speech Data Format

The data type determines the format of the information returned to Unified ICME in the `user.microapp.caller_input` ECC variable:

- **Boolean.** Returned to Unified ICME as "true" or "false."
- **Date.** Returned to Unified ICME as a fixed-length date string with the format `yyyymmdd` where `yyyy` is the year, `mm` is the month, and `dd` is the day.
- **Currency.** Returned to Unified ICME as a string with the format `UUUmmm.mm` or `mmm.mm`, where `UUU` is the three-character currency indicator (for example, USD), and `mmm.mm` is the currency amount with a decimal point.



Note Whether `UUU` is used depends on the ASR capabilities and on whether the caller said it unambiguously (for example, "dollar" and "dinar" are ambiguous, and so the `UUU` segment is not included in the return value).

- **Number.** Returned to Unified ICME as a string of digits from 0 to 9 which can optionally include a decimal point and/or a plus or minus sign as a prefix to indicate that the number is positive or negative.

- **Time.** Returned to Unified ICME as a five-character string in the format hhmmx, where hh is hours, mm is minutes and x is one of the following:
 - a** - AM
 - p** - PM
 - ?** - unknown/ambiguous

Get Speech and Entry Completion

The ASR Engine tests Get Speech input entry against two conditions to determine which entry is complete:

- If a caller enters input, the ASR Engine checks whether the input is within the set of valid grammar for this script.
- If a caller *does not* enter input, the ASR Engine verifies that the No Entry Timeout value has been reached.



Caution It is important that you set up your Unified ICME script to test for the scenarios that follow.

Input Entry Is Complete

After a caller enters input, the ASR Engine validates the input against the valid grammar set that was defined using the Set node to define the `user.microapp.grammar_choices` ECC variable.

Then Unified CVP does the following:

- If the input is valid, Unified CVP stores the input in the `user.microapp.caller_input` ECC variable, exits the node through the Checkmark (success) branch, and returns control to Unified ICME.
- If the input is not valid, Unified CVP treats it as an invalid entry and does the following:
 - If the Number of Invalid Entry Tries value *has not* been reached, Unified CVP plays the "invalid message" file and re-plays the menu prompt.
 - If the Number of Invalid Entry Tries value *has* been reached, Unified CVP stores the last-entered invalid digit in the `user.microapp.caller_input` variable, exits the node through the X (failure) branch, sets the `user.microapp.error_code` ECC variable to **16** (Reached Maximum Invalid Tries), and returns control to Unified ICME.



Note See [Unified CVP Script Error Checking](#).

No Entry Timeout Occurs

When the caller does not enter input within the No Entry Timeout period and:

- The Number of No Entry Tries value has not been reached, Unified CVP plays the "no entry" error message and re-plays the prompt.

- The Number of No Entry Tries value *has* been reached, Unified CVP exits the node through the X (failure) branch, sets the **user.microapp.caller_input** ECC variable to NULL, the **user.microapp.error_code** ECC variable to **17** (Reached Maximum No Entry Tries), and returns control to Unified ICME.

Get Speech and External VoiceXML

You can use the Get Speech micro-application to pass information to and from an external VoiceXML file. The following table describes how to set the Get Speech script to use external VoiceXML.

To set up the Get Speech micro-application to use external VoiceXML, set the Media Library Type to "V". The IVR Service creates VoiceXML that calls the external VoiceXML that is specified in the external VoiceXML file name. The URL to the external VoiceXML is formed from a combination of the media_server, locale, App_Media_Lib and external VoiceXML file name. If the VoiceXML file name does not contain a file extension, the default "*.VoiceXML" is used.

If the external VoiceXML is used, the only GetSpeech VRU Script parameters that are used are:

- "Number of Invalid Entry" errors, and
- "Number of No Entry" errors.

The IVR Service "NoEntry" and "InvalidEntry" retry logic are used if the external VoiceXML returns a <noinput> or <nomatch> event.

Error Handling

Error handling

The error handling for an external VoiceXML called from the Get Speech micro-application includes the following:

- If you set the "Media Library Type" to "V" and you do not set an "External VoiceXML Name" parameter, an "Invalid VRU Script Name" error is returned to Unified ICME.

Passing Information to the External VoiceXML

There are two methods of passing information to the external VoiceXML, either by <param> elements or URL elements. You can pass up to 1050 characters to the external VoiceXML by using an ECC Variable array.

Table 11: To External VoiceXML ECC Variable Array

ECC Variable Name	Type	Max. Number of Elements	Max. Size of Each Element
user.microapp.ToExtVXML	Array	5	210

This variable array contains a list of semicolon delimited name/value pairs. The following is an example of the syntax:

Table 12: Sample Array Definition

Variable Name	Values
user.microapp.ToExtVXML[0]	"Company=Cisco;Job=technical writer"

Variable Name	Values
user.microapp.ToExtVXML[1]	"Location=Boxborough;Street=Main"
user.microapp.ToExtVXML[2]	"FirstName=Gerrard;LastName=Thock"
user.microapp.ToExtVXML[3]	"Commute=1 hour;Car=Isuzu"
user.microapp.ToExtVXML[4]	"BadgeID=2121212"

Unified CVP links all five elements of the "ToExtVXML" array, parses the contents, and then puts each of the name/value pairs in the VoiceXML that it creates.

You define an ECC variable to determine which method to use when passing information to the external VoiceXML.

Table 13: Use External VoiceXML ECC Variable

ECC Variable Name	Type	Max. # of Elements	Possible Values
user.microapp.UseVXMLParams	Scalar	1	<ul style="list-style-type: none"> • Y - (Yes) Use the values in the user.microapp.ToExtVXML variable array elements. • N - (No) Append the name/value pairs in user.microapp.ToExtVXML to the URL of the external VXML. <p>Default: "N"</p>

- If **user.microapp.UseVXMLParams** is set to a value other than "Y" or "N," the IVR Service sends a Misconfigured ECC Variable error message to Unified ICME.
- If the **user.microapp.UseVXMLParams** variable is not set, the default method is "VXML Parameters."
- If the **user.microapp.UseVXMLParams** is set to its default value "N," the .vxml file extension is not used.

Using <param> Elements

All name/value pairs declared in the **user.microapp.ToExtVXML** variable array are added to the VoiceXML file that the IVR Service creates in <param> elements. These <param> elements pass information to the external VoiceXML. Unlike the URL parameters, VoiceXML parameters are a complete VoiceXML solution that does not require media server side scripting.

You must declare names specified in the name/value pairs as variables in the external VoiceXML. Otherwise the VoiceXML Interpreter produces a error.semantic error that is returned to Unified ICME as error code "10." Using the example above in the table [Table 12: Sample Array Definition, on page 59](#), you must define the following form level declarations in the external VoiceXML.

```
<var name="Company"/>
<var name="Job"/>
<var name="Location"/>
<var name="Street"/>
```

```
<var name="FirstName"/>
<var name="LastName"/>
<var name="Commute"/>
<var name="Car"/>
<var name="BadgeID"/>
```

This is why that Unified CVP does not allow the Script Writer to specify a name without a value. Specifying only a name in a ToExtVXML parameter is useless because the external VoiceXML already has the variable defined.

URL Parameter Element

When you use the URL parameter element option, the name/value pairs that you set in the **user.microapp.ToExtVXML** parameter are appended to the URL to the external VoiceXML. The media server side scripting logic parses the URL and passes the parameters to the external VoiceXML document. Unlike the “VXML Parameters,” this is not a VoiceXML solution and requires media-server side scripting.

Using the examples in [Table 12: Sample Array Definition, on page 59](#), the URL to the external VoiceXML is in the following form:

```
http://server/en-us/app/MyVXML?Company=Cisco&Job=technical+writer&
Location=Boxborough&Street=Main&FirstName=Gerrard&LastName=Thock&
Commute=1+hour&Car=Isuzu&BadgeID=2121212
```

ECC Variable Array Formula

Use the following equation to determine how many bytes you are sending to and from the external VoiceXML. This calculation ensures that you do not overload the ECC Variable with too many bytes.

$$5 + (1 + \text{Maximum_Length}) * (\text{Maximum_Array_Size})$$

For example, if you are sending three array elements to the external VoiceXML of the maximum 210 bytes, the equation looks like this:

$$\begin{aligned} 5 + (1 + 210) * 3 \\ 5 + (211 * 3) \\ 5 + 633 \\ 638 \end{aligned}$$

Although the maximum number of bytes you can set each variable to is 210, maxing out each variable goes over the 1050 character limit. Make sure you use this formula to keep your character limit under the 1050 maximum.

Use a single 210-byte array element in each direction.



Note When you are returning a call from the external VoiceXML, the `user.microapp.caller_input` variable is automatically returned.

Notes

- If the **user.microapp.ToExtVXML** array is either not defined on Unified ICME or empty, the IVR Service does not pass any parameters to the external VoiceXML.

- You do not need to define the **user.microapp.ToExtVXML** array to contain five elements. However, it must be defined as an ARRAY variable, not a SCALAR variable, even if you are using only one element. The IVR Service can handle five array elements.
- If the **user.microapp.ToExtVXML** is either undefined or partially defined, a warning message appears on the VRU PIM console window.
- Although the array elements are linked together, you can't span a name/value pair to multiple array elements. This is because before you parse for the name/value pairs, the IVR Service inserts a semicolon between two array elements if there is not one.
- The IVR Service produces a "Misconfigured ECC Variable" error if there is not a "=" symbol between two semicolons.
- The IVR Service produces a "Misconfigured ECC Variable" error if the "=" symbol is the first or last character between two semicolons (for example, if there is not a name or a value).
- The IVR Service produces a "Misconfigured ECC Variable" error if the "name" part of the name/value pair contains a space.
- The IVR Service treats each of the name/value parameters as strings. The IVR Service does not check to see if the value parameter is an integer.

Passing Data Back to Unified ICME with External Voice XML

Unified CVP can return 1050 characters for external VoiceXML.



Note All other Get Speech nodes are limited to the 210 characters returned in **user.microapp.caller_input**.

The following ECC Variable array is added:

Table 14: From External VoiceXML ECC Variable Array

ECC Variable Name	Type	Max. Number of Elements	Max. Size of Each Element
user.microapp.FromExtVXML	Array	4	210

The Get Speech micro-app returns up to 1050 characters by populating the **user.microapp.caller_input** variable and each element of the **user.microapp.FromExtVXML** array.



Note Use the ECC Variable array formula when defining the FromExtVXML variable. Do not define the user.microapp.FromExtVXML array to contain 5 elements. However, it must be defined as an ARRAY variable, not a SCALAR variable, even if you are using only one element.

Voice XML Requirements

External VoiceXML is called via the <subdialog> VoiceXML element. Because of the design of VoiceXML, the variable names in the customer defined VoiceXML must be coordinated with Unified CVP. Otherwise, it won't be possible to pass external VoiceXML data back to Unified CVP. The following table lists the VoiceXML variables and the ECC variables they correspond to.

Table 15: From External VoiceXML ECC Variable Definition

External VoiceXML Variable Name	Unified ICME ECC Variable	Max. Variable Size
caller_input	user.microapp.caller_input	210
FromExtVXML0	user.microapp.FromExtVXML[0]	210
FromExtVXML1	user.microapp.FromExtVXML[1]	210
FromExtVXML2	user.microapp.FromExtVXML[2]	210
FromExtVXML3	user.microapp.FromExtVXML[3]	210

Define the following form level declarations in the external VoiceXML.

```
<var name="caller_input"/>
```

When passing the information back to Unified ICME, use the following syntax for caller input:

```
<assign name="caller_input" expr="input$.utterance"/>
```

If the external VoiceXML sets “input” to “sales” and “FromExtVXML2” to “stocks”, the user.microapp.caller_input is set to “sales” and user.microapp.FromExtVXML[2] is set to “stocks”.

Notes

- The variables declared in the customer VoiceXML must be named exactly as specified. VoiceXML is case-sensitive.
- The “caller-input” variable must be declared and used in the VoiceXML. The only way that it is acceptable to not populate this variable is if the external VoiceXML is returning an error event. Examples of error events include <badfetch>, <noinput>, and <nomatch>. If the “caller_input” variable is not set to a value and an error event is not generated, the Unified CVP assumes that a “No Entry” error occurred.
- The FromExtVXML variables are optional. You can use these variables if the “caller_input” variable is not sufficient. If you are not using this additional data then the ECC Variable array does not need to be defined.
- If the user.microapp.FromExtVXML ECC Variable array is undefined or partially defined, a warning message appears when the IVR Service starts up on the VRU PIM console window.
- If you set a “FromExtVXML” variable in the external VoiceXML, the user.microapp.FromExtVXML must be defined on Unified ICME. If it is not, the Unified CVP does not attempt to set the value and an error appears on the VRU PIM console window. However, no error appears in the Unified CVP log files.
- If you need more than 210 characters, but less than the full 1050 characters, you can declare the user.microapp.FromExtVXML array to be less than four elements long. If you do this, you only use the corresponding number of External VoiceXML Variables. For example, if you configure a two-element “FromExtVXML” ECC Variable array, you can utilize the “caller_input”, “FromExtVXML0” and “FromExtVXML1” VoiceXML variables.
- If the external VoiceXML sets a FromExtVXML variable to a value that is longer than the maximum ECC Variable length, an error message appears in the VRU PIM console window and the value is not set. No error appears in the Unified CVP log files.

Sample External VoiceXML Code

This section provides sample external VoiceXML code.

```
<?xml version="1.0">
<vxml version="2.0">
<var name="caller_input"/>
  <form id="getcredit">
<field name="input">
  <prompt>
    What is your credit card type?
  </prompt>
  <help>
    I am trying to collect your credit card type.
  </help>
  <reprompt/>
</field>
<field name="FromExtVXML0">
  <prompt>
    What is your credit card number?
  </prompt>
  <help>
    I am trying to collect your credit card information.
  </help>
  <reprompt/>
</field>
<field name="FromExtVXML1">
  <prompt>
    I am trying to collect the expiration date of the credit card number you provided.
  </prompt>
  <help>
    I am trying to collect the expiration date of the credit card number you provided.
  </help>
  <reprompt/>
</field>
</form>
<assign name="caller_input" expr="input$.utterance"/>
<return namelist="caller_input FromExtVXML0 FromExtVXML1"/>
</block>
<catch event="telephone.disconnect.hangup">
  <return event="telephone.disconnect.hangup"/>
</catch>
<catch event="error.badfetch">
  <return event="error.badfetch"/>
</catch>
<catch event="error.semantic">
  <return event="error.semantic"/>
</catch>
<catch event="error.unsupported.format">
  <return event="error.unsupported.format"/>
</catch>
<catch event="error.unsupported.element">
  <return event="error.unsupported.element"/>
</catch>
<catch event="error.unsupported.language">
```

```

    <return event="error.unsupported.language"/>
  </catch>
  <catch event = "error.com.cisco.media.resource.unavailable.asr">
    <return event=" error.com.cisco.media.resource.unavailable.asr"/>
  </catch>
  <catch event = "error.com.cisco.media.resource.unavailable.tts">
    <return event=" error.com.cisco.media.resource.unavailable.tts"/>
  </catch>
  <catch event = "error.com.cisco.media.resource.failure.asr">
    <return event=" error.com.cisco.media.resource.failure.asr"/>
  </catch>
  <catch event = "error.com.cisco.media.resource.failure.tts">
    <return event=" error.com.cisco.media.resource.failure.tts"/>
  </catch>
  <catch event = "error.com.cisco.media.resource">
    <return event=" error.com.cisco.media.resource"/>
  </catch>
  <catch event = "error">
    <return event="error"/>
  </catch>
  <form>
</vxml>

```

External VoiceXML File Contents

An external VoiceXML file must adhere to the following rules:

- It must not use the <transfer>, <exit> or <disconnect> elements. However, it can use the <Goto> and <submit> elements.
- It must have <return> elements at all exit points in the document.
- It must check for all error events and the “telephone.disconnect.hangup” event. Each event handler must have a <return> element that includes the “event” attribute.
- It must contain <catch> event handlers for all events thrown by the Gateway. These catch handlers can have their own customer-defined logic, but they *must* include the statements that are listed in the sample VoiceXML provided in the previous section.

The following External VoiceXML document example illustrates the contents of a VoiceXML document that follows these rules.



Note This example assumes that the VRU Script Name value is **PM,CustomerVXML,V**.

External VoiceXML Document Example

```

<?xml version="1.0"?>
<vxml version="2.0">
  <form id="CustomerVXML" scope="dialog">
    <catch event="error.com.cisco.callhandoff.failure">
      <return/>
    </catch>

    <object name="dummyobj" classid="builtin://com.cisco.callhandoff">
      <param name="return" expr="true" valuetype="data"/>
      <param name="app-uri" expr="'builtin://dummyobj'" valuetype="data"/>
      <prompt bargein="true">

```

```

    <audio src="http://192.168.1.20:80/en-us/app/Hello_World.wav" />
  </prompt>
  <filled>
    <return/>
  </filled>
</object>

<catch event="error.badfetch">
  <return event="error.badfetch"/>
</catch>
<catch event="error.semantic">
  <return event="error.semantic"/>
</catch>
<catch event = "error.unsupported.format">
  <return event="error.unsupported.format"/>
</catch>
<catch event = "error.unsupported.element">
  <return event="error.unsupported.element"/>
</catch>
<catch event="telephone.disconnect.hangup">
  <return event="telephone.disconnect.hangup"/>
</catch>
<catch event="error">
  <return event="error"/>
</catch>
<block>
  <return/>
</block>
</form>
</vxml>

```



Note For a complete explanation of VoiceXML file grammar format, refer to <http://www.w3.org/TR/speech-grammar/>. Also, consult the ASR Server user documentation for a list of supported grammar elements.

The example that follows illustrates another external grammar file that prompts callers for the state that they live in.

External Grammar file

```

<?xml version = 1.0?>
<grammar version= 1.0 root= action xml:lang= en-us >
<rule id= action scope= public >
<one-of>
<item> California </item>
<item> Arizona </item>
<item> Connecticut </item>
</one-of>
</rule>
</grammar>

```

After a caller responds with the state the caller lives in, the ASR Engine determines if the caller said **California**, **Arizona**, or **Connecticut**. If the caller said the name of one of these states, the text listed in the **<item>** element is passed to the IVR Service and, **ICM** software. If a caller responds with a name not included in this list, an invalid entry error is returned to the IVR Service.

Type-Ahead Support for ASR

Type-ahead support for ASR is only supported for DTMF when the Gateway is the client and input type is set to **D**.

Scripting for Unified CVP with Call Studio

You can use Call Studio to build sophisticated IVR applications which can then be loaded onto a VXML Server machine for execution.

To invoke a VXML Server application, create a Unified ICME routing script that

- Includes a user.microapp.ToExtVXML[0] ECC variable instructing the VoiceXML Gateway to interact with the VXML Server directly to execute the application
- Instructs the application to pass back results to Unified ICME

This section describes

- Call Studio and how to use it to pass data to Unified ICME
- How to integrate Call Studio scripts with Unified ICME scripts
- How to deploy Call Studio Scripts in Unified CVP

About Call Studio

Call Studio is an Eclipse-based service creation environment whose output is an intermediary file which describes the application flow.

Among its many features, the Call Studio scripting environment

- Has a drag-and-drop interface with a palette of IVR functions
- Can perform database queries
- Can be extended with Java code written to perform any task a Java application can perform

The following figure shows a Call Studio application that can be used with the Unified CVP Standalone with ICM Lookup call flow model.

The screenshot displays the Cisco Unified Call Studio Builder interface. The main workspace shows a call flow diagram starting with 'Start Of Call', followed by 'Subdialog Start_01', 'Audio_01', and 'ReqICMLabel_01'. From 'ReqICMLabel_01', the flow branches into 'Audio_03' (done) and 'Audio_02' (error), both leading to 'Subdialog Return_01'. The 'Element Configuration' pane on the right shows the configuration for the 'ReqICMLabel' element, including a 'Timeout' of 3000 and a list of variables: 'callvar1', 'callvar2', 'callvar3' (value: contentCV3), 'callvar4', 'callvar5', 'callvar6', 'callvar7', 'callvar8', 'callvar9', 'callvar10', 'FromExtVXML0', 'FromExtVXML1', 'FromExtVXML2', 'FromExtVXML3', and 'caller_input'.

The 'Prompt Manager' pane at the bottom shows a table of audio items:

Page	Element Group	Element Name	AG Name	AG Language	AG Count	Audio Item Name
page 1		Audio_01	Initial	Default	1	audio item 1
page 1		Audio_02	Initial	Default	1	audio item 1
page 1		Audio_03	Initial	Default	1	audio item 1

Call Studio ReqICMLabel Element to Pass Data

The ReqICMLabel element allows a Call Studio script to pass caller input, Call Peripheral variables, and Expanded Call Context (ECC) variables to a Unified ICME script. The ReqICMLabel must be inserted into a Call Studio script as a decision element. In Call Studio, the returned Unified ICME label result can be used by other elements in the same application, such as the Transfer or Audio element. The Transfer element sends instructions to the IOS Voice Browser to transfer the caller to the desired location.

After the ReqICMLabel exits its path, you can retrieve the values set by the Unified ICME script by selecting the Element Data tab for the ReqICMLabel element. The element data value is `{Data.Element.ReqICMLabelElement.result}`. ReqICMLabelElement is the name of the ReqICMLabel element in the Call Studio script. The default name for this element is ReqICMLabel_<n>. For example, if you changed ReqICMLabel to GetICMLabel, the value returned from Unified ICME is `{Data.Element.GetICMLabel.result}`, where *result* is the variable of the ReqICMLabel element that contains the Unified ICME label.

Table 16: Settings

Name (Label)	Type	Required	Single Setting Value	Substitution Allowed	Default	Notes
Call Peripheral Variables 1 - 10 (callvar1 - callvar10)	String	No	Yes	Yes		Call Peripheral variables passed by the Call Studio script to the Unified ICME Server. This setting can be a maximum of 40 characters. The Unified ICME Server returns a name-value pair for up to 10 Call Peripheral Variables in a result. Any value that is placed in callvar<n> from a Call Studio script is returned unchanged, if the Unified ICME Script does not change it.
Call Peripheral Variables Return 1 - 10 (callvarReturn1 - callvarReturn10)	String	No	Yes	Yes		Call Peripheral variables created upon the return of the Unified ICME Label request, regardless of whether or not these variables are filled by the Unified ICME Script. You need two sets of these variables to keep reporting the To ICM Call Peripheral Variables separate from what is returned from Unified ICME.
FromExtVXML0 - 3 (External VXML 0 - External VXML 3)	String Array	No	Yes	Yes		Expanded Call Context (ECC) variables passed by the Call Studio script to the Unified ICME Server. Each variable is a string of name-value pairs, separated by semicolons, for up to four external VoiceXML variables. This setting can be a maximum of 210 characters.
ToExtVXML0 - 4 (External VXML 0 - External VXML 4)	String Array	No	Yes	Yes		Expanded Call Context (ECC) variables received from the Unified ICME script. The Unified ICME Server returns a string of name-value pairs, separated by semicolons, for up to five external VoiceXML variables.

Name (Label)	Type	Required	Single Setting Value	Substitution Allowed	Default	Notes
Timeout	Integer	Yes	Yes	Yes	3000 (ms)	The number of milliseconds that the transfer request waits for a response from the Unified ICME Server before timing out. Note This value is increased or decreased by increments of 500 ms.
caller_input (Caller Input)	String	No	Yes	Yes		This setting can be a maximum of 210 characters. The caller_input is only passed to Unified ICME from Call Studio.

Table 17: Element Data

Name	Type	Notes
result	String	Unified ICME Label returned from a Unified ICME server. You can use this result as input to other Call Studio elements, such as Transfer or Audio. The element data value is {Data.Element.ReqICMLLabelElement.result}.
callvar<n	String	Call Peripheral variables that the Call Studio scripts passes to the Unified ICME Server. Valid Call Peripheral Variables are callvar1 - callvar10.
callvarReturn<n	String	Call Peripheral variables that the Unified ICME script returns to the VXML Server. Valid Call Peripheral Variables are callvarReturn1 - callvarReturn10. For example, if a Unified ICME script contains Call Peripheral variable 3 with the string value “CompanyName=Cisco Systems, Inc”, you can access the value of CompanyName that is returned by the Unified ICME script by using Data.Element.ReqICMLLabelElement.callvarReturn3 The returned value is “Cisco Systems, Inc.”

Table 18: Session Data

Name	Type	Notes
name	String	<p>Value for a name-value pair contained in a ToExtVXML variable returned in the Unified ICME label. You must know which name-value pairs are set in the Unified ICME script to retrieve the correct value from the Call Studio script.</p> <p>For example, if a Unified ICME script contains a user.microapp.ToExtVXML0 variable with the string value "CustomerName=Mantle", specify Data.Session.CustomerName. If the same Unified ICME script contains a user.microapp.ToExtVXML0 variable with the string value "BusinessType=Manufacturing", you can access the customer business type returned by the Unified ICME script by using Data.Session.BusinessType.</p>

Table 19: Exit States

Name	Notes
done	The element execution is complete and the value is successfully retrieved.
error	The element failed to retrieve the value.

Studio Element Folder is "Cisco."

Integrate Call Studio Scripts with Unified ICME Scripts - Traditional Method

This section describes how to integrate the VXML Server into the Unified CVP solution in the traditional way. This process involves

- Creating a Unified ICME script with ECC variables configured for VXML Server
- Creating a VRU Script to run in the Unified ICME script



Note There are two ways of integrating VXML Server into the Unified CVP solution. The first method is the same as in previous releases where ECC variables are used for specification of the parameters needed for the integration. This method is named as the 'traditional' method in the description below. Unified CVP continues to support the first method. The second method is described in the [Integrate Call Studio Scripts with Unified ICME Scripts - Simplified Method](#).

Procedure

Step 1 Specify the URL and port number of the VXML Server that you want to reach, for example:

```
http://10.78.26.28:7000/CVP/Server?application=HelloWorld
```

In the example above, 10.78.26.28 is the URL and 7000 port number; the values are delimited by a colon (:).

Note 7000 is the default port number for a VXML Server.

Step 2 In the Unified ICME script, first set the media_server ECC variable to:

http://10.78.26.28:7000/CVP

Step 3 Set the app_media_lib ECC Variable to "..", (literally two periods in quotes).

Step 4 Set the user.microapp.ToExtVXML[0] ECC variable to application=HelloWorld.

Note This example indicates that the VXML Server will execute the "HelloWorld" application. To execute a different application, change the value of user.microapp.ToExtVXML[0].

Step 5 Set the UseVXMLParams ECC Variable to "N."

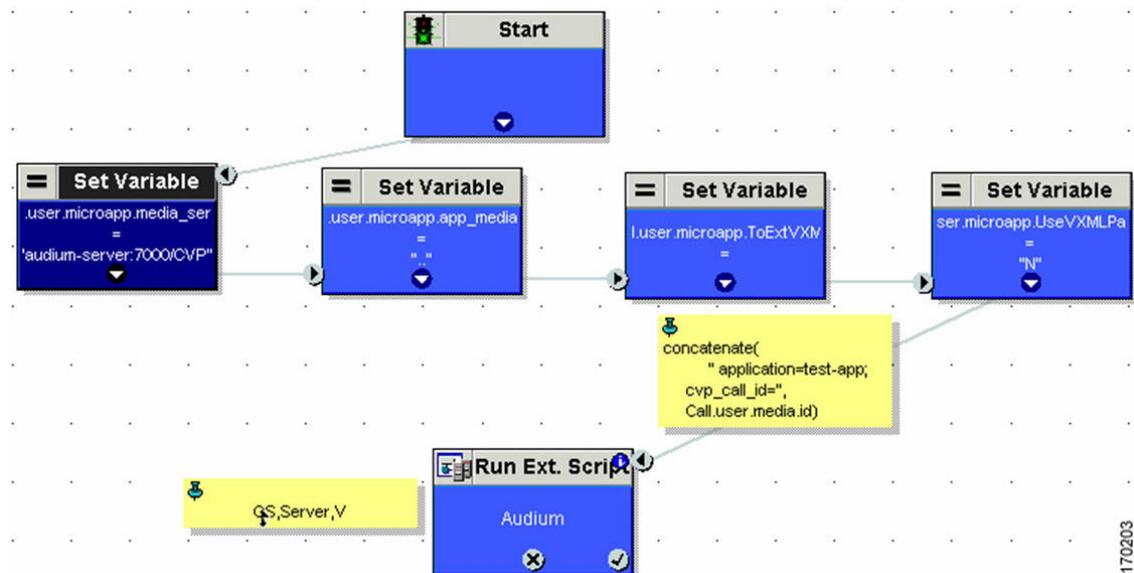
Step 6 Set the concatenate element by following the instructions in *Correlating Unified CVP/Unified ICME Logs with VXML Server Logs.*

Step 7 Create a Run External Script node within the Unified ICME script with a VRU Script Name value of GS,Server,V.

Note Link this node to the nodes configured in previous steps.

- Configure the timeout setting in the Network VRU Script to a value substantially greater than the length of the timeout in the VXML Server application. (This timeout is only be used for recovery from a failed VXML Server.)
- Always leave the **Interruptible** check box in the Network VRU Script Attributes checked. Otherwise, calls queued to a VXML Server application may stay in the queue when an agent becomes available.

Figure 1: Example Script Configuration



After you configure the Unified ICME script, configure a corresponding VXML Server script with Call Studio. The VXML Server script must

- Begin with a Unified CVP Subdialog_Start element (immediately after the Call Start element)

- Contain a Unified CVP Subdialog_Return element on all return points (script must end with a Subdialog_Return element)
- Must include a value for the call input for the Unified CVP Subdialog_Return element
- Must add Data Feed/SNMP loggers to enable reporting

Integrate Call Studio Scripts with Unified ICME Scripts - Simplified Method

It is applicable only if the CVP Call Server and the VXML Server are co-located. This method simplifies the Unified ICME script configuration and reduces the script nodes that need to be configured. The GS micro-application assumes that it invokes the VXML Server that is co-located with the Call Server if the following conditions are met:

- The GetSpeech (GS) micro-application invokes a VXML Server application as follows:
 - The "Media File Name" of the GS is set to "Server" e.g., "GS,Server,V". ("Server" is case-sensitive).
 - - The "Media Library Type" of the GS is set to "V" (External VoiceXML).
- The ECC variable user.microapp.media_server has not been encountered in the Unified CCE script when the GS RunScript node is run.

With this method, the following ECC variables are not needed:

- user.microapp.media_server
- user.microapp.UseVXMLParams
- user.microapp.app_media_lib

Procedure

-
- Step 1** Set the user.microapp.ToExtVXML[0] ECC variable to application=HelloWorld.
- Note** This example indicates that the VXML Server executes the "HelloWorld" application. To execute a different application, change the value of user.microapp.ToExtVXML[0] accordingly.
- Step 2** Create a Run External Script node within the Unified ICME script with a VRU Script Name value of GS,Server,V.
- Configure the timeout setting in the Network VRU Script to a value greater than the timeout value in the VXML Server application. (This timeout is only used for recovery from a failed VXML Server.)
 - Always leave the **Interruptible** checkbox in the Network VRU Script Attributes checked. Otherwise, calls queued to a VXML Server application may stay in the queue when an agent becomes available.
- Step 3** After you configure the Unified ICME script, configure a corresponding VXML Server script with Call Studio. The VXML Server script must
- Begin with a Unified CVP Subdialog_Start element (immediately after the Call Start element)
 - Contain a Unified CVP Subdialog_Return element on all return points (script must end with a Subdialog_Return element)

- Must include a value for the call input for the Unified CVP Subdialog_Return element
- Must add Data Feed/SNMP loggers to enable logging, as shown:



Call Studio Scripts in Unified CVP

Call Studio scripts can be deployed in one of the following ways:

- In Call Studio, create and deploy the Call Studio scripts to the local machine using the **Archive** option.
- In the Operations Console, upload the archived Call Studio script file from the local machine to the Operations Server and deploy it to other VXML Server machines.

Deploy Call Studio Scripts Using Call Studio

Procedure

- Step 1** Create or modify one or more VoiceXML application scripts.
- Step 2** Deploy one or more VoiceXML application scripts to the local machine using the archive option. The archived scripts are saved as a zipped file under a user-specified directory, for example:
- C:\Program Files\Cisco\CallStudio
- Note** The sample folder is C:\Cisco\CallStudio, which is also the default folder.
- Step 3** Use Call Studio to set up the loggers using the ActivityLogger, ErrorLogger, and Admin Logger tools. Set up the Unified CVP Datafeed logger for each application.
- Note** Call Studio also includes CVPDatafeedLogger and CVPSNMPLLogger. Call Studio lets you change other parameters for these loggers, such as log file size, log lever, et cetera.
- See the Call Studio documentation for more information.

Deploy Call Studio Scripts Using the Operations Console

Procedure

- Step 1** From the web browser, enter the following URL:

https://ServerIP:9443/oamp or http://ServerIP:9000/oamp

Step 2 Enter your user ID in the User Name field.

Note The first time you log in after installing Unified CVP, enter Administrator, the default user account.

Step 3 In the Password field, enter your password, as follows:

- If you are logging in to the default Administrator account, enter the password that was set for this account during installation.
- If the user ID or password is invalid, the Operations server displays the message, "Invalid Username or password." Click the link, enter your user ID and password again, and click **OK**.

The Operations Console Welcome window appears.

Step 4 Select **Bulk Administration > File Transfer > Scripts and Media**.

Step 5 From the Device Association drop-down menu, select **Gateway**.

Step 6 In the Available pane, select one or more archived script files to deploy.

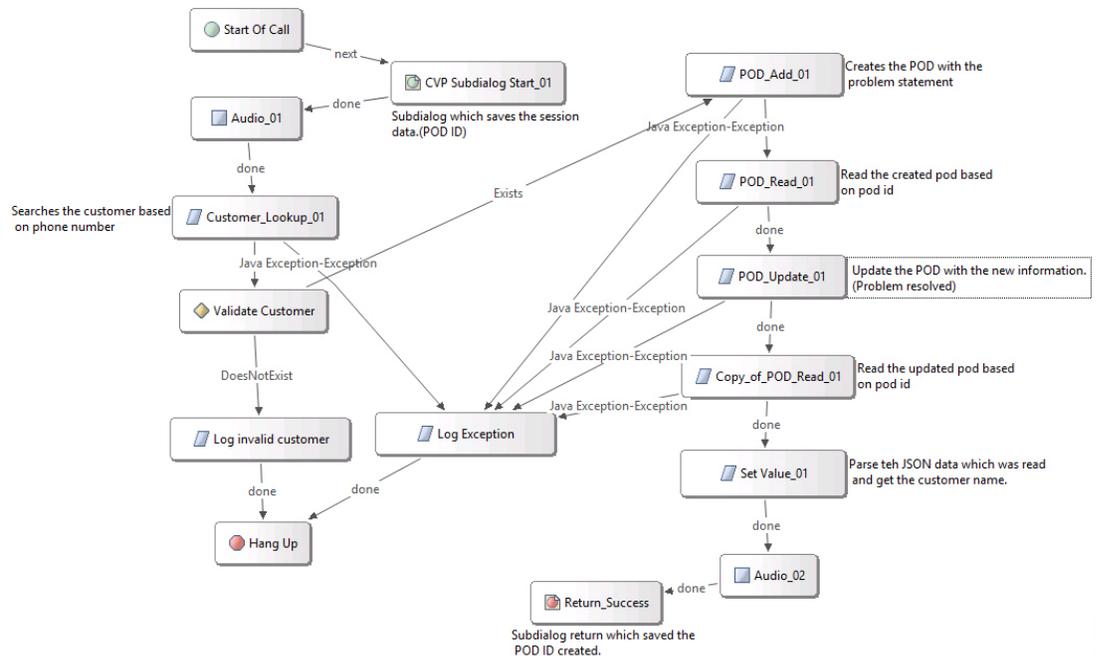
Step 7 Click the **arrow icon** to move the file from *Available* to *Selected*.

Step 8 Click **Transfer** to transfer the selected archived scripts file(s) to the selected device.

Scripting for Unified CVP with Context Service

You can create a new Call Studio application by selecting **File > New > Call Studio Project** from the Call Studio and adding different elements, based on your requirements. This illustration shows a sample script:

Figure 2: Sample Script for Unified CVP with Context Service



510035

The following key explains the different context service elements:

- **Start Of Call Element:** Created, by default, with an empty project.
- **CVP Subdialog Start_01 Element:** This element defines the start of the subflow.
- **Customer_Lookup_01 Element:** This element searches for existing customer, based on phone number.
- **Validate Customer Element:** This is a decision element, which confirms customer is valid.
- **Log invalid Element:** This element returns error if customer not found.
- **Log Exception Element:** This element returns exception if any.
- **POD_Add_01 Element:** This element creates a new POD for the call using the customer id and other parameters.
- **POD_Read_01 Element:** This element reads the created POD based on POD ID.
- **POD_Update_01 Element:** This element updates the created POD with additional information .
- **Copy_of_POD_Read_01 Element:** This element reads the updated POD based on POD ID.
- **Set Value_01 Element:** This element parses the JSON data which was read and get the customer name.
- **Return_Success Element:** This subdialog return element holds the POD ID created.

For more details on these elements, see the *Element Specifications for Cisco Unified CVP VXML Server and Cisco Unified Call Studio* at <http://www.cisco.com/c/en/us/support/customer-collaboration/unified-customer-voice-portal/products-programming-reference-guides-list.html>.

System Media Files

The following tables describe the English System Media Files installed by Unified CVP. These system media files are intended as samples only. It is the Customer/Media Administrator's responsibility to record all the system prompts for all the locales.

The table that follows lists the System Media File information for cardinal numbers.

Table 20: System Media Files, Cardinal Numbers

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types /When Media File Is Used
		point	point	Number
		minus	minus	Number
0	48	0	zero	All except DOW
1	49	1	one (masculine version), uno (es-mx and es-es)	All except DOW
2	50	2	two	All except DOW

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types /When Media File Is Used
3	51	3	three	All except DOW
4	52	4	four	All except DOW
5	53	5	five	All except DOW
6	54	6	six	All except DOW
7	55	7	seven	All except DOW
8	56	8	eight	All except DOW
9	57	9	nine	All except DOW
		10	ten	Same for the rest of all the numbers
		11	eleven	
		12	twelve	
		13	thirteen	
		14	fourteen	
		15	fifteen	
		16	sixteen	
		17	seventeen	
		18	eighteen	
		19	nineteen	
		20	twenty	
		21	twenty-one	
		22	twenty-two	
		23	twenty-three	
		24	twenty-four	
		25	twenty-five	
		26	twenty-six	
		27	twenty-seven	
		28	twenty-eight	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		29	twenty-nine	
		30	thirty	
		31	thirty-one	
		32	thirty-two	
		33	thirty-three	
		34	thirty-four	
		35	thirty-five	
		36	thirty-six	
		37	thirty-seven	
		38	thirty-eight	
		39	thirty-nine	
		40	forty	
		41	forty-one	
		42	forty-two	
		43	forty-three	
		44	forty-four	
		45	forty-five	
		46	forty-six	
		47	forty-seven	
		48	forty-eight	
		49	forty-nine	
		50	fifty	
		51	fifty-one	
		52	fifty-two	
		53	fifty-three	
		54	fifty-four	
		55	fifty-five	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types /When Media File Is Used
		56	fifty-six	
		57	fifty-seven	
		58	fifty-eight	
		59	fifty-nine	
		60	sixty	
		61	sixty-one	
		62	sixty-two	
		63	sixty-three	
		64	sixty-four	
		65	sixty-five	
		66	sixty-six	
		67	sixty-seven	
		68	sixty-eight	
		69	sixty-nine	
		70	seventy	
		71	seventy-one	
		72	seventy-two	
		73	seventy-three	
		74	seventy-four	
		75	seventy-five	
		76	seventy-six	
		77	seventy-seven	
		78	seventy-eight	
		79	seventy-nine	
		80	eighty	
		81	eighty-one	
		82	eighty-two	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		83	eighty-three	
		84	eighty-four	
		85	eighty-five	
		86	eighty-six	
		87	eighty-seven	
		88	eighty-eight	
		89	eighty-nine	
		90	ninety	
		91	ninety-one	
		92	ninety-two	
		93	ninety-three	
		94	ninety-four	
		95	ninety-five	
		96	ninety-six	
		97	ninety-seven	
		98	ninety-eight	
		99	ninety-nine	
		oh	oh	24TOD, Date
		hundred	hundred	Number, 24TOD, Date, Currency
		thousand	thousand	Number, Date, Currency
		million	million	Number, Currency
		billion	billion	Number, Date, Currency
		trillion	trillion	Number, Currency

The table that follows lists the System Media File information for ordinal numbers.



Note If ordinal system prompts are to be used in a script for a purpose other than dates, they should be recorded as application prompts with the true ordinal values.

Table 21: System Media Files, Ordinal Numbers

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		1ord	first	Date
		2ord	second	Date for all ordinal numbers
		3ord	third	
		4ord	fourth	
		5ord	fifth	
		6ord	sixth	
		7ord	seventh	
		8ord	eighth	
		9ord	ninth	
		10ord	tenth	
		11ord	eleventh	
		12ord	twelveth	
		13ord	thirteenth	
		14ord	fourteenth	
		15ord	fifteenth	
		16ord	sixteenth	
		17ord	seventeenth	
		18ord	eighteenth	
		19ord	nineteenth	
		20ord	twentieth	
		21ord	twenty-first	
		22ord	twenty-second	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		23ord	twenty-third	
		24ord	twenty-fourth	
		25ord	twenty-fifth	
		26ord	twenty-sixth	
		27ord	twenty-seventh	
		28ord	twenty-eight	
		29ord	twenty-nineth	
		30ord	thirtieth	
		31ord	thirty-first	

The table that follows lists the System Media File information for measurements.

Table 22: System Media Files, Measurements

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
½	189	one_half	one half	Char
¼	188	one_quarter	one quarter	Char
¾	190	three_quarters	three quarters	Char
A, a	65,97	a	A	Char
B, b	66,98	b	B	Char
C, c	67,99	c	C	Char
D, d	68,100	d	D	Char
E, e	69,101	e	E	Char
F, f	70,102	f	F	Char
G, g	71,103	g	G	Char
H, h	72,104	h	H	Char
I, I	73,105	I	I	Char
J, j	74,106	j	J	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
K, k	75,107	k	K	Char
L, l	76,108	l	L	Char
M, m	77,109	m	M	Char
N, n	78,110	n	N	Char
O, o	79,111	o	O	Char
P, p	80,112	p	P	Char
Q, q	81,113	q	Q	Char
R, r	82,114	r	R	Char
S, s	83,115	s	S	Char
T, t	84,116	t	T	Char
U, u	85,117	u	U	Char
V, v	86,118	v	V	Char
W, w	87,119	w	W	Char
X, x	88,120	x	X	Char
Y, y	89,121	y	Y	Char
Z, z	90,122	z	Z	Char
Œ, œ	140,156	oe_140_156	Ligature OE	Char
À,à	192,224	a_192_224	A grave	Char
Á,á	193,225	a_193_225	A acute	Char
Â,â	194,226	a_194_226	A circumflex	Char
Ã,ã	195,227	a_195_227	A tilde	Char
Ä,ä	196,228	a_196_228	A umlaut	Char
Å,å	197,229	a_197_229	A with ring above	Char
Æ,æ	198,230	ae_198_230	Ligature AE	Char
È,è	200,232	e_200_232	E grave	Char
É,é	201,233	e_201_233	E acute	Char
Ê,ê	202,234	e_202_234	E circumflex	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
Ë,ë	203,235	e_203_235	E umlaut	
Ì,ì	204,236	i_204_236	I grave	Char
Í,í	205,237	i_205	I acute	Char
Î,î	206,238	i_206	I circumflex	Char
Ï,ï	207,239	i_207	I umlaut	Char
Ð	208	char_208	character 208	Char
ð	240	char_240	character 240	
Ò,ò	210,242	o_210_242	O grave	Char
Ó,ó	211,243	o_211_243	O acute	Char
Ô,ô	212,244	o_212_244	O circumflex	Char
Õ,õ	213,245	o_213_245	O tilde	Char
Ö,ö	214,246	o_214_246	O umlaut	Char
x	215	multiply	multiplication sign	Char
Ø,ø	216,248	o_216_248	oh stroke	Char
Ù,ù	217,249	u_217_249	U grave	Char
Ú,ú	218,250	u_218_250	U acute	Char
Û,û	219,251	u_219_251	U circumflex	Char
Ü,ü	220,252	u_220_252	U umlaut	Char
Ý,ý	221,253	y_221_253	Y acute	Char
Ɔ	222	char_222	character 222	Char
ß	223	ss	double s	Char
÷	247	divide	division sign	Char
ƚ	254	char_254	character 254	Char
ÿ,ÿ	159,255	y_159_255	character 159 or 255	Char

The table that follows lists the System Media File information for month values.

Table 23: System Media Files, Months

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types /When Media File Is Used
		January	January	Date
		February	February	Date
		March	March	Date
		April	April	Date
		May	May	Date
		June	June	Date
		July	July	Date
		August	August	Date
		September	September	Date
		October	October	Date
		November	November	Date
		December	December	Date

The table that follows lists the System Media File information for month values.

Table 24: System Media Files, Days

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types /When Media File Is Used
		Sunday	Sunday	DOW
		Monday	Monday	DOW
		Tuesday	Tuesday	DOW
		Wednesday	Wednesday	DOW
		Thursday	Thursday	DOW
		Friday	Friday	DOW
		Saturday	Saturday	DOW

The table that follows lists the System Media File information for month values.

Table 25: System Media Files, Time

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		hour	hour	Etime, 24TOD per locale, TOD per locale
		hours	hours	Etime,24TOD per locale,TOD per locale
		minute	minute	Etime
		minutes	minutes	Etime
		second	second	Etime,24TOD
		seconds	seconds	Etime,24TOD
		on	on	per locale(unused for en-us)
		at	at	per locale(unused for en-us)
		am	am	TOD
		pm	pm	TOD
		oclock	oclock	TOD

The table that follows lists the System Media File information for currency values.

**Note**

The customer's Media Administrator may want to replace the contents of "currency_minus" (for the negative amount) and "currency_and" (the latter can even be changed to contain silence).

Table 26: System Media Files, Currency

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		currency_minus	minus	Currency
		currency_and	and	Currency

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
\$	36	USD_dollar	dollar	Currency
		USD_dollars	dollars	Currency
		Note Unified CVP uses the USD_dollar.wav and USD_dollars.wav media files; the dollar.wav and dollars.wav used by ISN Version 1.0 are no longer installed.		
\$	36	CAD_dollar	dollar	Currency
		CAD_dollars	dollars	Currency
		HKD_dollar	dollar	Currency
		HKD_dollars	dollars	Currency
¢	162	cent	cent	Currency
		cents	cents	Currency
		euro	euro	Currency
£	163	GBP_pound	pound	Currency
		GBP_pounds	pounds	Currency
		penny	penny	Currency
		pence	pence	Currency
		MXN_peso	peso	Currency
		MXN_pesos	pesos	Currency
		centavo	centavo	Currency
		centavos	centavos	Currency

The table that follows lists the System Media File information for gaps of silence and miscellaneous phrases.

Table 27: System Media Files, Silence and Miscellaneous Phrases

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		silence_.1_sec	(.1 second of silence)	Used for pauses where needed
		silence_.25_sec	(.25 second of silence)	Used for pauses where needed

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		silence_.5_sec	(.5 second of silence)	Used for pauses where needed
		silence_1_sec	(1 second of silence)	Used for pauses where needed
		and	and	Etime,TOD,25TOD

The table that follows lists the System Media File information for ANSI characters.

Table 28: System Media Files, ANSI Characters

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
	32	space	space	Char
!	33	exclamation_mark	exclamation mark	Char
"	34	double_quote	double quote	Char
#	35	pound	pound	Char
%	37	percent	percent	Char
&	38	ampersand	ampersand	Char
'	39	apostrophe	apostrophe	Char
(40	open_parenthesis	open parenthesis	Char
)	41	close_parenthesis	close parenthesis	Char
*	42	asterisk	asterisk	Char
+	43	plus	plus	Char
,	44	comma	comma	Char
-	45	hyphen	hyphen	Char
.	46	period	period	Char
/	47	slash	slash	Char
:	58	colon	colon	Char
;	59	semicolon	semicolon	Char
<	60	less_than	less than	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
=	61	equal	equal	Char
	62	greater_than	greater than	Char
?	63	question_mark	question mark	Char
@	64	at_symbol	at	Char
[91	left_square_bracket	left square bracket	Char
\	92	backslash	backslash	Char
]	93	right_square_bracket	right square bracket	Char
^	94	caret	caret	Char
_	95	underscore	underscore	Char
`	96	single_quote	single quote	Char
{	123	open_brace	open brace	Char
	124	pipe	pipe	Char
}	125	close_brace	close brace	Char
~	126	tilde	tilde	Char
'	130	char_130	low single quote	Char
<i>f</i>	131	char_131	F with hook	Char
”	132	low_double_quote	low double quote	Char
...	133	ellipsis	ellipsis	Char
†	134	char_134	character 134	Char
‡	135	char_135	character 135	Char
^	136	char_136	character 136	Char
‰	137	per_mille	per mile	Char
Š	138	char_138	character 138	
<	139	left_pointing_angle	left pointing angle	Char
‘	145	left_single_quote	left single quote	Char
’	146	right_single_quote	right single quote	Char
“	147	left_double_quote	left double quote	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
”	148	right_double_quote	right double quote	Char
·	149	bullet	bullet	Char
—	150	en_dash	en dash	Char
—	151	em_dash	em dash	
~	152	small_tilde	small tilde	Char
™	153	trade_mark	trade mark	Char
š	154	char_154	character 154	Char
›	155	char_155	character 155	Char
¡	161	exclamation_mark_inverted	inverted exclamation mark	Char
▣	164	char_164	character 164	Char
⏏	166	broken_pipe	broken pipe	Char
§	167	section	section	Char
¨	168	char_168	character 168	Char
©	169	copyright	copyright	Char
^a	170	char_170	character 170	Char
«	171	left_double_angle_quote	left double angle quote	Char
¬	172	not	not	Char
-	173	char_173	character 173	Char
®	174	registered	registered	Char
—	175	char_175	character 175	Char
°	176	degree	degree	Char
±	177	plus_minus	plus or minus	Char
²	178	superscript_2	superscript two	Char
³	179	superscript_3	superscript three	Char
´	180	acute_accent	acute accent	Char
μ	181	micro	micro	Char

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
¶	182	paragraph	paragraph	Char
·	183	middle_dot	middle dot	Char
¸	184	cedilla	cedilla	Char
¹	185	superscript_1	superscript one	Char
°	186	char_186	character 186	Char
»	187	right_double_angle_quote	right double angle quote	Char
¿	191	question_mark_inverted	inverted question mark	Char

