Unified CVP Scripting

- Writing Scripts for Unified CVP, on page 1
- Before You Begin, on page 1
- Scripts to Access Unified CVP from Packaged CCE, on page 2
- Invoke Unified CVP Micro-applications Through Routing Scripts, on page 2
- Unified CVP Call Studio Scripting, on page 2
- Scripting for Unified CVP with Packaged CCE , on page 3
- Writing Packaged CCE Applications for Unified CVP, on page 12
- Unified CVP Micro-applications, on page 12
- Scripting for Unified CVP with Call Studio, on page 54

Writing Scripts for Unified CVP

This section discusses using Packaged CCE configuration and script editing to access the Unified CVP solution.

It includes information about how to:

- Set up Packaged CCE to interact with Unified CVP
- Write applications for Unified CVP

Note This section contains important information for Unified CVP application developers. It also may be of interest to Call Center Managers, Unified CVP System Managers, and Packaged CCE system managers.

Before You Begin

This chapter makes the following assumptions:

- The information in this chapter assumes that you are already familiar with using the Unified CCE Administration and Script Editor tools for call center operations and management.
- When creating Script Editor 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. Following this practice will avoid potential issues with data transfer between different systems.

Scripts to Access Unified CVP from Packaged CCE

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

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

- **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 Unified CVP applications.

Note For more information, refer to Scripting for Unified CVP with Call Studio, on page 54.

Invoke Unified CVP Micro-applications Through Routing Scripts

The 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 Packaged CCE routing script to produce a viable voice interaction with the caller.

Instead of developing full scale Unified CVP applications using micro-applications, use Unified CVP scripts developed using Call Studio to create the Unified CVP applications. 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.

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

Unified CVP Call Studio Scripting

Sophisticated Unified CVP 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 for execution. To invoke a VXML Server application, the script writer includes a Get Speech (GS) micro-application via the Run External Script node in the Packaged CCE routing script. This micro-application instructs the VoiceXML Gateway to interact with the VXML Server directly to run the application. The final results are passed back to Packaged CCE.

Some of the Call Studio scripting environment features include:

- A drag-and-drop interface with a palette of Unified CVP functions

- The ability to do database queries
- Extensibility with Java code written to perform any task a Java application can perform

Note Packaged CCE does not support using the *MicroApp* nodes that are available in the Script Editor. All MicroApp implementation must be done using the *Run External Script* node in Script Editor. Refer to Writing Packaged CCE Applications for Unified CVP, on page 12 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 Writing Packaged CCE Applications for Unified CVP, on page 12.

Scripting for Unified CVP with Packaged CCE

The sections that follow include:

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

Micro-applications

Micro-applications are a set of specific Unified CVP functions that can be invoked by Packaged CCE, 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 media file and retrieves digits from the caller.
- **Menu**. Plays a media menu file and retrieves a single telephone keypad entry from the caller.
- **Get Speech**. Runs a Call Studio script on VXML Server.
- **Capture**. The Capture (CAP) micro-application enables you to trigger the storage of current call data at multiple points in the Packaged CCE routing script.

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

Note Using ASR/TTS(speech) through micro-applications is not supported. You have to use Call Studio scripts for any caller interaction that requires use of ASR/TTS (speech).

Simple Example Script: Welcome to XYZ Corporation

Suppose you want to create a script that has an example call flow as follows.

This simple script performs the following functions:

- Runs the GetInfo Call Studio script on the VXML Server to collect some caller input based on the example callflow.
- Based on caller input, queues for a sales or support agent .
- If an agent is not available, runs the MOH Call Studio script which will play music-on-hold to the caller until an agent becomes available.

Procedure

Step 1 A call arrives at Packaged CCE and runs a Packaged CCE script.

Step 2 The caller hears a welcome prompt.

Step 3 The script sends the call to Unified CVP for collecting some information from the caller before queuing the call for an agent. For example, a menu is offered such as "press 1 for sales and 2 for support," as well as entering an account number.

Step 4 If the caller is an existing customer, the caller-entered account number is used to retrieve additional information about the caller from an external database.

Step 5 Caller-entered digits and the additional information about the caller are returned back to the Packaged CCE script to be shown a screen pop to the agent, when an agent becomes available.

Step 6 The call is then queued waiting for an agent in a particular skill group, based on the caller selection of the type of service.

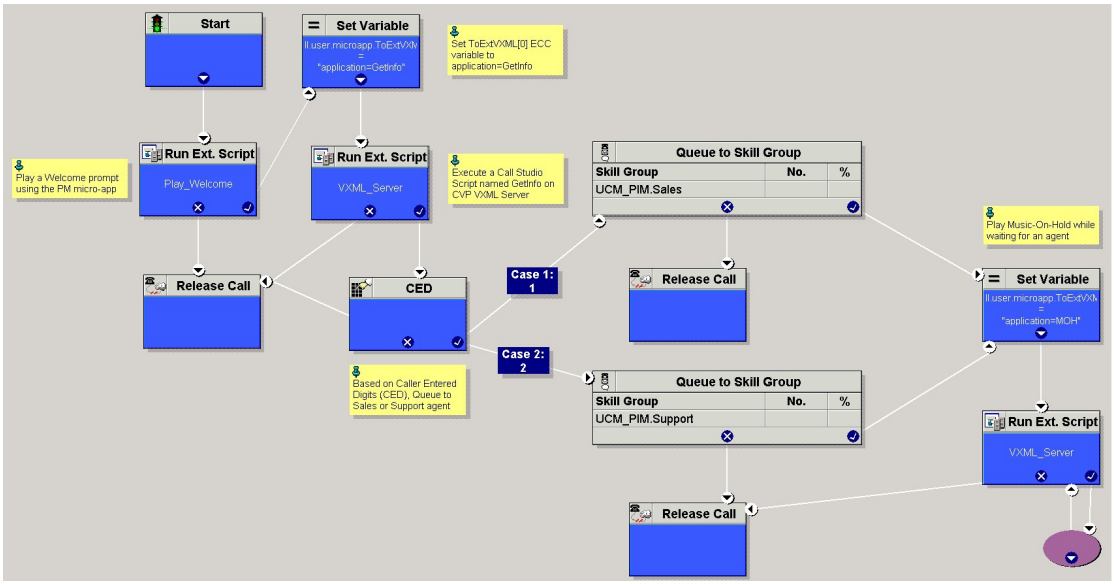
Step 7 If an agent is available, the caller is connected to that agent. The agent desktop displays the caller information collected via caller input as well as database lookup.

Step 8 If an agent is not available, the call is sent back to Unified CVP for playing music-on-hold while the caller waits for an agent to become available.

Step 9 The information collected from the caller is preserved as call context on the call until the agent becomes available.

You can create a script such as the one shown in the following figure.

Figure 1: Packaged CCE Script with Call Flow



This simple script performs the following functions:

- Runs the GetInfo Call Studio script on the VXML Server to collect some caller input based on the example callflow.
- Based on caller input, queues for a sales or support agent.
- If an agent is not available, runs the MOH Call Studio script which will play music-on-hold to the caller until an agent becomes available.

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

Packaged CCE Unified CVP Micro-app Connection

Before the Unified CVP can be accessible through the Script Editor’s Run External Script node, you must first set up Packaged CCE with special Unified CVP parameters using the Unified CCE Administration tool.

Begin by using the Unified CCE Administration Network VRU Script tool to define Unified CVP parameters. See Network VRU Scripts

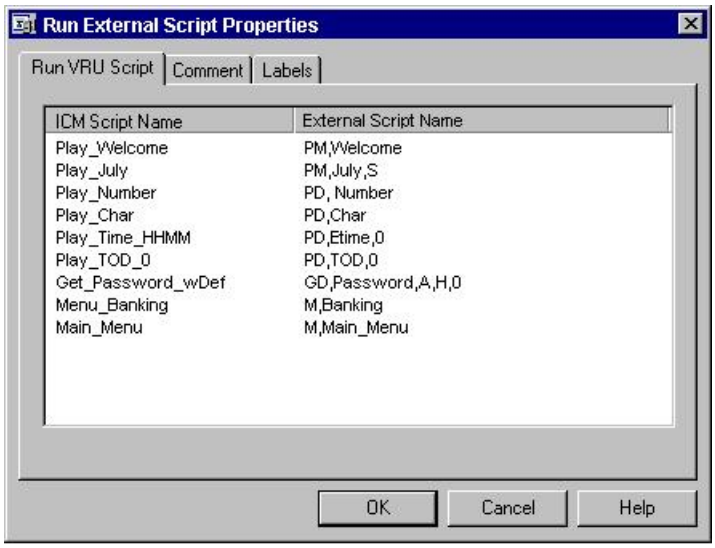
Note As shown in the two columns of the following table, certain entries for the VRU Script Name and Configuration Param fields are case-sensitive.

Attribute	Allowed Values	Applies to	Case-Sensitive?
Attribute: VRU Script Name (for example, PM, GD).	PM, GD	All micro-applications	N
Attribute: Media Library Type (A, S, V)	A, S, V	All micro-applications	N
Barge-in Allowed	Y/N	All micro-applications	N
Data playback type	Number, Char	PlayData (PD)	N
Time Format	HHMM, HHMMSS, HHMMAP	PlayData (PD)	N
Timeout Message Override	Y/N	Get Digits (GD), Get Speech (GS), Menu (M)	N
Invalid Entry Message Override	Y/N	Get Digits (GD), Get Speech (GS), Menu (M)	N
DTMF Termination Key	N	All micro-applications	N
Media File Name		All micro-applications	Y

Once the network VRU script configuration 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.

Figure 2: Run External Script Node



Information Exchange Between Packaged CCE and Unified CVP

When Packaged CCE 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 Unified CVP parameters are passed to Unified CVP through Expanded Call Context (ECC) variables and/or Call.Peripheral variables. Other parameters are sent in the usual VRU messaging interface (Packaged CCE /Unified CVP Service Control Interface).

Packaged CCE 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 typically 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 Packaged CCE scripts:

ASR error

Failure of an Advanced Speech Recognition component.

General error

General error occurred.

Invalid configuration param

Data passed from Packaged CCE to the Unified CVP 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 Packaged CCE to the Unified CVP 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 Unified CVP Service did not recognize. ECC variable definitions must be the same in Packaged CCE 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 Packaged CCE to the Unified CVP Service did not exist on the Media Server.

Unknown micro-application

Micro-application name passed from Packaged CCE to the Unified CVP Service did not exist on the Unified CVP Service.

Unsupported locale

The VoiceXML Interpreter (that is, gateway) did not recognize the locale passed from the Unified CVP Service.

Unsupported VoiceXML element

The VoiceXML Interpreter (that is, gateway) did not recognize a VoiceXML element passed from the Unified CVP Service, VXML Server, or media server.

Unsupported VoiceXML format

The VoiceXML Interpreter (that is, gateway) did not recognize a VoiceXML format passed from the Unified CVP Service, VXML Server, or media server.

Each Unified CVP micro-application has individualized settings for **user.microapp.error_code**, 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
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: • Media file does not exist. • Invalid URL for Media file.	One of the following: • Media file does not exist • Invalid URL for Media L file	One of the following: • Media file does not exist • Invalid URL for Media L file	One of the following: • Media file does not exist • Invalid URL for Media L file	One of the following: • Media file does not exist • Invalid URL for Media file	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

Error Code	Play Media	Play Data	Get Digits	Menu	Get Speech	Capture
13	N/A	Variable data is invalid	N/A	N/A	N/A	N/A
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 ended the call	Called party ended the call	Called party ended the call	Called party ended the call	Called party ended the call	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

Error Code	Play Media	Play Data	Get Digits	Menu	Get Speech	Capture
41	VXML Server application error	N/A	N/A	N/A	VXML Server application error	N/A
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.)

Note However, if a configuration error, or a network or component failure of some sort, prevents the micro-application from being run 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.

Writing Packaged CCE Applications for Unified CVP

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

You do this using two Packaged CCE software tools:

- Unified CCE Administration
- Packaged CCE Script Editor

Use Unified CCE Administration to configure Unified CVP Network VRU scripts. The following section describes using the Script Editor to access Unified CVP micro-applications.

Related Topics

Add and Maintain Network VRU Scripts

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 Scripts tool.

Step 2 Select the **ICM Script/VRU Script Name** you want to run.

Step 3 Modify the Comments tab as needed.

Step 4 Modify the Labels tab as needed.

Step 5 When finished, click **OK** to submit the changes and close the dialog box.

Unified CVP Micro-applications

The sections that follow describe the parameters that can be defined through Unified CCE Administration 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 theNetwork VRU Script 's Attributes tab must be separated by a comma.
- If a parameter value is not specified, the micro-application uses its default.

Dynamic Audio File Support for Micro-applications

Unified CVP lets you use a single micro-application and specify the prompt using call variables and the Packaged CCE 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, and Get Digits micro-applications.

Note When A is specified as the Media Library, it means Unified CVP looks for the media file under the C:\inetpub\wwwroot\en-us\app folder by default and when S is specified, it looks under the C:\inetpub\wwwroot\en-us\sys folder by default.

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,A	**PM** - Uses the Play Media micro-application. **-3** - Plays the file specified in Call.PeripheralVariable3. **A** - Acquires the file from the application media files folder (for example, C:\inetpub\wwwroot\en-us\app).

Notes

- If you do not specify a file extension for the file name in the Call.PeripheralVariable, the default media file extension is applied (for example, .wav for audio files).
- 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 Unified CVP Service creates a VoiceXML that does not contain a media prompt.
- 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

You can specify a media server for a micro-application was to use the ECC variable `user.microapp.media_server`.

The global default media server can be specified in **Unified CCE Administration** > **Overview** > **Infrastructure Settings** > **Device Configuration** > **CVP Server** > **IVR** tab.The default media server is used by the micro-applications if the ECC variable `user.microapp.media_server` is missing or empty in the Packaged CCE 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

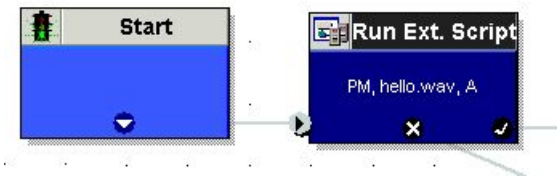
- Play Media (PM)
- Play Data (PD)
- Get Digits (PD)
- Menu (M)

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

Figure 3: Packaged CCE Script with Play Media Using ECC Variable

The following screen shot shows the Packaged CCE script where Play Media micro-application plays a media file using a default media server.

Figure 4: Packaged CCE Script with Play Media Using Default Media Server



Capture Micro-application

The Capture (CAP) micro-application allows you to trigger the storage of current call data at multiple points in the Packaged CCE routing script. The CAP micro-application must be configured as a VRU script, and it is run 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 Packaged CCE 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.

Packaged CCE 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 Packaged CCE. 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.
- 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 Unified CVP VRU leg.

The Capture micro-application places a heavy demand on Packaged CCE resources. Each time you use it, Packaged CCE 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 Packaged CCE in terms of processing time and disk space, which despite the minimal indexing, may impact Packaged CCE ’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 Packaged CCE Administration’s Network VRU Scripts tool to specify parameters.

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).
  - **null** - (default) If this field 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.
  - **-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 Packaged CCE .

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

  The valid options are:

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

- **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.

  For more information about barge-in, see How Unified CVP Handles Barge-In, on page 17.

- **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.

  For more details, see *Configure Play Media Micro-application to Use Streaming Audio*.

- **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 usually used when two or more PM and/or PD microapps are used in a loop in the Packaged CCE 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 Packaged CCE script when the user barges in.

How Unified CVP Handles Barge-In

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, on page 47

Configure Play Media Micro-application to Use Streaming Audio

Use the 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.

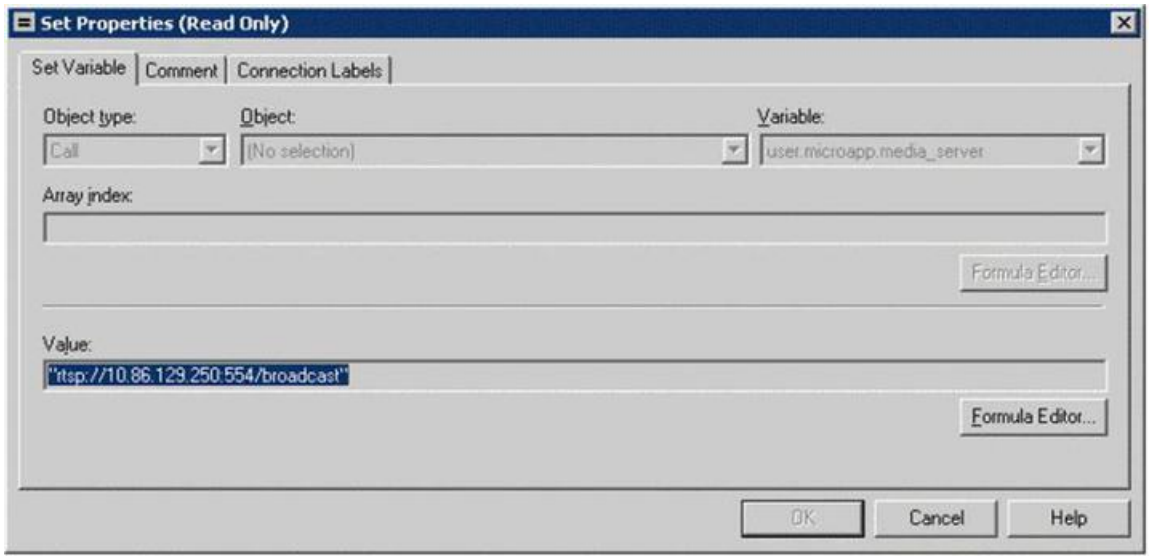
Note The IOS gateway only supports µ-law wav files in 8-bit format.

You must enclose the stream URL and stream name values in quotation marks.

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.



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

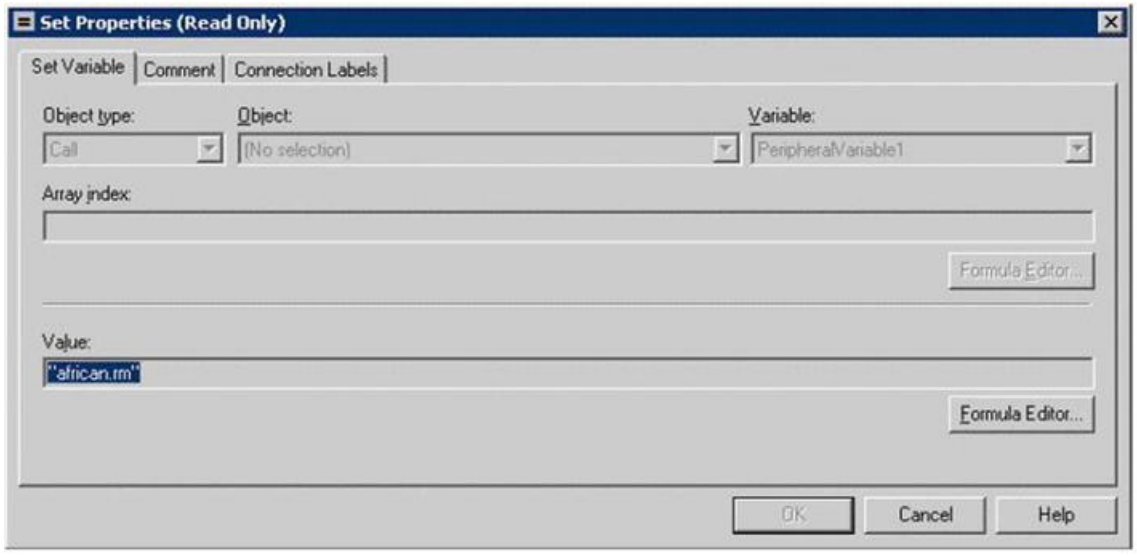
  Note
  The URL must begin with an *rtsp://* prefix (Real-time Streaming Protocol) to stream audio over the network. A trailing forward slash is not permitted in the URL.

- 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 Peripheral Variables is PeripheralVariable1 through PeripheralVariables10.



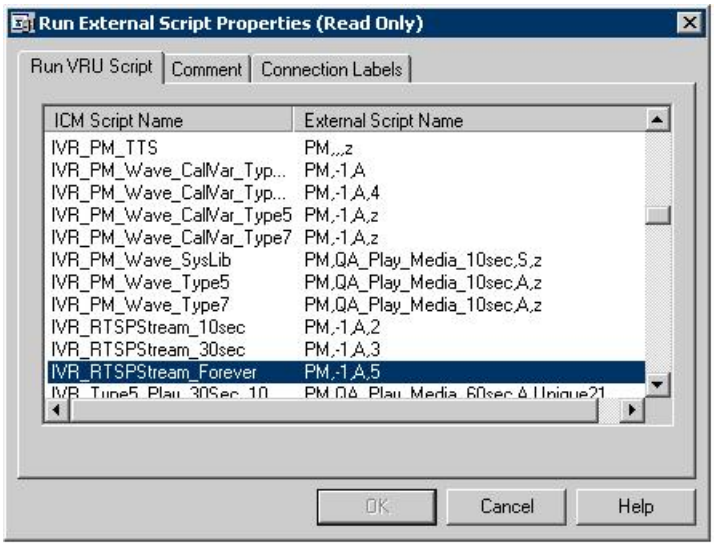
- In the Value field, specify the stream name and click **OK**.

  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 Unified CVP_RTSPStream_Forever script's external script name contains four parameters: PM, -1, A, 5. The second parameter, **-1**, instructs Unified CVP to play the stream name declared in **PeripheralVariable1** (shown in Step 2). 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 CCE Script Editor to configure CCE 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 is run and the stream plays from the targeted streaming server and proceeds generally.

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

Step 5 Optionally, you can use the Packaged CCE Administration's Network VRU Scripts tool 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 standard value is 10 seconds.

If the SIP Call with Unified CVP 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 CVP Service, and as a result a code 38 rejection is generated in the Gateway logs.

Related Topics

Configuring Custom Streaming Ringtones , on page 20

Configure Custom Streaming Ringtones

You can configure custom ringtone patterns that enable you to play an audio stream to a caller in place of the usual ringtone. Customized streaming ringtones are based on the dialed number destination and, when configured, play an in-progress broadcast stream to the caller while the call is transferred an agent.

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 *is not* 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 *is* allowed.

Example	Field Name	Field Contents	Tells Unified CVP...
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 *is* allowed.
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 *is not* allowed.
5	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 *is not* allowed.
6	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 *is not* 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* topic.

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 Packaged CCE Peripheral Variables (PeripheralVariable1 through PeripheralVariables10).
- The **user.microapp.play_data** elements.

Configure Network VRU Script Settings for Play Data Micro-application

Use the Unified CCE Administration Network VRU Script tool's Attributes tab to specify parameters.

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.

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.

Before you begin

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, on page 24.

- **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, on page 22.

- **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.

  For more information about barge-in, see Play Data and Data Storage, on page 22.

- **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 generally used when two or more PM and/or PD microapps are used in a loop in the CCE 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 CCE 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. 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.

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 Unified CVP 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, on page 27.

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.	-###############.###### The leading minus (-) is optional and is played as "minus." 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). The decimal point is represented as a period (.) and played as "point." It is optional if there is no floating portion. The floating point portion of the number is optional and can contain a maximum of six digits. Trailing zeros are played.	**en-us** and **en-gb** typical spoken form: • -123 = "minus one hundred twenty three" • 35.67 = "thirty five point six seven" • 1234.0 = "one thousand, two hundred, thirty four point zero" **es-mx** and **es-es** typical spoken form: • -120 = "menos ciento veinte" • 10.60 = "diez coma seis cero" • 1,100 = "mil cien"
Char	Play the stored data as individual characters.	All printable American National Standards Institute (ANSI) characters are supported. **Note** Code Page 1252 is ANSI standard. It contains ASCII (characters 0-127) and extended characters from 128 to 255	**en-us** and **en-gb** typical spoken form: • abc123= "A, B, C, one, two, three" **es-mx** and **es-es** typical spoken form: • 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: • MMDDYYYY format: 20000114 = “January fourteenth, two thousand” **en-gb** typical spoken form: • DDMMYYYY format: 20000114 = “Fourteenth of January, two thousand” **es-mx** and **es-es** typical spoken form: • 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: • HHMM format: 0830= “eight hours thirty minutes” • HHMMSS format: 083020= “eight hours, thirty minutes, twenty seconds” **es-mx** and **es-es** typical spoken form: • 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	**en-us** and **en-gb** typical spoken form: • 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.” **es-mx** and **es-es** typical spoken form: • HHMM format: 0100 = “una a.m.” • HHMMAP format: 1203 = “doce y tres p.m.” • HHMMSS format: 242124 = “doce veintiuno a.m.”
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: • 7 = “Saturday” **es-mx** and **es-es** typical spoken form: • 7 = “Sabado”

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 4: 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

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		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
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	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		24	twenty-four	
		25	twenty-five	
		26	twenty-six	
		27	twenty-seven	
		28	twenty-eight	
		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	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		51	fifty-one	
		52	fifty-two	
		53	fifty-three	
		54	fifty-four	
		55	fifty-five	
		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	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		78	seventy-eight	
		79	seventy-nine	
		80	eighty	
		81	eighty-one	
		82	eighty-two	
		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

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		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 5: 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	nineth	
		10ord	tenth	
		11ord	eleventh	
		12ord	twelveth	
		13ord	thirteenth	
		14ord	fourteenth	
		15ord	fifteenth	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		16ord	sixteenth	
		17ord	seventeenth	
		18ord	eighteenth	
		19ord	nineteenth	
		20ord	twentieth	
		21ord	twenty-first	
		22ord	twenty-second	
		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 6: 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

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
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
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

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
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
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

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
Þ	222	char_222	character 222	Char
ß	223	ss	double s	Char
÷	247	divide	division sign	Char
þ	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 7: 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 8: 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

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
		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 9: 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 prefer to replace the contents of "currency_minus" (for the negative amount) and "currency_and" (the latter can even be changed to contain silence).

Table 10: 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
$	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 11: 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
		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 12: 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

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
/	47	slash	slash	Char
:	58	colon	colon	Char
;	59	semicolon	semicolon	Char
<	60	less_than	less than	Char
=	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
f	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
ˆ	136	char_136	character 136	Char
‰	137	per_mille	per mile	Char
Š	138	char_138	character 138	

Symbol (where applicable)	Decimal Value	Media File Name	Media File Content	Data Play Back Types / When Media File Is Used
<	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
”	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

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

Play Data Configuration Examples

The following table shows several configuration examples for Play Data.

Table 13: 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.

If the VRU Script Name field setting is…	It means…	If the Configuration Param field is…	It means…
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* topic.

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 Packaged CCE for further processing using the Caller-Entered Digits (CED) field in the CCE/Unified CVP Messaging interface. (This is available in the Packaged CCE script through the variable Call.CallerEnteredDigits).

Configure Network VRU Script Settings for Get Digits Micro-application

Use the Unified CCE Administration Network VRU Script tool 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 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, 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
- **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, on page 47.

- **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

  For more information about barge-in, see How Unified CVP Handles Barge-In, on page 17.

  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, on page 47.

- **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**).
- **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
- **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, see System Media Files, on page 27.

- **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, on page 47.

- **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
If the value is set to 0, the Unified CVP 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 14: Get Digits Configuration Examples for .wav Files

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.
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, on page 47.			

If the VRU Script Name field setting is…	It means…	If the Configuration Param field setting is…	It means…
GD, -4, S	**gd** - Use the Get Digits micro-app **-4** - Calls the file specified in Call.PeripheralVariable4 **S** - Acquires the file from the System media library	6,12,	**6** - Minimum field length **12** - Maximum field length Accept defaults for all other settings

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 Unified CVP 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 Unified CVP 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 Packaged CCE .

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 Packaged CCE 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 >= (greater than or equal to) the minimum length and <= (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 Unified CVP Service, SIP Service discards the termination key (only the password digits are included in the CED returned to Packaged CCE).

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 Packaged CCE 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 Packaged CCE .

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 Packaged CCE .

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 Packaged CCE for further processing using the Caller-Entered Digits (CED) field in the Packaged CCE / Unified CVP Messaging interface.

Configure Network VRU Script Settings for the Menu Micro-application

Use the Packaged CCE Administration Network VRU Script tool 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 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
- **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

  For more information about barge-in, see How Unified CVP Handles Barge-In, on page 17.

- **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, on page 27

Menu Configuration Examples

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

Table 15: Menu Configuration Example - DTMF Application

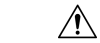
If the VRU Script Name field setting is…	It means...	If the Config Param setting is...	It means...
M,Banking	**M** - Use the Menu micro-app. **Banking** - Play the Media file named "Banking.wav." **Note** This file may contain a message such as: "For Checking, press 1. For Savings, press 2. For Money Market, press 3."	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).
M,Main_Menu	**M** - Use the Menu micro-app. **Main_Menu** - Play the Media file called "Main_Menu.wav." **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."	0-2/9,,4,2,2	**0-2/9** - Accept numbers 0, 1, 2, and 9. **,** (Skipped parameter) - Accept the default barge-in setting (Y). **4** - No Entry Timeout value (in seconds). **2** - Number of no entry tries allowed. **2** - Number of invalid tries allowed. Accept all other defaults (Timeout Msg Override, Invalid Entry Msg Override).
M,-2,S	**M** - Use the Menu micro-app. **-2** - Plays the file specified in Call.PeripheralVariable2. **S** - Acquires the file from the System media library.	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* topic.

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 Packaged CCE 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 CCE 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 Packaged CCE .
- 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 Packaged CCE .

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 Packaged CCE .

Get Speech Micro-application

The Get Speech (GS) micro-application is used to run a Call Studio script on VXML Server.

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

Use the Packaged CCE Administration’s Network VRU Script tool to specify parameters.

Note By default a pre-configured network VRU script called VXML_Server has already been configured in Packaged CCE. This should be used in all Run External Script nodes that intend to run a Call Studio script. When using an optional feature like Courtesy Callback, you must configure additional GS network VRU scripts.

Procedure

Step 1 Configure VRU Script field parameters:

- **Micro-application type**. For Get Speech, valid options are: **GS** or **gs**.
- **Media File Name**. Only the value **Server** is supported for this field for GS.
- **Media Library Type**. Only the value **V** is supported for this field for GS.
- **Uniqueness value**. Optional. A string identifying a VRU script name as unique.

Step 2 Configure the Configuration Param field parameters:

Note
Configuration parameters 1-10 are only for non-Packaged CCE deployments with Unified CVP where GS is supported with external VXML. Only the Pass FTP Information parameter (parameter 11) is configurable when using the Agent Greeting recording feature.

- **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. 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 *Elements Specifications for Cisco Unified CVP VXML Server and Cisco Unified Call Studio* guide for more information.
  - **ftpPath** - A 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`

Passing Information to the Call Studio Scripts Executing on VXML Server

You can pass up to 1050 characters to the Call Studio scripts executing on VXML server by using an ECC Variable array.

Table 16: 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 17: Sample Array Definition

Variable Name	Values
user.microapp.ToExtVXML[0]	"Company=Cisco;Job=technical writer"
user.microapp.ToExtVXML[1]	"Location=Boxborough;Street=Main"
user.microapp.ToExtVXML[2]	"FirstName=Gerrard;LastName=Thock"
user.microapp.ToExtVXML[3]	"Commute=1hour;Car=Isuzu"

Unified CVP sends each name/value pair as a session variable on the call to VXML server (for example, a session variable named **Company** with a value of **Cisco**). The session variables are accessible in the Call Studio scripts.

Passing Data Back to Packaged CCE from the VXML Server

Unified CVP can return 840 characters from the VXML server.

The following ECC Variable array is added:

Table 18: 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 840 characters by populating the **user.microapp.caller_input** variable and each element of the **user.microapp.FromExtVXML** array.

Note By default user.microapp.FromExtVXML ECC variable is pre-defined for Packaged CCE but not enabled. You can use the predefined ECC variable or update the length based on your needs.

Scripting for Unified CVP with Call Studio

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

To invoke a VXML Server application, create a Packaged CCE routing script that

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

This section describes

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

High-Level Configuration Instructions

This chapter presents a set of high-level instructions for configuring many of the Unified CVP call flow models (deployment models).

Each set of call flow model instructions contains:

- A brief overview of that call flow model
- High-level instructions for configuring the components in that call flow model
- References to detailed instructions (elsewhere in this guide, in online help, or in other documents) for performing each high-level task

This chapter also includes information, or pointers to information, for configuring the Gateway, Packaged CCE VRU handling and Unified CVP Call Server (including the SIP Service, Packaged CCE service, and Unified CVP Service).

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 Packaged CCE script. The ReqICMLabel must be inserted into a Call Studio script as a decision element. In Call Studio, the returned Packaged CCE 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 Packaged CCE 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 Packaged CCE is {Data.Element.GetICMLabel.result}, where *result* is the variable of the ReqICMLabel element that contains the Packaged CCE label.

Table 19: 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 thePackaged CCE server. This setting can be a maximum of 40 characters. The Packaged CCE 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 Packaged CCE 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 Packaged CCE Label request, regardless of whether or not these variables are filled by the Packaged CCE script. You need two sets of these variables to keep reporting to the Packaged CCE Call Peripheral Variables separate from what is returned from Packaged CCE.
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 Packaged CCE Packaged CCE 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 Packaged CCE script. The Packaged CCE server returns a string of name-value pairs, separated by semicolons, for up to five external VoiceXML variables.
Timeout	Integer	Yes	Yes	Yes	3000 (ms)	The number of milliseconds that the transfer request waits for a response from the Packaged CCE server before timing out. **Note** This value is increased or decreased by increments of 500 ms.

Name (Label)	Type	Required	Single Setting Value	Substitution Allowed	Default	Notes
caller_input (Caller Input)	String	No	Yes	Yes		This setting can be a maximum of 210 characters. The caller_input is only passed to Packaged CCE from Call Studio.

Table 20: Element Data

Name	Type	Notes
result	String	Packaged CCE label returned from a Packaged CCE 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.ReqICMLabelElement.result}.
callvar<*n*>	String	Call Peripheral variables that the Call Studio scripts passes to the Packaged CCE server. Valid Call Peripheral Variables are callvar1 - callvar10.
callvarReturn<*n*>	String	Call Peripheral variables that the Packaged CCE script returns to the VXML Server. Valid Call Peripheral Variables are callvarReturn1 - callvarReturn10. For example, if a Packaged CCE 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 Packaged CCE script by using Data.Element.ReqICMLabelElement.callvarReturn3 The returned value is "Cisco Systems, Inc."

Table 21: Session Data

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

Table 22: 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 CCE 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 CCE script with ECC variables configured for VXML Server
- Creating a VRU Script to run in the Packaged CCE script

Integrate Call Studio Scripts with Packaged CCE Scripts

The following steps describe how to integrate Call Studio scripts with Packaged CCE :

Procedure

Step 1 Set the user.microapp.ToExtVXML[0] ECC variable to application=HelloWorld.

Note
This example indicates that the VXML Server runs the “HelloWorld” application. To run a different application, change the value of user.microapp.ToExtVXML[0] accordingly.

Step 2 Create a Run External Script node within the Packaged CCE 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 Packaged CCE 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

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 Call Studio, use the **Deploy Remotely** option to deploy the scripts to an FTP Server.

Deploy Call Studio Scripts Using Call Studio

Procedure

Step 1 Create or modify one or more VoiceXML application scripts.

Step 2 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 CVPSNMPLogger. 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.

Step 3 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.