Media File Options

Deployment and Management of Voice Prompts

You can deploy voice prompts using following approaches:

- Local File System

  The voice prompt files are stored on a local system and audio prompts are retrieved without using bandwidth. With this approach, VoiceXML Gateways do not have to retrieve audio files for playing prompts, so WAN bandwidth is not affected. However, if a prompt needs to be changed, you must change it on every VoiceXML Gateway.

  - IOS VoiceXML Gateway—prompts are deployed on flash memory.

    IOS VoiceXML Gateway can either be VoiceXML Gateway or PSTN Gateway, which has Ingress Voice Gateway and VoiceXML Gateway colocated. Store only critical prompts such as error messages or other messages that can be used when the WAN is down.

    When recorded in G.711 mu-law format, typical prompts of average duration are about 10 to 15 KB in size. When sizing gateways for such implementations, size the flash memory by factoring in the number of prompts and their sizes, and also leave space for storing the Cisco IOS image.

  - Cisco VVB—prompts are installed on local file system.

    Built-in CVP prompts are packaged with Cisco VVB product and installed during installation. You can change *Error* tone default prompt through Cisco VVB Administrator console.

- Media Server

  Each local VoiceXML Gateway, if configured properly, can cache many or all prompts, depending on the number and size of the prompts (up to 2 GB for Cisco VVB and 100 MB for IOS). The best way to test whether your Media Server is appropriately serving the media files is to use a web browser and specify the URL of a prompt on the Media Server, such as *http://10.4.33.130/en-us/sys/1.wav*. Your web browser should be able to download and play the .wav file without any authentication.

The design of Media Server deployment depends on the following factors:

- Number of media files to be played on each gateway.
- Network connectivity between the gateway and the Media Server.
- Frequency in which the media files are changed.

Media File Deployment Design Concepts

The concepts described in this section are relevant to media file deployment design.

Bandwidth Calculation for Prompt Retrieval

When prompts are stored on an HTTP media server, the refresh period for the prompts is defined on that server. The bandwidth consumed by prompts consists of the initial loading of the prompts at each VoiceXML Gateway and of the periodic updates at the expiration of the refresh interval.

To calculate the bandwidth that your prompts consume, multiply the number of prompts by average size of each prompt. As an example of determining the bandwidth consumed by prompts, assume that a deployment has 50 prompts with an average size of 50 KB (50,000 bytes) each. Also assume that the refresh period for the prompts is defined as 15 minutes (900 seconds) on the HTTP media server. The WAN bandwidth required for prompts in this deployment can be calculated as follows:

(50 prompts) * (50,000 bytes/prompt) * (8 bits/byte) = 20,000,000 bits

(20,000,000 bits) / (900 seconds) = 22.2 kbps per branch

TCP Socket Persistence

Unified CVP does not support TCP socket persistence.

WAN Acceleration

The Cisco Wide Area Application Services (WAAS) system consists of a set of devices called Wide Area Application Engines (WAEs) that work together to optimize TCP traffic over your network. Cisco WAAS uses a combination of TCP optimization techniques and application acceleration features to overcome the most common challenges associated with transporting traffic over a WAN. Cisco WAAS deployed at the periphery of the network on the VoiceXML Gateway side performs the following functions:

- Makes changes in TCP header to optimize the traffic.
- Acts as a large HTTP cache located locally.
- Reduces the traffic more using compression algorithms.
- Reduces traffic by using Data Redundancy Elimination (DRE) techniques.

Cisco WAAS is deployed in inline mode where whole data is forced to pass through the Cisco WAAS.

IOS Gateway Media File Deployment

Cisco IOS Caching and Streaming

The Cisco IOS VoiceXML Gateway uses an HTTP client, which is a part of Cisco IOS. The client fetches VoiceXML documents, audio files, and other file resources.

Caching and streaming are two key properties associated with playing audio prompts. These two properties are closely related to each other, and they can affect system performance greatly when the router is under load.

Streaming and Non-Streaming Modes

In non-streaming mode, the entire audio file must be downloaded from the HTTP server onto the router before the Media Player can start playing the prompt. This implies a delay for the caller. If the audio file is relatively small, the caller will not notice any delay because downloading a small file takes only a few milliseconds. The delay caused by loading larger files can be overcome by using either caching or streaming mode.

In streaming mode, the Media Player streams the audio in media chunks from the HTTP server to the caller. As soon as the first chunk is fetched from the server, the Media Player can start playing. The advantage of streaming mode is that there is no noticeable delay to the caller, regardless of the size of the audio prompt. The disadvantage of streaming mode is that, because of all of the back-and-forth interactions from fetching the media file in chunks, the performance deteriorates. Additionally, the ability to cache the files in memory reduces the advantage of streaming large files directly from the HTTP server.

For recommendations on when to use streaming and non-streaming mode for prompts, see section Design Considerations for Large Number of Media Files, on page 6.

Cache Types

There are two types of cache involved in storing media files: the IVR Media Player cache and the HTTP Client cache.

The HTTP Client cache is used for storing files that are downloaded from the HTTP server. In nonstreaming mode, the entire media file is stored inside the HTTP Client cache. In streaming mode, the first chunk of the media file is stored in the HTTP Client cache and in the IVR cache, and all subsequent chunks of the file are saved in the IVR cache only. The HTTP Client cache can store 100 MB of prompts, while the IVR cache is limited to 32 MB.

Use only nonstreaming mode, so that the IVR prompt cache is never used and the HTTP Client cache is the primary cache. In nonstreaming mode, the HTTP Client cache can also store 100 MB of prompts, while the IVR cache is limited to 16 MB.

To configure the HTTP Client cache, use the following Cisco IOS commands:

http client cache memory file *1-10000*

The 1–10000 value is the file size in kilobytes. The default maximum file size is 50 KB, but you can also have a file size up to 600 KB file size. Any file that is larger than the configured HTTP Client memory file size will not be cached.

http client cache memory pool *0-100000*

The 0–100000 value is the total memory size available for all prompts, expressed in kilobytes. A value of zero disables HTTP caching. The default memory pool size for the HTTP Client cache is 10 MB. The memory pool size is the total size of all prompts stored on the media server, which is up to 100 MB.

Query URL Caching

A query is a URL that has a question mark (?) followed by one or more **name=value** attribute pairs in it. The Unified CVP VXML Server uses query URLs extensively when generating the dynamic VoiceXML pages that are rendered to the caller. Because each call is unique, data retrieved from a query URL can waste cache memory and a possible security risk, because the query URL can contain information such as account numbers or PINs.

Query URL caching is disabled by default in Cisco IOS. To ensure that it is disabled, enter a **show run** command in Cisco IOS and ensure that the following Cisco IOS command does not appear:

Gateway configuration: **http client cache query**

Cisco VVB Media File Deployment

Caching and Query

Cisco VVB uses an HTTP client, which is a part of the product. The client fetches VoiceXML documents, audio files, and other file resources.

Caching property is associated with VXML resources, audio prompts, grammar and script files.

A query is a URL that has a question mark (?) followed by one or more **name=value** attribute pairs in it. By default, Query URLs are not cached.

Cache Aging

The HTTP Client manages its cache by the freshness of each cached entry. Whether a cached entry is fresh or stale depends on two numbers: Age and FreshTime. Age is the elapsed time since the file was last downloaded from the server. FreshTime is the duration that the file is expected to stay in the HTTP Client cache since the file was last downloaded.

Several variables that can affect the FreshTime of a file, such as HTTP message headers from the server and the cache refresh value configured using the command line interface (CLI).

The FreshTime of a file is determined in the following sequence:

1. When a file is downloaded from the HTTP server, if one of the HTTP message headers contains the following information, the max-age is used as the FreshTime for this file:

   Cache-Control: max-age = <*value in seconds*>

2. If Step 1 does not apply, but the following two headers are included in the HTTP message, the difference (Expires – Date) is used as the FreshTime for this file:

   Expires: <*expiration date time*>

   Date: <*Current date time*>

3. The HTTP/1.1 specification, RFC 2616 (HyperText Transport Protocol), recommends that either one of the HTTP message headers as described in Step 1 or 2 should be present. If the server fails to send both 1 and 2 in its HTTP response, then take 10 percent of the difference between Date and Last-Modified from the following message headers:

   Last-Modified: <*last-modified date time*>

   Date: <*Current date time*>

So the FreshTime for this file is calculated as:

FreshTime = 10% * ([Date] – [Last-Modified])

4. The CLI allows the user to assign a FreshTime value to the files as a provisional value:

   http client cache refresh *1-864000*

   The default refresh value is 86400 seconds (24 hours). The configured HTTP Client cache refresh has no effect on files when any of the message headers in steps 1 to 3 are present. If the resultant FreshTime from the CLI command calculation turns out to be less than the system default (which is 86400 seconds), the FreshTime will be set to the default value (86400 seconds). This command is not retroactive. That is, the newly configured refresh value applies only to new incoming files, and it has no effect on the entries already in the cache.

Note Step 4 is not applicable to Cisco VVB.

Stale files are refreshed on an as-needed basis only. A stale cached entry can stay in the cache for a long time until it is removed to make room for either a fresh copy of the same file or another file that needs its memory space in the cache.

A stale cached entry is removed on an as-needed basis when all of the following conditions are true:

- The cached entry becomes stale.
- Its refresh count is zero (0); that is, the cached entry is not being used.
- Its memory space is needed to make room for other entries.

Note When the Age exceeds the FreshTime and the file needs to be played, the HTTP Client checks with the media server to determine whether or not the file has been updated. When the HTTP Client sends a GET request to the server, it uses a conditional GET to minimize its impact on network traffic. The GET request includes an If-Modified-Since in the headers sent to the server. With this header, the server returns a 304 response code (Not Modified) or returns the entire file if the file was updated recently.

This conditional GET applies only to nonstreaming mode. In streaming mode, the HTTP Client always issues an unconditional GET. There is no If-Modified-Since header included in the GET request that results in an unconditional reload for each GET in streaming mode.

You can reload individual files into cache by entering the following command:

Gateway configuration: **test http client get http://10.0.0.130/en-us/sys/1.wav reload**

Note This command is only applicable to IOS VoiceXML Gateway.

Note HTTPS for media files is not supported.

Design Considerations for Large Number of Media Files

In situations where a large number of different media files (.wav) is played to the customers, the gateway cannot cache all the media files because of space constraints in the gateway.

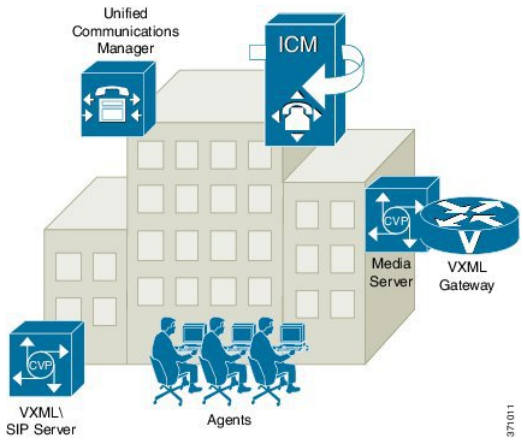
For example, consider an enterprise having a large number of agents. All of the agents have their own customized agent greeting file. It is impossible to cache all the customized agent greeting files in the gateway flash because of space constraints in the gateway.

Collocated Media Server with VoiceXML Gateway

The following section outlines the recommended solution when a Media Server and VoiceXML Gateway coexist in a LAN environment, if the bandwidth is abundant over the LAN, the prompt download should not add noticeable delay.

The following figure shows the collocated deployment for Media Server and VoiceXML Gateway.

Figure 1: Collocated Deployment for Media Server with VoiceXML Gateway

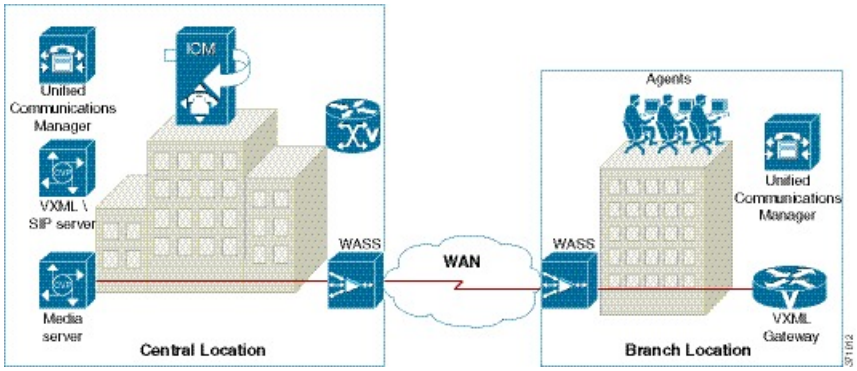


Distributed Media Server and VoiceXML Gateway Separated by a High Latency Link

This section outlines the recommended solution when a Media Server and VoiceXML Gateway are separated by a WAN.

The following figure shows the distributed deployment over WAN.

Figure 2: Distributed Deployment Over WAN



In this situation downloading the media files from Media Server across a high latency WAN to the VoiceXML Gateway can add noticeable delays to the caller. This delay will greatly impact the user experience. The delay will be proportional to the size and number of media files transported across the WAN. This delay can be optimized using Cisco Wide Area Application Services (WAAS).

Considerations for Streaming

Consider the following factors for both the LAN deployment and the WAN accelerator deployment:

- Maximum network round-trip time (RTT) delays of 200 milliseconds.

  For example, during the transfer of files from the CVP Operations Console to the Ingress or the VXML Gateway using Bulk Administration File Transfer (BAFT).

- Maximum number of streaming sessions supported per gateway with no additional overhead of video with media forking.

  For example, the maximum number of calls supported per gateway in streaming is 275. This number is valid only for the Cisco 3945E Integrated Services Router (ISR) with no additional overhead at a maximum network RTT delay of 200 milliseconds.

The following table describes the media file deployment scenarios over LAN and WAN:

Scenario	Frequency of Change	Over LAN	Over WAN
Small number of files	Rare	Cached	Cached
Small number of files	Often	Streamed or Cached	Streaming with WAAS
Large number of files	Rare	Streamed	Streaming with WAAS
Large number of files	Often	Streamed	Streaming with WAAS

Note Cisco VVB does not support Media Streaming feature.

Media Server Association with Call Server and VXML Server

Note Unified CVP Call Server, Media Server, and Unified CVP VXML Server are co-resident on the same server.

If your Unified CVP Call Server, Media Server, and Unified CVP VXML Server reside on the same hardware server and you have multiple co-resident servers, Unified CVP does not automatically use the same physical server for call control, VXML, and media file services. If the components are co-resident, no component is forced to use the other co-resident components, and Unified CVP might possibly use the components located on another server.

By default, the components are load balanced across all of the physical servers and do not attempt to use the same server for all of the services. During thousands of calls, all of the components on all of the servers are load balanced and equally utilized, but one specific call could be using several different physical servers. For example, for one particular call you can be using SIP call control on one server, VoiceXML on another server, and the media files on another server.

You can simplify management and troubleshooting by configuring Unified CVP to use the same physical server for all of these functions on a per-call basis. If there is only one server in the system, then simplification is not a concern. The instructions in the following procedures show you how to configure Unified CVP so that it uses components on the same physical server instead of load balancing and using a random server for each component.

Choose Coresident Unified CVP VXML Server in ICM Script Editor

Procedure

Step 1 Set up the **media_server** ECC variable that specifies your Unified CVP VXML Server in the ICM script by using use the Formula Editor to set the **media_server** ECC variable to **concatenate("http://",Call.RoutingClient,":7000/CVP")**.

Call.RoutingClient is the built-in call variable that ICM sets automatically for you. The routing client name in ICM is usually not the same as the Unified CVP Server's hostname.

Step 2 Apply the routing client name as a hostname in the VXML gateway. Do not use noncompliant characters such as an underscore as part of the hostname because the router cannot translate the hostname to an IP address if it contains noncomplaint characters. Use the **ip hostname strict** command in the router to prevent the use of invalid characters in the hostname. This action ensures that the hostname is acceptable to Unified CVP.

Step 3 Configure the routing client hostname for every Unified CVP Server Routing Client.

Choose Coresident Media Server in Call Studio

Procedure

Step 1 In the ICM script, set one of the **ToExtVXML[]** array variables with the call.routingclient data, such as ServerName=call.routingclient. This variable is passed to the Unified CVP VXML Server, and the variable is stored in the session data with the variable name ServerName.

Step 2 In Cisco Unified Call Studio, use a substitution to populate the Default Audio Path. Add the Application_Modifier element found in the Context folder, and specify the Default Audio Path in the Settings tab in the following format: **http://***{Data.Session.ServerName}*

Choose Coresident VXML Server Using Micro-Apps

If you are using Micro-Apps in conjunction with the Unified CVP VXML Server, pay careful attention to the **media_server** ECC variable in the ICM script because the same variable is used to specify both the Unified CVP VXML Server and the media server, but the contents of the variable use a different format depending on which server you want to specify. Use the **media_server** ECC variable as indicated in this procedure whenever you want to use a Micro-App for prompting. If you subsequently want to use the Unified CVP VXML Server, rewrite this variable by following the previous procedure.

Procedure

Step 1 Set up the **media_server** ECC variable that specifies your Media server in the ICM script by using the Formula Editor to set the **media_server** ECC variable to **concatenate("http://",Call.RoutingClient)**

Call.RoutingClient is the built-in call variable that ICM sets automatically for you. The routing client name in ICM usually is not the same as the Unified CVP Server hostname.

Step 2 Use the name of the routing client as a hostname in the VoiceXML Gateway.

Do not use noncompliant characters such as an underscore as part of the hostname because the router cannot translate the hostname to an IP address if it contains any noncomplaint characters. Use the **ip hostname strict** command in the router to prevent the use of invalid characters in the hostname and to ensure that the hostname is acceptable to Unified CVP.

Step 3 Configure the routing client hostname for every Unified CVP Server Routing Client.