GVP Documentation Supplement

Setting Up Voiceprint Carrier Message Detection

5/10/2025

Setting Up Voiceprint Carrier Message Detection

With Voiceprint Carrier Message Detection, you can configure Genesys Media Server to recognize prerecorded messages that carriers in a region may provide, and tailor your outbound response accordingly.

Contents

How It Works

During the pre-connect period of a call, the Genesys Media Server could receive one of many highly specific carrier messages. These messages vary with each situation and country and region, but they are definable and you can build a database of them. Using that database, your system can match the message that it hears during preconnect with a known recording—thus identifying a known situation—and then use that information to take the appropriate action.

If the remote party is an answering machine (AM), Media Server will know that and wait for the appropriate AM signal before delivering a message. If the remote party is an out-of-service number and Media Server finds a matching recording, you can configure it to take a different action.

Steps to Configure

You can use Voiceprint configuration options to define the database of sounds that MCP matches against incoming pre-connect messages, in real time.

1. Acquire or develop a list of pre-connect carrier messages that your system can hear.

    > **Tip**
    >
    > To compile your list of carrier messages, Genesys suggests seeking help from your Partner or Genesys Professional Services, or by recording the various carrier messages in that country or region.

2. Open Genesys Administrator (GA) or Genesys Administrator Extension (GAX) and navigate to your Media Control Platform (MCP) application, and within it, the `mpc` section.

3. Enable Voiceprint with this exact configuration:

    *   `cpa.enable_carrier_messages = true`

4. Assign a name to each carrier message with configurations like these examples:

    *   `cpa.carriermsg.0 = file://c:\media\carriermsgs\IsBusy.wav`

    *   `cpa.carriermsg.1 = http://someserver/media/carriermsgs/IsBusy2.mp3`

    *   `cpa.carriermsg.2 = http://someserver/media/carriermsgs/IsBusy3.wav`

    *   `cpa.carriermsg.3 = http://someserver/media/carriermsgs/NotInUse.wav`

    *   `cpa.carriermsg.4 = http://someserver/media/carriermsgs/CannotbeReached.wav`

    *   `cpa.carriermsg.5 = http://someserver/media/carriermsgs/Operator_NotInUse.wav`

        You can define up to 100 carrier messages (from `carriermsg.0` to `carriermsg.99`). We recommend defining more than 10 messages for best results. Gaps or empty locations in your configuration are legal, as in this example with nothing to the right of the equals sign:
        `cpa.carriermsg.0 =`

> ### Tip
>
> If your configuration includes multiple MCPs, consider the advantages of standardization for filenames and directory paths. You can achieve this by defining the audio file database and setting the preconnect lists on a single MCP, testing to be sure that's what you want, and finally exporting the related options from that MCP object. From there, you simply import your designed and tested configuration into the other Media Servers and MCPs that perform call progress detection.

5. Add your result lists, using configurations that match specific results with specific carrier messages. Consider these examples:

   - `[mpc]cpa.preconnectresult.busy.list = 0 1 2`
     BUSY is the response to cpa.carriermsg.0 through cpa.carriermsg.2

   - `[mpc]cpa.preconnectresult.custom1.list = 3 4 5`
     CUSTOM1 is the response to cpa.carriermsg.3 through cpa.carriermsg.5

   - `[mpc]cpa.postconnectresult.machine.list = 7`
     MACHINE is the response to cpa.carriermsg.7 (See Configuring for Answering Machine Detection below.)

   Success! You have instructed Media Server how to identify up to 100 carrier messages and what action to take in each case.

> ### Important
>
> **One Number, One List:** Every value in every list must be unique. For example, you can place the number 1 (meaning `cpa.carriermsg.1`) in only one list. Duplicates corrupt VoicePrint detection results.

Configuring for Answering Machine Detection

If an answering machine takes the call, Media Server waits until the "leave-a-message" audio plays, then delivers the appropriate audio. Consider the following details:

- You must configure a postconnect carrier message result list, which could look like this:
  `cpa.postconnectresult.machine.list = 0 1 2 4`
  In this configuration, detecting **carriermsg.0**, **1**, **2**, or **4** triggers the result **Machine**.
  The One Number, One List rule applies here, too.

- Carrier message detection still occurs in preconnect mode—it *always* does. But as stated above, the results are delayed.

- MCP must be configured to start in preconnect mode, and after AM detection, move to postconnect mode while running, allowing the carrier message detection results to be returned after the call is connected.

Optional Parameters

You can customize your setup to meet specific needs in your organization or to address troubleshooting issues. See Voiceprint Analysis Configuration Options and the Media Control Platform Configuration Options documentation for all details on these parameters.

[mpc]**cpa.cm_enable_initial_tone_filter** = [true | false] (Default = false)
This parameter enables initial tone filtering, which suppresses the generic single tone at the beginning of the audio frames for carrier message detection. (To recognize the multiple ring tones in some countries, MCP must remove the generic single tone that precedes all of them.)

[mpc]**cpa.cm_initial_silence_suppression_level** = [integer value] (Default = 64)
This parameter specifies the initial level needed for silence suppression in carrier message detection. Increasing the value will result in higher energy noise being suppressed.

[mpc]**cpa.cm_match_percent** = [integer value] (Default = 80) <percent>
This parameter specifies the how strict we are in carrier message matching. A higher value means stricter matching, which may restrict the results. A lower value means more lenient matching, which yields more results, but may reduce accuracy.

[mpc]**cpa.readduration** = [integer value] (Default = 60)
This parameter specifies the maximum number of seconds that MCP reads a carrier message before stopping. This prevents delays when MCP reads carrier messages that are too long.

[mpc]**cpa.nframes_cm_detection** = [integer value] (Default = 50)
This parameter specifies how many frames from the audio files and real-time incoming audio that MCP to uses for detection. A higher number means better accuracy but more time consumed in detection. A lower number means less time consumed but also less accuracy.

Troubleshooting Tips

- **Bad carrier message files:** Ensure that the carrier message audio files are in the correct format for MCP to use them.

- **Duplicate indices:** Duplicate indices will cause matching to not work properly. Ensure that there are no duplicates in result lists. Follow the One Number, One List rule.

- **Accurate Indices:** Ensure that the indices that you are trying to match are listed in the correct result lists.

- **File location:** Ensure that MCP can access the files (particularly HTTP files) and that the paths are correct. ERRORs in the logs will tell you which files have issues.

- **MCP in the preconnect state:** MCP must be in the preconnect state for preconnect carrier message detection to work. An MCP configured in the wrong state may send the incorrect input to the call progress analysis, and cause detection failures.