GENESYS

GVP Deployment Guide

Media Control Platform

5/9/2025

Media Control Platform

The Media Control Platform is the core component of GVP, because it executes the actual voice applications in the solution. In addition, it is used by other communication layer components, such as SIP Server, to provide broader customer service scenarios, such as agent interactions, and many other functions.

This section provides an overview of the following topics:

- Media Control Platform Components
- Media Control Platform Services
- Media Control Platform Functions
- Interpreters

Media Control Platform Components

The Media Control Platform is composed of:

- A core executable file that consists of the Call Manager application programming interface (CMAPI) and the SIP Line Manager.
- The Media Server, which is a group of libraries (and third-party transcoder dynamic-link libraries [DLL]) that run in-process in the Media Control Platform, for media processing and Real-time Transport Protocol (RTP) streaming.

> ### Tip
> The library files in the GVP installation packages for Linux have a .so file extension (not .dll).

- The Next-Generation Interpreter (NGI), which is a DLL that runs in-process in the Media Control Platform. The interpreter DLL is loaded by the CMAPI application.

- The legacy GVP Interpreter (GVPi), which is a DLL that runs in-process in the Media Control Platform on Windows only. The interpreter DLL is loaded by the CMAPI application.

> ### Tip
> For more information about NGI and GVPi, see Interpreters.

- The Media Resource Control Protocol (MRCP) Client, which is a group of libraries that runs in-process in the Media Control Platform, to handle MRCPv1 or MRCPv2 communication with automatic speech

recognition (ASR) and text-to-speech (TTS) speech engines.

- The Fetching Module, previously a separate component, is integrated with the 9.0 Media Control Platform and communicates directly with the NGI.

For information about installing the Media Control Platform, with basic configuration, see Installing GVP.

Media Control Platform Services

VoiceXML

Media Control Platform services are defined by voice applications that are executed when a SIP session is established between the Media Control Platform and the service user. The Media Control Platform can host various application execution environments and use multiple implementations of a particular language. The Media Control Platform is most often used to deploy dialog-based services that are built using VoiceXML.

NETANN

The Media Control Platform supports two other predefined services: announcements and conferencing. In conjunction with an underlying media-processing resource, the Media Control Platform can provide extended versions of all services defined in Network Announcements (NETANN) for example, announcements with pre-recorded audio prompts.

The Media Control Platform also supports a record service that is initiated when an incoming SIP INVITE message contains the record parameter in the Request URI and annc is in the user part of the request.

The Media Control Platform offers services in accordance with the Internet Engineering Task Force (IETF) Request for comments (RFC) 3261 (SIP) and RFC 4240 (NETANN) standards and the Burke Draft (SIP interface to VoiceXML media services). The NETANN interface is accessed through the Resource Manager, but it can be accessed directly in a standalone configuration.

> ### Tip
> NETANN defines a number of extensions to SIP that clients use to request execution of particular classes of applications, including simple announcements, conferences, and dialogs. NETANN is defined in RFC 4240.

MSML

The Media Control Platform supports the conferencing service through Media Server Markup Language (MSML). In conjunction with an underlying media-processing resource, the Media Control Platform can provide extended MSML conferencing features, such as the ability to set the conference role, perform prechecks to ensure the audio or video prompt file is found before the conference begins, and support for relative path URIs to the media file.

The Media Control Platform supports dual-channel Call Recording service through MSML that is initiated when an incoming SIP INVITE message contains the record parameter in the Request URI and the MSML parameter is in the user part of the request, In this case, however, a different type of conference-based recording is indicated. See Dual-Channel Call Recording.

In addition, the Media Control Platform supports a DTMF URL scheme through MSML, which enables the specification of a sequence of DTMF digits to generate, record, and collect DTMF events within a single SIP session.

The Media Control Platform can be deployed without VoiceXML support, as an MSML only server. It implements MSML server functionality through its MSML application module according to the `draft.saleem.msml.txt` standard.

For a list of the supported standards for Media Control Platform services, see Specifications and Standards.

Service Delivery

The Media Control Platform controls overall execution of the voice applications, but the applications rely heavily on access to media-processing resources. One or more underlying, third-party media-processing resources (such as media servers, speech recognition servers, or speech synthesis servers) deliver ASR and TTS services.

The media-processing resources handle RTP packets in three ways:

- By using direct or indirect RTP streams to interact with the service user.
- By processing or interpreting RTP packets received from the service user.
- By generating RTP packets for transmission to the service user.

Interaction with the media-processing resources occurs by various methods that include the RFC 4240 standard and MRCPv1 and MRCPv2.

Media Control Platform Functions

The Media Control Platform performs these functions:

- Initiates outbound calls.
- Handles network-initiated call disconnections.
- Performs application-initiated calls.
- Supports VoiceXML applications.
- Plays audio, video, and TTS prompts.
- Streams TTS, audio, and video
- Records utterance data.
- Records audio and video.
- Supports dual audio channel and dual video channel call recordings.
- Collects call recordings.

- Performs ASR and dual tone multi-frequency (DTMF) input handling (barge-in or non-barge-in).

- Streams audio data to an ASR server for speech recognition.

- Reserves ASR and TTS resources at call initiation.

- Transfers calls.

- Sends active speaker notifications to the conference creator.

- Conference calls that use audio and video, and support an unlimited number of participants.

- Performs transcoding from one media codec to another when required for example, by bridging media sessions.

- Logs data and produces metrics.

- Performs Call Progress Detection (CPD) and analysis.

- Provides dual-stack functionality, where one call leg uses IPv4 communication and the other IPv6.

- Supports Apple's HTTP Live streaming (draft-pantos-http-live-streaming-16) subset of features. Media Server can play media and master playlists of the type `On-Demand`, `Event`, and `Live`.

- Supports HTTPS schema for MSML HLS (HTTP Live Streaming) and the `gvp:precheck` attribute of MSML play.

For more information about how the Media Control Platform performs its functions, see How the Media Control Platform Works.

Interpreters

The Next Generation Interpreter (NGI) and the Legacy GVP Interpreter (GVPi) are Voice Extensible Markup Language (VoiceXML) interpreter components on theMedia Control Platform. The CCXML Interpreter (CCXMLI) is the Call Control Extensible Markup Language (CCXML) interpreter on the Call Control Platform used for executing call-control applications.

VoiceXML Interpreters

The VoiceXML interpreters request VoiceXML pages from a web application server (optionally through a fetching/caching proxy), compile the pages into an internal representation, and execute them to manage a dialog with a user. As part of this dialog management process, the VoiceXML interpreter also requests resources (such as speech recognition and speech synthesis sessions) from other platform components.

The interpreters are responsible for driving the underlying platform to execute the VoiceXML application. The interpreters interpret the VoiceXML applications to determine the interactions that occur with a caller, and the Media Control Platform provides the media services.

The VoiceXML interpreters are Windows dynamic-link libraries (DLL) or Linux Shared Objects that run in-process on the Media Control Platform. In GVP 8.1and above, the GVPi is available only for Windows.

For Windows deployments, the Media Control Platform can run one or both VoiceXML interpreters (NGI and GVPi), and both are installed by default. Voice application, or IVR Profile, provisioning determines which interpreter to use for a particular voice application. You can also specify which interpreter to use as the default VoiceXML interpreter for GVP.

The following subsections briefly describe the GVP 9.0 interpreters, to provide a context for the syntactic and semantic differences between the applications that they support.

NGI

The NGI is the default VoiceXML interpreter for voice applications that are running on GVP 9.0. It was introduced with GVP 8.0 and is built on scalable architecture that leverages multi core and multiprocessor environments.

The NGI parses VoiceXML documents in stricter accordance with the VoiceXML and Speech Synthesis Markup Language (SSML) schemas, with GVP extensions. Any element or attribute that violates the schemas generates a parsing error.

In 8.1.5, a new parser was introduced for XML documents that are retrieved by using the <data> element. Its behavior differs from the previous parser in the following ways:

- Entity declaration elements (<!ENTITY> elements) in the XML document type declaration are not handled and an `error.semantic` is generated when XML documents that contain these elements are retrieved.

- Namespace declaration attributes (`xmlns` attributes) within an element are not exposed as normal attributes in the exposed DOM object.

- If there is no namespace declaration with a local name in the XML document, the prefix property of the Node object in the exposed DOM.

- If the same local name is redefined with another namespace URI, the document does not treat the re-definition correctly.

- The Attr object's child Nodes property is evaluated to null, instead of the value of the attribute.

- The evaluation of the <data> name variable returns the XML document. In the old implementation, it returned the [object VG_DOM_CLASS] string.

> **Tip**
>
> You can revert the NGI back to the old behavior by setting the value of the `data.use_xerces_dom_parser` configuration object in the [vxmli] section of the Media Control Platform Application to true.

For more information about NGI support for the VoiceXML and SSML schemas and GVP extensions, see the GVP Voice XML Help.

The NGI supports the CTI interface through SIP INVITE and REFER messages, and in GVP 8.1 and above, supports Linux as well as Windows.

GVPi

The GVPi, which was new in GVP 8.1, is the legacy GVP 7.6.x VoiceXML interpreter that was present in the IP Communication Server (IPCS). It enables GVP to support VoiceXML 2.1 applications implemented in GVP 7.6. GVPi also supports interactions with IVR Server through CTI Connector. The

Media Control Platform and CTI Connector communicate by using SIP.

In addition to VoiceXML 2.0 and 2.1 applications, the GVPi can process XML applications that use Telera XML (TXML) extensions for call-control functionality. TXML call-control functions include creating outbound-call legs or bridging calls without using the <transfer> tag, queuing calls, and managing the call legs.

Within GVPi, the Page Collector module uses HTTP and HTTPS to retrieve VoiceXML documents, scripts, grammars, and media content, similar to what the Fetching Module does for the NGI. The functionality that was provided by the Call Flow Assistant (CFA) in earlier releases of GVP is now divided between GVPi and the CTI Connector.

CTI Interface

GVPi (on the Media Control Platform) interacts with the CTI Connector through the SIP protocol by using SIP INFO messages. All of the CTI features available in GVP 7.6 are supported in this release for example, treatments, transfers, user data, and interaction data.

> ## Tip
> GVPi is supported on Windows operating systems only.

For more information about GVPi support for VoiceXML and for TXML call control, see the GVP 8.1 Legacy Genesys VoiceXML 2.1 Reference Manual.