GVP Deployment Guide

How the Media Control Platform Works

4/8/2026

How the Media Control Platform Works

Read here about how the Media Control Platform performs its role in a GVP deployment:

- Operational Overview
- Media Services
- Speech Services
- Transfers
- Conferencing
- Debugging VoiceXML Applications
- HTTP Basic Authentication
- Masking sensitive data in MCP trace logs

In principle, the Media Control Platform works with the Resource Manager and other GVP components to process calls in a similar way whether the VoiceXML interpreter is the NGI or the GVPi. However, the way in which the NGI and the GVPi perform certain functions is different and there are differences in some areas of feature support. For simplicity, and for the purpose of providing an overview of the way GVP works, this section describes Media Control Platform functioning with the NGI only.

For more information about the differences between GVPi and NGI support for GVP features, see the Genesys Voice Platform 8.1 Application Migration Guide.

Operational Overview

The Media Control Platform receives requests for call and media services from the Resource Manager in the form of SIP INVITE messages. The platform can conference, transfer, or redirect calls by using other kinds of SIP messages (see Transfers). The platform can also initiate outbound calls by sending SIP INVITE requests through the Resource Manager or directly to the destination.

The platform provides media, conferencing, and other bridging services for both Media Control Platform and Call Control Platform calls. Media Control Platform services are defined by VoiceXML applications that are executed as part of the process of establishing a SIP session between the platform and the service user. In addition, the platform supports NETANN and MSML conferencing and prompt announcement services. SIP Server uses NETANN and MSML services to carry out media operations.

Key and Certificate Authentication

The platform also supports the configuration of a password for key and certificate authority to perform server authentication, by using the attributes of the `sip.transport.<n>` configuration option. When acting as a server, the Media Control Platform supports mutual authentication for clients.

Network Traffic Partitioning

The Media Control Platform supports partitioning of network traffic across various network interfaces, including SIP, HTTP, MRCP, RTSP, and RTP. For a complete list of configuration options that are used for specific types of network traffic, see Appendix I in the Genesys Voice Platform 8.5 User's Guide.

Inbound Calls

The Media Control Platform handles inbound service requests for call or media services, as follows:

1. The Media Control Platform, acting as a SIP User Agent Server (UAS), receives a SIP INVITE from the Resource Manager. Because the Resource Manager modifies the SIP request by inserting service prerequisites for the 1. IVR Profile, the SIP Request-URI includes a voicexml parameter that specifies the URL of the initial page of the required VoiceXML application. Alternatively, the Media Control Platform can be configured (in the sip.vxmlinvite configuration option) to accept calls in which the originator specifies the initial VoiceXML URL in the Request-URI of the SIP INVITE. In these cases, provided that the syntax and format of the Request-URI are correct, the normal Resource Manager method of mapping calls to IVR Profiles will be bypassed.
   - The Media Control Platform recognizes the following Request-URI formats:
     - `sip:dialog.vxml.<URL>@host.com`, where the URL portion must be properly encoded (`draft-rosenberg-sip-vxml` format)
     - `sip:<user>@host.com;voicexml=<URL>` (NETANN dialog service format)
     - `sip:conf=<ID>@host.com` (NETANN format, for calls to join a specified conference without going through a VoiceXML application)
     - `sip:msml[=conf_id]@ms.example.com[;uri-parameters]`, where the request is for MSML service when a conference call is created by MSML. The join request is transmitted to the Media Control Platform in a separate SIP INFO message. The conference ID in the SIP Request URI indicates to the Resource Manager that the callers for this conference must be routed to the same Media Control Platform.
   - The Media Control Platform supports the following service parameters in the Request-URI:
     - **voicexml** The value must conform to the URI syntax that is defined in RFC 3986.
     - **maxage** The value must be all digits.
     - **maxstale** The value must be all digits.
     - **method** The value must be either get or post.
     - **postbody** The HTTP body for POST requests.
     - **timeout** The value must be numeric.
     - **gvp.alternatevoicexml** Specifies an alternative VoiceXML page if the VoiceXML interpreter fails to fetch the primary page.
     - **gvp.config.<parameter name>** Sets the values of certain platform configuration options for the duration of the media session. This mechanism enables an IVR Profile to override certain Media Control Platform configuration parameters, for the session that is being executed in the context of this VoiceXML application. For the configuration parameters whose values can be set dynamically, see the Media Control Platform reference information appendix in the Genesys Voice Platform 8.5 User's Guide.

Special characters in the Request-URI parameters from the SIP interface must be URL-encoded (escaped). These include ? (%3F), = (%3D), and ; (%3B).

- The Resource Manager passes the value of the following IVR Profile `gvp.policy` parameters in the Request-URI, for handling by the Media Control Platform:
  - `mcp-asr-usage-mode`
  - `mcp-max-log-level`
  - `mcp-sendrecv-enabled`

  For more information about these policy parameters, see the chapter about provisioning IVR Profiles in the Genesys Voice Platform 8.5 User's Guide.
- Media services are required, so the Resource Manager includes the SIP User Agent (UA) Session Description Protocol (SDP) offer in the SIP INVITE. For more information about how the Media Control Platform negotiates media services, see Step 6.

2. For valid INVITE requests, the platform immediately responds to the Resource Manager with a 100 TRYING message. In addition, configurable options enable you to specify whether the platform will also send intermediate provisional responses while the call is being set up. Provisional responses can include custom SIP headers, which must have the prefix, `X-`.

> **Tip**
> The Media Control Platform does not support sending early media after a provisional response has been sent.

   For the responses that the Media Control Platform sends if an error occurs during call setup, see the appendix about SIP response codes in the Genesys Voice Platform 8.5 User's Guide.

3. The Media Control Platform passes all the generic SIP Request-URI parameters to the VoiceXML interpreter.
4. The VoiceXML interpreter sends an HTTP/HTTPS or file retrieval request to the Fetching Module to fetch the initial page. The request includes the timeout, maxage, and maxstale values, if present, to determine whether the fetch can be satisfied from the cache store. For more information about how caching is used to improve Media Control Platform performance, see Caching.
5. The VoiceXML interpreter compiles and interprets the initial page, and all subsequent pages, so that the Media Control Platform can execute the application. The VoiceXML application is ready to proceed when the VoiceXML document is fetched, parsed, and compiled. The NGI supports the following encodings for VoiceXML pages and external ECMAScripts objects:
   - UTF-8
   - UTF-16
   - ISO-8859-x
   - Far-East encoding for Japanese, Chinese, and Korean The VoiceXML interpreter retrieves the encoding information for a document from the encoding attribute of the XML header, or from the charset attribute of the <script> tag.
6. At the same time that it passes SIP INVITE information to the VoiceXML interpreter (see Step 3), the Media Control Platform passes the SDP to the Media Server, so that it can negotiate media capabilities. For information about capability negotiation, and the codecs that the Media Control Platform supports, see Codec Negotiation on page 129. For information about the file formats that are supported for playing and recording audio and video for various codecs, see the Media Control Platform reference information appendix in the Genesys Voice Platform 8.5 User's Guide.

7. When the VoiceXML application is ready to be executed, the Media Control Platform sends a 200 OK response to the initial INVITE request. The response includes the Media Server SDP answer, if applicable. If the initial INVITE and the ACK that is returned do not contain the required SDP information, a media-less dialog is established. In general, the VoiceXML application starts when the 200 OK response is acknowledged (that is, when the platform receives an ACK). However, it is the VoiceXML application itself that determines whether it is ready to start. In particular, if a media-less dialog has been established, the VoiceXML application will not start until the platform receives a re-INVITE that includes the SDP information for the caller.
8. When the VoiceXML application starts, it controls the session. The VoiceXML interpreter is responsible for driving the Media Control Platform to execute the VoiceXML application appropriately. The NGI performs speech and DTMF recognition, and it issues commands to the platform to execute call and media operations.
    - The platform sends and receives SIP INFO messages for the following application events:
        - To make a request The application can specify the content type and content body of the SIP INFO message.
        - To send or receive data The application can send data in custom SIP headers. The platform can send information that it receives in SIP INFO headers to the NGI, and this information is provided to the application in shadow variables. Note that when the SIP INFO content type is application/dtmf-relay, it is treated as DTMF input instead of an application event. The content format is `Signal = <digit>`.
        - The application uses dialogs to initiate transfers as required. For more information about how the Media Control Platform performs transfers, see Transfers. The NGI supports use of the userdata attribute in the <transfer> tag, to abstract CTI data. The NGI exposes CTI userdata to the application in a session variable, session.com.genesyslab.userdata.
        - The platform provides media services through the Media Server, for operations such as playing prompts and recording audio and video. For more information, see Media Services .
        - For ASR or TTS, the Media Control Platform controls speech resources through the MRCP Client. For more information, see Speech Services.
9. The VoiceXML application can invoke other VoiceXML applications. The VoiceXML interpreter is responsible for issuing commands to the Fetching Module to fetch VoiceXML pages and other applications.
10. When a caller disconnects (that is, when a BYE is received), the platform notifies the Voice XML application (through the connection.disconnect.hangup event). If the BYE includes a Reason header, the value of the Reason header is passed verbatim to the application. If the application disconnects, the platform generates a BYE request.
11. For each VoiceXML session, the Media Control Platform generates call-detail records, which it sends to the Reporting Server. For more information, see CDR Reporting.
12. For each VoiceXML session, the Media Control Platform sends logs and metrics (VoiceXML application event logs) to the log sinks and, from here, to the Reporting Server. For more information about metrics, see Metrics. For descriptions of the Media Call Control Platform metrics, see the Genesys Voice Platform 8.5 Metrics Reference.

Media Services

The Media Server provides the following services:

- Prompt playback
- Recording
- DTMF digit detection and handling
- ASR streaming (streaming TTS audio to the SIP call, and streaming audio data to an ASR server to perform speech recognition)
- Audio encoding and transcoding
- Audio and video streaming

The media channel is established directly between the Media Server and the remote party (through a media gateway, if required), over RTP. The Media Server also supports Secure RTP (SRTP).

Selected Features

The following are some of the advanced features that the Media Server provides for audio and video services:

- Support for audio, video, and mixed audio-video for calls and conferences
- Support for an unlimited number of participants in conferences
- VCR controls that enable the caller to navigate within an audio or video stream by using DTMF keys (for example, play, pause, stop, resume, and skip forward or backward)
- Full call recording for audio and video, including configurable support for recording DTMF input
- Call recording support for third-party server
- Fine-grained control of conference input and output through configurable parameters for gain control, audio mixing, video switching, and so on
- Mechanisms to guarantee the required level of real-time performance for time-critical functions (for example, generating output content in advance and buffering it)
- Per-prompt control of DTMF barge-in
- Support for Call Progress Analysis (CPA)

> **Tip**
> CPA can be performed by an external media gateway, such as AudioCodes, by the Dialogic card, or by the Media Control Platform itself.

- Flexible packet size and SDP configurable ptime parameters

- Terms of Service (ToS) tagging for RTP packets
- Mechanisms to specify maximum record size
- Wave and AVI container support for additional codecs (see Codec Negotiation)
- Support for the DTMF send method based on the SDP origin field
- DTMF distribution in the conference
- Audio and video sources can play from separate URLs in parallel
- Initial bursting to fill the Dialogic playback buffers quickly

Dual-Channel Call Recording

The Media Control Platform performs many types of recording functions (see Media Control Platform Functions on page 42), including advanced MSML server functions, such as, dual-channel call recording.

Dual RTP Streams

The Media Control Platform s Media Server module can replicate the RTP streams of two inbound calls in a Call Recording session (indicated by the Request-URI) to a third-party recorder. SIP Server initiates this request by using MSML.

The SDP and other connection-specific parameters are passed in the an attribute that is used to start additional recordings, pause, stop, and restart streaming.

HA for Clients

The Media Control Platform adheres to the RFC 3263 standard, in which SIP uses DNS procedures to enable a client to resolve a SIP URI to an IP address, port, and transport protocol. SIP also uses DNS to enable a server to send a response to a backup client if the primary fails.

For a complete list of GVP-supported specification and standards, see Specifications and Standards.

Call-Progress Detection and Analysis

The Media Control Platform stores the most recent call-progress detection (CPD) events. When the call is connected, the last collected CPD event is sent to the application module, which determines how to handle the results.

The default CPD parameters are defined in the Media Control Platform's configuration. These tuning parameters can be overwritten by the IVR Profile's service parameters, that are passed to Media Control Platform by the Resource Manager.

The Media Control Platform reports CPD results to VoiceXML application as they are detected. If the Media Control Platform does not perform Call Progress Analysis (CPA), the CPD result is provided to VoiceXML applications and processed by the Media Control Platform through Media Server Markup

Language (MSML) requests. Call Progress Analysis (CPA) is initiated by a CPA request in the MSML dialog or by the VoiceXML application.

Detection Methods

The Media Control Platform supports two methods of CPD:

- Gateway-based with audiocodes, where SIP Server is the CPD provider.
- Core-based, where the Media Server module of the Media Control Platform is the CPD provider. CPD is triggered by the MSML <cpd> element.

When core-based CPD is implemented, the CPD result is passed to a dialog that is initiating a VoiceXML application, by using a MSML <dialogstart> request, or a <send> request with an event=start parameter. The CPD result is contained in the MSML <gvp:params> element.

The <gvp:params> request does not validate the content of the name and value parameter pair. The CPD result string that is sent to the VoiceXML dialog in the <gvp:params> element is mapped to the CPD event, such as value=cpd.sit.nocircuit.

For a complete list of CPD events, see "Appendix B: MSML Specification" in the Genesys Media Server 8.5 Deployment Guide.

Analysis Logging

Result analysis and logging can performed when the `cpa.enable_log_param` and `cpa.enable_log_result` configuration options are enabled in the mpc section of the Media Control Platform Application. This information is logged in the Media Control Platform metrics log.

CPA parameter logging and CPA tone-setting logging are enabled if the cpa.enable_log_param option is configured as true, and the logging timestamp is determined when detection (CPD) is started.

CPA parameter logging and CPA result logging are enabled by using the `cpa.enable_log_param` and `cpa.enable_log_result` configuration options.

The `cpa_parameter` logging and cpa_tone_setting logging configuration options are enabled if the `cpa.enable_log_param` option is configured as true, and the logging timestamp determines when CPA detection is started.

The cpa_result logging option is enabled if the `cpa.enable_log_result` option is configured as true, and the logging timestamp is determined by when Media Control Platform reports the CPD result.

The metrics are contained within the `cpa.enable_log_param` and `cpa.enable_log_result` log messages, which contain the following information:

- `cpa.enable_log_param`
  - Global Call ID
  - Start time of detection
  - Configuration information that was used in detection.
- `cpa.enable_log_result`

- Global Call ID
- Time that the detected event was reported.
- Detected CPD results

The tenant ID and the IVR Profile name is also included in the metrics log in the call_reference entry in the following format:

```
call_reference <SIP Call-ID>|< GVP-SESSION-ID|< GVP-Tenant-ID>| IVR Profile Name
```

Log Format

The CPA log format differs slightly for parameter, tone setting and results logging, as described below:

Parameter Logging Enabled when the `cpa.enable_log_param` option value is set to true, and detection is initiated by an MSML <cpd> request, or by a VoiceXML application, in the following log format:

```
[<field name="">=<value>[|<field name="">=<value>...]]
```

where: `name` is the name of the tuning parameter and `value` is the value of the parameter. For example:

```
max_preconnect_time=30000|max_postconnect_time=20000|max_beep_det_time=30000|
no_limit_timeout=30000|chunks_not_flush_on_state_chg=90000|machine_greet_dur=1800|
voice_pause_dur=1000|max_voice_signal_dur=800|fax_duration=160|voice_range_db=25|
voice_level_db=17.5|max_ring_cnt=9|sil_before_beep=4500|preconnect_tone_det_mode=0|
notime_ringback_match_percent=50|ontime_preconnect_match_percent=60
```

Tone Setting Logging Enabled when the cpa.enable_log_param option value is set to true, and detection is initiated by an MSML <cpd> request, or by a VoiceXML application, in the following log format:

```
[<field name="tone_name"/>=<"tone_value"/>[|segment=<"seg_value"/>[,<field name="name"/>=<"value"/>...] ]]
```

where: `tone_name` is the tone name, `tone_value` is the tone description, `seg_value` is the segment value, `name` is the name of the parameter, and `value` is the value of the parameter. See the following two examples:

Example 1:

```
cpa_tone_setting busy=na_busy
```

Example 2:

```
cpa_tone_setting ringbak=tone1|segment=1,f1min=0,f1max=0,f2min=0,f2max=0,ontimemin=20,ontimemax=20,
offtimemin=0,offtimemax=0|segment=2,f1min=0,f1max=0,f2min=0,f2max=0,ontimemin=20,ontimemax=20,offtimemin=0,
offtimemax=0|segment=3,f1min=0,f1max=0,f2min=0,f2max=0,ontimemin=20,ontimemax=20,offtimemin=0,offtimemax=0
```

Result Logging Enabled when the cpa.enable_log_result option value is set to true, and the CPA result is detected by the Media Control Platform, in the following log format:

```
<field name="value"/>
```

where `value` is one of the following CPA results:

- Human
- Answering Machine
- No Media
- Answering Machine Beep
- Answering Beep Long Silence
- No Beep Long Answering Machine
- Fax
- No Answer Max Ring
- No Answer Timeout
- SIT Vacant Circuit
- SIT Operator Intercept
- SIT Recorder
- Custom1
- Custom2
- Custom3
- Custom4
- No Answer Buffer Limit
- No Media Buffer Limit

• Busy • Fast Busy • SIT No Circuit	• Timeout • Stopped • Unknown

For example: `cpa_result Answering machine detected`

Overwriting CPA Configuration Options

The `gvp.service` options in the IVR Profile can overwrite the CPA configuration options. The IVR Profile service parameter must be prefixed by `voicexml.gvp.config` for VoiceXML services, and by `msml.gvp.config` for MSML services.

For example, to overwrite CPA the `mpc.cpa.maxbeepdettime` option in the IVR Profile for VoiceXML service, add the following name/value pair to `gvp.service-parameters`:

Name: `gvp.service-parameters.voicexml.gvp.config.mpc.cpa.maxbeepdettime`

Value: `fixed,30000`

For a complete list of CPA configuration options that can be overwritten by the gvp.service-parameter in the IVR Profile, see Appendix B in the Genesys Voice Platform 8.5 User's Guide.

Codec Negotiation

The Media Control Platform supports the standard RFC 3264 offer/answer mechanism to negotiate capabilities for media services: The caller includes an SDP offer in the SIP INVITE, the receiving party answers with matched SDP capabilities in the `200 OK`, and the originating caller acknowledges and confirms the negotiated SDP in an ACK message.

The Media Control Platform also supports receiving SIP INVITE messages without SDP. In these cases, it generates an SDP offer in the `200 OK` response. For outbound calls, it also supports receiving SDP in the 183 Session Progress response.

In addition, the platform supports in-call media information updates through a `re-INVITE/200 OK/ACK` sequence.

> **Tip**
>
> If a SIP re-INVITE is sent to the Media Control Platform to alter the SDP while audio is playing, it can cause the loss of some audio when the Media Control Platform flushes its buffer.

The Media Control Platform can support the following codecs:

pcm**	g726	amr	h264
pcma	g729	amr-wb	vp8
g722	g729[b]	h263	telephone-event
g722.2	g729a[b]	h263-1998	gsm

A configurable parameter (`mpc.codec`) enables you to customize the list of codecs that are advertised in SDP offers, or that are used to match the remote party s offer.

If both the Media Control Platform and the remote end point are configured to negotiate multiple codecs for a call session, multiple audio codecs can be used within a single SIP call.

> **Tip**
>
> In certain media operations, such as Conference, CPD/CPA require internal transcoding to a native primitive codec L16. Therefore, the codec being used must be added to the `mpc.transcoders` configuration option in the Media Control Platform Application. If the default IVR Profile has specific transcoding disabled, operations that require that it will fail.

For information about the supported audio and video file formats, see the Media Control Platform reference information appendix in the Genesys Voice Platform 8.5 User's Guide.

SDP Negotiation for Telephony Events

The Media Server module supports telephony events (and the PSTN Connector) by sending DTMF digits by one of three methods when telephone events are negotiated by SDP:

- Telephony tones or signaling (RFC 4733)
- Inband signaling
- SIP INFO messages

To determine which method to use, the Media Server checks the `mpc.sdp.map.origin.[n].dtmftype` parameter that is mapped to the remote SDP s origin field (o=) with the `mpc.sdp.map.origin.[n]` parameter. Therefore, the method is determined by applying the following logic:

- When the Media Control Platform is sending DTMF digits:
  - If the o= and s= attributes in the request from the caller matches the `mpc.sdp.map.origin.[n]` mapping, the DTMF method is dictated by the `mpc.sdp.map.origin.[n].dtmftype` (where `[n]` can be a number from `0` to 9).
  - If a telephony event is negotiated, the RFC 4733 method is used.
  - If neither of the first two scenarios exists, the method that is used is based on the `mpc.rtp.dtmf.send` configuration parameter.
- When the Media Control Platform is receiving DTMF digits:
  - If a telephony event is negotiated, the RFC 4733 method is used together with SIP INFO (if it is listed in the `mpc.rtp.dtmf.receive` configuration parameter).
  - If the first scenario does not exist, DTMF digits are allowed. based on the `mpc.rtp.dtmf.receive` configuration parameter.

If the SDP negotiation results in an media-less SIP dialog (RFC 5552), or the remote SDP has an IP address like `0.0.0.0`, the VoiceXML application does not execute. To run the VoiceXML application, the UAC must initiate SDP negotiation with a RE-INVITE that results in an active media channel and negotiation of a valid remote IP.

MSML-Based Media Services

When enabled for Media Server Markup Language (MSML), SIP Server responds to a media service request by sending an INVITE message first, to establish a connection with the media server, then an INFO message to start the particular service, such as treatment or conference.

When sending an INVITE for MSML service to GVP or Genesys Media Server, SIP Server includes the following special parameters in the Request URI to help identify the kind of MSML service being asked for, as well as for which tenant:

- **media-service** Includes the value treatment, media, cpd, conference, or record, depending on the requested service.
- **tenant-dbid** Includes the identification number for the tenant to which the SIP Server switch

belongs.

The table below describes which service types are covered by the media-service values.

Table: Media-service Values and MSML Service Types

media-service value	MSML service type
treatment	Used for any of the following treatment types, as asked for in the initiating RequestApplyTreamtent: • Music • Silence • CollectDigits • PlayAnnouncement • PlayAnnouncementAndDigits • PlayApplication (including VoiceXML) • RecordUserAnnouncement
cpd	Used for Call Progress Detection (CPD) for outbound calls.
record	Used for both static (DN-level configured) and dynamic (T-Library initiated) call recording.
conference	Used for conferences or for supervisor monitoring.
media	Used for the following services: • greetings (static or dynamic) • music-on-hold, music-in-queue • ringback or busy tone • nailed-up connections • third party in push video-scenarios • RingBack, Busy, FastBusy, as initiated by RequestApplyTreatment

Saving call detail records (CDRs) for these media services is optional, and controlled by the option `media-service-cdrs.reduce`.

Option: `media-service-cdrs.reduce`
Section: cdr Default: `true`
Valid values: `true`, `false`
Takes effect at: start or restart

Disables/enables the storage to the remote database of Resource Manager and Media Control Platform CDRs that have these media service types: media, cpd, record or conference.

- `true` disables CDR storage to the remote database.
- `false` enables CDR storage to the remote database.

Speech Services

The Media Control Platform manages MRCP Client sessions with third-party speech engines. The Media Control Platform provides speech recognition and speech synthesis commands to the MRCP Client, and the MRCP Client communicates these to the MRCP server(s) to carry out speech requests.

- For MRCPv1, the MRCP Client uses Real Time Streaming Protocol (RTSP) to establish MRCPv1 control sessions.
- For MRCPv2, the MRCP Client uses SIP and SDP to create the client/server dialog and set up the media channels to the server. It also uses SIP and SDP, over Transport Control Protocol (TCP) or Transport Layer Security (TLS), to establish MRCPv2 control sessions between the client and the server, for each media processing resource that is required for that dialog.

The platform sends the RTP stream directly to the MRCP server for ASR, and it receives the RTP stream directly from the MRCP server for TTS.

Grammars

The Media Control Platform generates the following grammars:

- Hotkey grammars Grammars that are used to match the UNIVERSALS properties for the hotwords Help, Cancel, and Exit. There are separate grammar files for each supported speech engine. The hotkey grammars are stored in the `C:\Program Files\Common Files\GCTI\www\gvp\mcp\<app_obj_name>\grammar` directory.

> **Tip**
>
> The default hotkey grammars may not contain the correct strings for the hotwords in certain languages. Verify that the grammars are correct for the languages that are required in your deployment, and correct or add any required strings as necessary.

- Built-in grammars The set of built-in grammars provided in the VoiceXML specification. The Media Control Platform provides these because some engines do not support VoiceXML built-in grammars internally on the engine side. The built-in grammars are stored in the `/var/www/gvp/mcp/<app_obj_name>/grammar` directory.
- Inline and Implied grammars Men**and option grammars that the Medial Control Platform generates dynamically. These grammars are temporarily stored in the `<MCP Installation Path>\tmp` directory.

In addition, the Media Control Platform supports native DTMF grammar handling with a built-in DTMF recognizer.

For the default languages and built-in grammars that are supported when strict grammar mode is enabled, see the description of the conformance.supported_* options in the vxmli configuration section.

Microsoft Internet Information Services (IIS) on the Media Control Platform host serves the grammars to the off-board ASR server. The Apache HTTP Server provides the same service when the Media Control Platform is installed on Linux. Ensure that you configure the IIS and Apache application servers to serve the required grammars.

Transfers

VoiceXML or CCXML applications use the <transfer> tag in VoiceXML dialogs to initiate transfers.

Transfer Types

From the perspective of the VoiceXML or CCXML application, there are three types of call transfers:

- Blind The application is detached from the incoming call (and the outbound call, if one is involved) as soon as the transfer is successfully initiated. This means that the application is unable to detect the result of the transfer request.
- Consultation (also referred to as supervised) The application is detached from the incoming call when the transfer process is successfully completed. If the transfer process is unsuccessful, the application retains a relationship with the call. In this way, the application is able to report transfer failures.
- Bridge The application is not detached from the incoming call. When the transfer ends, control of the call always returns to the application, regardless of the transfer result.

Whisper Transfer

In addition, the whisper transfer feature enables the platform to delay connection of the caller and called party after the transfer operation has been performed.

Whisper transfer enables the platform to continue performing media operations with the called party, and to transfer the call out later. Whisper transfer enables the VoiceXML application developer to write an application that first consults with the called party, to determine whether the called party will accept the transferred call. If the called party accepts the call, the transfer proceeds. If the called party rejects the call, the called party is disconnected, and the VoiceXML application can return control to the original caller.

AT&T Transfer Types

AT&T Transfer Connect allows the platform to transfer the call to an agent by using DTMF tones. GVPi and the NGI interact with the network by issuing DTMF tones, and the network provides call-state and call-progress updates through DTMF tones. Three transfer types are supported:

- Courtesy Transfer This transfer is equivalent to a Blind transfer. The VoiceXML interpreter is disconnected as soon as the network receives the agent number.
- Consult and Transfer This transfer type is equivalent to a Blind or Consultation transfer. The VoiceXML interpreter remains on the call until a successful connection is established between the caller and the agent.
- Conference and Transfer This transfer type is equivalent to a Blind or Consultation transfer. Private communication between the VoiceXML interpreter and the agent occurs while the caller is on hold.
- Bridge The application is not detached from the incoming call. When the transfer ends, control of the call always returns to the application, regardless of the transfer result.

> **Tip**
>
> The user must subscribe to AT&T Toll-Free Transfer Connect Service to use the Courtesy Transfer method. The user calls the toll-free number and the call lands on GVP.

Transfer Methods

To implement the requests for the different types of transfer at the telephony layer, the Media Control Platform can use the following SIP transfer methods:

HKF Hookflash transfer, using DTMF digits (RFC 2833):

1. The Media Control Platform sends DTMF digits on the media channel leaving it to the media gateway or switch to perform the transfer on the network.
2. The call is disconnected by either the platform or the remote end, depending on the setting of options that you can configure. Otherwise, the call is disconnected after a configured timeout.

This is a one-leg transfer (it occupies only one channel on the platform).

REFER Transfer is based on a SIP REFER message (RFC 3515):

1. The platform sends a REFER request to the caller, with the called party (as specified in the VoiceXML application) in the Refer-To: header.
2. The transfer fails if a non-2xx final response is received for the REFER.

This is a one-leg transfer.

BRIDGE The Media Control Platform bridges the media path:

1. The platform sends an INVITE request to the called party, and a dialog is established between the called party and the platform.
2. The transfer fails if a non-2xx final response is received for the INVITE request.

This is a two-leg transfer, or join-style transfer (it occupies two channels on the platform). The platform stays in the signaling path and is responsible for bridging the two call legs.

REFERJOIN Consultative REFER transfer (RFC 3891), also referred to as REFER with replaces transfer:

1. The platform sends an INVITE request to the called party, and a dialog is established between the called party and the platform.
2. The platform also sends a REFER request to the caller, with the called party s information in the Replaces header.
3. The platform treats the transfer as successful if it receives a BYE from the caller after a 2xx response for the REFER.
4. The transfer fails if a non-2xx final response is received for the INVITE request or the REFER request.

This is a two-leg transfer.

MEDIAREDIRECT Media redirection transfer. The Media Control Platform uses SIP to handle call control between the caller and the called party, and the RTP media channel is connected directly between the caller and called party:

1. The platform sends an INVITE request to the called party, without SDP.
2. If the transfer is proceeding, the called party responds with a 200 OK that includes an SDP offer.
3. The platform forwards the SDP offer in a re-INVITE request to the caller.
4. The caller responds with a 200 OK that includes the SDP answer.
5. The platform forwards the SDP answer to the called party in an ACK response.
6. The transfer fails if a non-2xx final response is received for the initial INVITE request.

This is a two-leg transfer.

Transfer Method for NEC NEAX 61 Switch

The Media Control Platform supports the NEC61ISDN transfer method, which is a single B channel Blind transfer over ISDN. This transfer method can be specified in the VoiceXML application with the `<transfer> type=blind` parameter and `gvp:method=NEC61ISDN` attribute (case insensitive).

In this case, the Media Control Platform uses the SIP REFER transfer method to trigger the PSTN Connector to perform the transfer.

Transfer Methods for AT&T Transfer Connect

GVP supports AT&T Transfer Connect in-band and out-of-band signaling to transfer call control information and data. through the PSTN Connector. To implement the requests for AT&T transfers at the telephony layer, the Media Control Platform can use the following GVP transfer methods to perform transfers:

- ATTCOURTESY, ATTCONSULT, ATTCONFERENCE These inbound-call transfers are treated like any other inbound transfer. The Media Control Platform sends a request to the gateway to trigger the DTMF transfer, and the PSTN Connector passes the transfer information to the network through Dialogic ports:
- ATTOOBCOURTESY The PSTN Connector receives an outbound trigger or request from the network through Dialogic ports:
  The PSTN Connector sends an INVITE to the platform.
  - The platform initiates call setup and the PSTN Connector sends a `gc_AcceptCall` message to the AT&T network.
  - When the call is established, the platform sends a `200 OK` message to the PSTN Connector.
  - The PSTN Connector response with an ACK response and the two way media session is established.
  - The platform sends a REFER message that includes the `X-Genesys-Transfer-Method=ATTOOBCOURTESY` and the `X-Genesys-GVP-UUI` custom headers to the PSTN Connector.
  - The PSTN Connector sends a `202 Accepted` response to the platform and then sends a FACILITY message to the network with the target party number and UUI.
  - After the network passes on the information to the target party, it sends a FACILITY message to the PSTN Connector with the transfer results (success or failure).
  - The PSTN Connector passes this information to the platform with a NOTIFY message.
  - The call is disconnected on the PSTN side, and the platform issues a BYE message to the PSTN Connector.
  - The PSTN Connector responds with `200 OK`, and the call is released.
- ATTOOBCONSULT The PSTN Connector receives an outbound trigger or request from the network through Dialogic ports:
  - The PSTN Connector sends an INVITE to the platform.
  - The platform initiates call setup the PSTN Connector sends a `gc_AcceptCall` message to the network.
  - When the call is established, the platform sends a `200 OK` message to the PSTN Connector.
  - The PSTN Connector response with an ACK response and the two way media session is established.
  - The platform sends a REFER message that includes the `X-Genesys-Transfer-Method=ATTOOBCONSULT` and the `X-Genesys-GVP-UUI` custom headers to the PSTN Connector.
  - The PSTN Connector sends a `202 Accepted` response to the platform and then sends a FACILITY message to the network with the target party number and UUI.
  - After the network passes on the information to the target party, it sends a FACILITY message to the PSTN Connector with the transfer results (success or failure).
  - The PSTN Connector passes this information to the platform with a NOTIFY message (including the Return Code).

- The platform responds with a `200 OK` message.
- The call is disconnected on the PSTN side, and the platform issues a BYE message to the PSTN Connector.
- The PSTN Connector responds with `200 OK`, and the call is released.

- ATTOOBCONFERENCE The PSTN Connector receives an inbound trigger or request from the network through Dialogic ports:.
  - The PSTN Connector receives an INVITE from the platform that includes the `X-Genesys-Transfer-Method=ATTOOBCONFERENCE` and `X-Genesys-GVP-UUI` custom headers, and the Call ID.
  - As the platform is initiating call setup, the PSTN Connector sends a FACILITY message for redirection to the network that includes the target party number and the UUI.
  - The network responds with a FACILITY ACK, and the PSTN Connector sends a 180 Ringing message, followed by a `200 OK` message to the platform.
  - The platform sends an RTCP JOIN packet which enables the PSTN Connector to retrieve the caller on hold by sending a FACILITY message to the network.
  - With the caller on hold, the PSTN Connector mutes the caller leg to the platform and activates media on the agent leg to the platform, and then enables whispering to the agent.
  - The caller is taken off hold and the PSTN Connector mutes the agent leg to the platform, and the activates media on the caller leg to the platform.
  - When the transfer is completed successfully, the PSTN Connector is connected to the target party and the FACILITY ACK from the network indicates success.
  - The PSTN Connector disconnects the call as soon as the session termination is received from the platform.

For a complete list of PSTN transfers and the supported VoiceXML transfer types, see the table PSTN Transfers and Supported VoiceXML Transfer Types.

The NGI and GVPi control the transfer method that is used, based on the value that is specified for the method attribute in the VoiceXML application. If the method is not specified, the default method for the applicable transfer type is used. The default methods are configurable (in the `sip.defaultblindxfer`, `sip.defaultconsultxfer`, and `sip.defaultbridgexfer` configuration options). In addition, configurable parameters enable you to specify whether the Media Control Platform will use the BRIDGE or MEDIAREDIRECT method if one of the other methods fails.

Because of the actual mechanisms that are involved, the SIP transfer methods do not support all transfer types. The table below summarizes SIP transfer-method support for the different types of transfer. Supported transfer methods are configured by using the `sip.transfermethods` parameter in the Media Control Platform Application.

Table: SIP Transfer Methods and Supported VoiceXML Transfer Types

SIP transfer method	Supported transfer type	Notes
HKF	• Blind • Consultation Whisper transfer supported	• The DTMF digits are flash or other configured digits, followed by a phone number. • A different configured sequence of flash and digits can be dialed to abort the transfer.

SIP transfer method	Supported transfer type	Notes
REFER	• Blind • Consultation	• The default platform method for `type=blind`. • Transfer connect timeout is not supported. • The platform can be configured to send an INVITE hold to the caller. • For type=consultation, the platform also supports the NOTIFY method for notification of the transfer result. If the transfer fails, the platform takes the original caller off hold, and the VoiceXML application proceeds with the caller. **Note:** Transfer audio does not work with the method REFER. • The platform can be configured to send a BYE request to the caller, or to wait for a BYE from the caller. • For transfer requests from the Call Control Platform, the Media Control Platform sends a REFER request to the Call Control Platform, which throws a dialog.transfer event to the CCXML application. The type attribute for the event is always set to blind, whether the request from the VoiceXML application was for a blind transfer or consultation (supervised) transfer.
BRIDGE	• Blind • Consultation • Bridge Whisper transfer supported	• The default platform method for type=bridge. • Non-whisper transfers support the `connectwhen=immediate` attribute in the VoiceXML application. If this value is specified, a one-way media path from the called party

SIP transfer method	Supported transfer type	Notes
		to the caller is established before the call is connected. • If specified in the VoiceXML application, the platform can continue to support media operations (such as handling DTMF grammars, ASR, transactional recording, and playing transfer audio) during a bridge transfer. • For transfer requests involving the Call Control Platform, if the VoiceXML application uses a <send> tag to notify the Call Control Platform about a bridge transfer request, the Media Control Platform sends a SIP INFO message to the Call Control Platform.
REFERJOIN	• Blind • Consultation Whisper transfer supported	• The default platform method for type=consultation. • The platform can be configured to send an INVITE hold to the caller. • If the transfer fails, the platform takes the original caller off hold, and the VoiceXML application proceeds with the caller. • The platform can be configured to send a BYE request to the caller and then the called party, or to wait for a BYE from the caller. • For a whisper transfer, if the called party rejects the transfer request, the platform sends a BYE to the called party to disconnect the call. • Non-whisper transfers support the connectwhen= immediate attribute in the VoiceXML application. If this

SIP transfer method	Supported transfer type	Notes
		value is specified, the media path is established between the caller and the called party as soon as the media session is ready.
MEDIAREDIRECT	• Blind • Consultation • Bridge Whisper transfer supported	• For a whisper transfer, a media channel is established between the called party and the Media Control Platform for the consultative part of the transfer, if necessary. If the called party rejects the request, the platform sends a BYE request to the called party to disconnect the call. The media path and interaction between the platform and the caller then resumes. • If the caller disconnects during the transfer (that is, if the platform receives a BYE), the platform sends a BYE to the called party to disconnect the call. • If the called party disconnects during the transfer, the platform updates the caller s media path back to the platform, using a new re-INVITE if necessary.

Table: PSTN Transfers and Supported VoiceXML Transfer Types

PSTN transfers	SIP transfer method	Supported transfer type	Notes
AT&T Courtesy Transfer	ATTCOURTESY ATTOOBCOURTESY	• Blind	• In-band and out-of-band transfers are supported. • User-to-User Information (UUI) message in transfer is supported.
AT&T Consult Transfer	ATTCONSULT ATTOOBCONSULT	• Blind • Consultation	• In-band and out-of-band transfers are supported.

PSTN transfers	SIP transfer method	Supported transfer type	Notes
			• UUI message in transfer is supported for out-of-band only. • Transfer audio is not supported.
AT&T Conference Transfer	ATTCONFERENCE ATTOOBCONFERENCE	• Blind • Consultation • Bridge Whisper transfer supported	• In-band and out-of-band transfers are supported. • UUI message in transfer is supported for out-of-band only. • Whisper transfer is supported when type=consultation.
Dialogic Bridge Transfer	BRIDGE	• Blind • Consultation • Bridge Whisper transfer supported	• UUI message in transfer is supported.
Dialogic Blind Transfer	REFER	• Blind	• UUI message in transfer is supported • When `type=blind` and `connectwhen=immediate` the REFER message is sent after the outbound call receives alerting. For any other combination, the REFER with Replaces message is sent.
Single B Channel Transfer over ISDN (for NEC NEAX 61 switch)	NEC61ISDN	• Blind	• UUI message in transfer is supported.
Two-B Channel Transfer (TBCT)	REFERJOIN	• Blind	• UUI message in

PSTN transfers	SIP transfer method	Supported transfer type	Notes
		• Consultation Whisper transfer supported	transfer is supported. • When type=blind and connectwhen=immediate the REFER message is sent after the outbound call receives alerting. For any other combination, the REFER with Replaces message is sent.
Release Link Trunk (RLT) Transfer	REFERJOIN	• Blind • Consultation Whisper transfer supported	• UUI message in transfer is supported. • When type=blind and connectwhen=immediate the REFER message is sent after the outbound call receives alerting. For any other combination, the REFER with Replaces message is sent.
Explicit Call Transfer (ECT)	REFERJOIN	• Blind • Consultation Whisper transfer supported	• UUI message in transfer is supported. • When type=blind and connectwhen=immediate the REFER message is sent after the outbound call receives alerting. For any other combination, the REFER with Replaces message is sent.
Q Signaling (Q.SIG) Transfer	REFERJOIN	• Blind	• UUI message in

PSTN transfers	SIP transfer method	Supported transfer type	Notes
		• Consultation Whisper transfer supported	transfer is supported. • When type=blind and connectwhen=immediate the REFER message is sent after the outbound call receives alerting. For any other combination, the REFER with Replaces message is sent.

Table: PSTN Connector- and NGI-supported Transfers

PSTN Connector transfer	`gvp:method` attribute	`type` attribute	`gvp:connectwhen` attribute	Whisper transfer support	UUI support
Bridge	BRIDGE	blind / consultation / bridge	immediate / answered	Yes	Yes
Hookflash	REFERJOIN	blind / consultation	N/A	Yes	Yes
AT&T Courtesy Transfer (inband)	ATTCOURTESY	blind	N/A	No	Yes
AT&T Consult and Transfer (inband)	ATTCONSULT	blind / consultation	N/A	No	No
AT&T Conference and Transfer (inband)	ATTCONFERENCE	blind / consultation / bridge	N/A	Yes	No
AT&T Courtesy Transfer (out-of-band)	ATTOOBCOURTESY	blind	N/A	No	Yes
AT&T Consult and Transfer (out-of-band)	ATTOOBCONSULT	blind / consultation	N/A	No	Yes
AT&T Conference and Transfer (out-of-band)	ATTOOBCONFERENCE	blind / consultation	N/A	Yes	Yes
Single B channel blind transfer over	NEC61ISDN	blind	N/A	No	Yes

PSTN Connector transfer	gvp:method attribute	type attribute	gvp:connectwhen attribute	Whisper transfer support	UUI support
ISDN for NEC NEAX 61 switch					
Two B-Channel Transfer (TBCT)	REFERJOIN	blind / consultation	immediate / answered	Yes	Yes
Release Link Trunk Transfer (RLT)	REFERJOIN	blind / consultation	immediate / answered	Yes	Yes
Explicit Call Transfer (ECT)	REFERJOIN	blind / consultation	immediate / answered	Yes	Yes
Q.SIG Call Transfer and Path Replacement	REFERJOIN	blind / consultation	immediate / answered	Yes	Yes

Note: REFERJOIN is "REFER with Replaces".
For TBCT/RLT/ECT/QSIG transfer with REFERJOIN: `type=blind; connectwhen=immediate.`

- The REFER is sent after the Outbound call receives alerting.

For any other combination, the REFER with Replaces is sent after the Outbound call gets connected.

Implications of Transfer Method Transfer Type Combinations

In addition to offering varying levels of support for features such as whisper transfers and connection timeouts, the different combinations of SIP methods and VoiceXML transfer types can result in scenarios that can have significant implications for metrics, for general component activity logs, and, therefore, for GVP Reporting.

IPv6 Support in Inbound and Outbound Transfers

The Media Control Platform supports IPv4 and IPv6 when performing transfers. Transfers that create outbound calls are created independent of the IP version that is used on the inbound call, and follow the same rules as specified for a single outbound call. However, due to the nature of SDP negotiation, the MediaRedirect SIP transfer method might not work if both the inbound and outbound calls are using different IP versions.

Conferencing

In essence, conferencing is a special type of bridge transfer. The Media Control Platform supports conferencing in NETANN format and through MSML dialogs. The Conference application module handles calls and manages call interactions with the conference bridge for NETANN and MSML conferencing.

NETANN

Calls join a conference directly, by specifying the conference-bridge identifier (conference ID) in the SIP Request-URI in NETANN format: `<sip:conf=<conf ID>@host.com`

Platform-level configuration options (in the conference configuration section) support standard NETANN conference requirements, such as configurable participant roles (talk-only, listen-only, or full duplex), audio-gain parameters, and the video-output

algorithm (first, loudest, or no video).

MSML

Calls join a conference directly, by specifying the conference bridge identifier (conference ID) in the SIP Request-URI in the MSML dialog: `sip:msml=conf-ID@<RM-IPaddress>`

Platform-level configuration options (in the conference configuration section) support standard MSML conference requirements, by providing different roles for each MSML conference leg, such as regular, agent, coach, monitor, push-, or pull-all, and by providing MSML video conferencing.

Debugging VoiceXML Applications

The NGI interfaces with the debug client GUI that is part of Genesys Composer.

If the real-time debugger is enabled (in the vxmli.debug.enabled configuration option), information about calls is passed between the NGI and the debugger client in SIP INVITE and 18x messages.

The debugger can skip or step through the NGI execution, execute JavaScript snippets, provide information about currently executing elements, and change some of the parameters for the elements being executed.

The NGI can also save to the file system all the information related to the transactions of a call. This feature is helpful for debugging both platform operations and VoiceXML applications.

HTTP Basic Authentication

The Media Control Platform's NGI supports a username and password for HTTP Basic Authentication. A GVP extension attribute of the <data> element takes an ECMAScript expression and evaluates it to a string. The interpreter then passes these values to the Fetching Module as the username and password authentication.

Masking sensitive data in MCP trace logs

To mask senstive data in MCP trace logs:

1. Set **[log] mask_sensitive_data** to `true`. This would only mask the Trace logs, but would not impact the Interaction logs containing the same information.
2. For Interaction logs with sensitive information, use **[ems] logconfig.MFSINK** to remove log entries with sensitive data.
3. Use the `gvp:private` flag as follows to mask sensitive content in Interaction logs.
   Resolved by set in the defaults-ng.vxml file:

   ```
   <property name="com.genesyslab.private" value="true"/>
   ```