Genesys

Genesys Voice Platform 8.1

User's Guide

About Genesys

Genesys is the world's leading provider of customer service and contact center software - with more than 4,000 customers in 80 countries. Drawing on its more than 20 years of customer service innovation and experience, Genesys is uniquely positioned to help companies bring their people, insights and customer channels together to effectively drive today's customer conversation. Genesys software directs more than 100 million interactions every day, maximizing the value of customer engagement and differentiating the experience by driving personalization and multi-channel customer service - and extending customer service across the enterprise to optimize processes and the performance of customer-facing employees. Go to www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Customer Care. Before contacting technical support, please refer to the ***Genesys Care Program Guide*** for complete contact information and procedures.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the ***Genesys Licensing Guide***.

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 81gvp_us_07-2013_v8.1.701.00

Table of Contents

List of Procedures

Preface

Welcome to the *Genesys Voice Platform 8.1 User's Guide*. This document provides detailed information about configuring, provisioning, and monitoring Genesys Voice Platform (GVP) and its components.

This document is valid for the 8.1 release of this product.

Note: For versions of this document created for other releases of this product, visit the Genesys Technical Support website, or request the Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.

This preface contains the following sections:

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on page 559.

About Genesys Voice Platform

GVP is a group of software components that constitute a robust, carrier-grade voice processing platform. GVP unifies voice and web technologies to provide a complete solution for customer self-service or assisted service.

In the Voice Platform Solution (VPS), GVP 8.1 is fully integrated with the Genesys Management Framework. GVP uses Genesys Administrator, the standard Genesys configuration and management Graphical User Interface (GUI), to configure, tune, activate, and manage GVP processes and GVP voice and call control applications. GVP interacts with other Genesys components and can be deployed in conjunction with other solutions, such as Enterprise

Routing Solution (ERS), Network Routing Solution (NRS), and Network-based Contact Solution (NbCS).

Note: GVP is a scalable solution with flexible configuration and deployment options.

Intended Audience

This document, primarily intended for system integrators and administrators, assumes that you have a basic understanding of:

- Computer-telephony integration (CTI) concepts, processes, terminology, and applications.
- Network design and operation.
- Your own network configurations.

You should also be familiar with the Genesys Framework architecture.

Making Comments on This Document

If you especially like or dislike anything about this document, feel free to e-mail your comments to `Techpubs.webadmin@genesyslab.com`.

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document. Please limit your comments to the scope of this document only and to the way in which the information is presented. Contact your Genesys Account Representative or Genesys Technical Support if you have suggestions about the product itself.

When you send us comments, you grant Genesys a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

Contacting Genesys Technical Support

If you have purchased support directly from Genesys, please contact `Genesys Technical Support`.

Before contacting technical support, please refer to the *Genesys Care Program Guide* for complete contact information and procedures.

Document Change History

This section lists topics that are new or that have changed significantly since the first release of this document.

Release 8.1.7

- Chapter 3, "Configuring Common Features," on page 41:
  - Added information on making TCP the preferred protocol to Task Summary: Configuring SIP Communications and Routing, on page 42.
  - Added information on MRCP V2 Client options to Table 4, "Default SIP Transports," on page 44.
  - Added information on the components MRCP and RM, and the option `logconfig.TRAPSINK`, to Table 6, "Default Log and Metrics Filters," on page 64.
- Chapter 4, "Configuring the Resource Manager," on page 87:
  - Modified the list of valid and example values for the `IP Type of Service for RTP/RTCP` variable in Table 13, "Selected Resource Manager Configuration Options," on page 89.
- Chapter 7, "Configuring the Media Control Platform," on page 145:
  - Modified the list of valid and example values for the `IP Type of Service for RTP/RTCP` variable in Table 23, "Selected Media Control Platform Configuration Options," on page 160.
- Chapter 9, "Configuring the Call Control Platform," on page 213:
  - Added information about specifying use of the `sips:` schema and controlling SIP Secure Mode to Table Task Summary:, "Configuring the Call Control Platform," on page 213.
  - Modified the list of valid and example values for the `IP Type of Service for RTP/RTCP` variable in Table 26, "Selected Call Control Platform Configuration Options," on page 217.
- Chapter 16, "Reporting Overview," on page 297:
  - Added references to the Service Quality reports Call Failures, Call Summary, and Latency Details in the section "Generating a Report with GAX" on page 302 and the section "GAX Report Generation Table" on page 304.
  - Rearranged sections to place parallel GA and GAX procedures together, modified titles, and added notes to sections that apply to both.
- Chapter 19, "Historical Reports," on page 341:
  - Added a note about in ASR/TTS Usage Peaks report for MRCPv1 and MRCPv2 in the section "ASR/TTS Usage Peaks" on page 358.
- Appendix A, "Module and Specifier IDs," on page 391:
  - Revised (additions, corrections, and restorations) Table 76, "CCP Media Controller Events," on page 418 and Table 77, "CCXMLI Log_4 INFO Events," on page 420.

- Appendix B, "Media Control Platform Reference Information," on page 439:
  - Added SIP Response codes to Table 97, "Error Response Handling—Outbound Calls," on page 453.
  - In the section "Dynamic Media Control Platform Parameters" on page 446, removed the configuration options `inbandxferprefix` and `inbandxfertimeout`.
- Appendix K, "SIP Customizable Headers and Parameters," on page 563:
  - Added an example VoiceXML page that will perform configuration that is described in the above paragraph, using `signalvar on` page 566.

Release 8.1.6

- "Preface," on page 13:
  - Revised a bullet point about SQ reports and NGi VXML apps in the section "Release 8.1.2" on page 19.
- Chapter 3, "Configuring Common Features," on page 41:
  - Added a note about SQ reports and NGi VXML apps to "Service Quality Analysis (SQA)" on page 65.
- Chapter 4, "Configuring the Resource Manager," on page 87:
  - Added the `FIPS Mode Enabled` option to Table 13, "Selected Resource Manager Configuration Options," on page 89.
- Chapter 6, "Provisioning IVR Profiles," on page 107:
  - Added new service types `cpd` and `treatment` to Table 16, "IVR Profile Configuration Options," on page 113.
  - Added new usage limits configuration options to (and rearranged existing limits in) Table 16, "IVR Profile Configuration Options," on page 113.
  - Added the section "Operational Parameter Management and Self-Service Applications" on page 127.
- Chapter 7, "Configuring the Media Control Platform," on page 145:
  - Added the section "callmgr Section" on page 160 and the `FIPS Mode Enabled` option to the table "Selected Media Control Platform Configuration Options".
  - Added the SDP parameter `maxptime` to the table "Selected Media Control Platform Configuration Options".
  - "callmgr Section" on page 160 and the `FIPS Mode Enabled` option to the table "Selected Media Control Platform Configuration Options".
- Chapter 8, "Configuring the MRCP Proxy," on page 199:
  - Updated the `FIPS Mode Enabled` option in Table 25, "Selected MRCP Proxy Configuration Options," on page 203.
- Chapter 9, "Configuring the Call Control Platform," on page 213:
  - Added the `FIPS Mode Enabled` option to the table Table 26, "Selected Call Control Platform Configuration Options," on page 217.

- Chapter 9, "Configuring the Call Control Platform," on page 213:
  - Restored the default audio codec value `3|gsm|audio/x-gsm|8000|1` to Table 26, "Selected Call Control Platform Configuration Options," on page 217.
- Chapter 10, "Configuring the CTI Connector," on page 229:
  - Added the `FIPS Mode Enabled` option to the table Table 27, "Selected CTI Connector Configuration Options," on page 234.
  - Added the section "CTIC (Genesys) and Treatments" on page 242.
  - Added the section "Multiple Trunk Group ID support for CTI Connector (ICM)" on page 244.
  - Added the section "CTI Connector (ICM) and ECC Variables" on page 245.
  - Added the section "CTIC (ICM) Parameter Notes" on page 246.
- Chapter 11, "Configuring the Supplementary Services Gateway," on page 249:
  - Added the `FIPS Mode Enabled` option to the table Table 30, "Selected SSG Configuration Options," on page 251.
- Chapter 14, "Configuring the Reporting Server," on page 275:
  - Added the section "Disabling CDR Storage for Resource Manager and Media Control Platform" on page 280.
- Chapter 16, "Reporting Overview," on page 297:
  - Modified the section "Generating a Report with GA" on page 298 to contain the existing "Generating a Report Using Genesys Administrator" on page 298 and the new "Generating a Report Using GAX" on page 303.
  - Added the section "GAX Report Generation Table" on page 304.
- Chapter 19, "Historical Reports," on page 341:
  - Added a note about SQ reports and NGi VXML apps to "Overview" on page 341.
  - Added the section "Media Service Call Peaks" on page 355.
  - Added the section "MCP VXML Call Arrivals" on page 356.
  - Added the section "MCP VXML Call Peaks" on page 356.
  - Call Peaks Reporting now supports the component-type MCP/VXML: Media Platform peaks for VXML calls based on MCP CDR records (8.1.6). See Table 63 on page 369.
  - Added the field CODEC to Table 63, "Per-Call IVR Actions Report Fields," on page 369.
- Appendix B, "Media Control Platform Reference Information," on page 429:
  - Added RTP transport for MP3 audio format to the section "RTSP Server Support" on page 436.

- Appendix C, "Tuning Call Progress Detection," on page 449:
  - Added the section "Continuous Tone Detection" on page 457.
- Appendix G, "Video Support," on page 497:
  - Added VP8 support to Table 104, "MIME Types," on page 498.
- Appendix I, "SSG HTTP Interface," on page 509:
  - Added the section "SSG Database Queue Clearing During a Restart" on page 528.
- Added Appendix K, "SIP Customizable Headers and Parameters," on page 553.
- Restored the MCP-supported codec GSM to these tables:
  - Table 23, "Selected Media Control Platform Configuration Options," on page 160
  - Three instances in Table 89, "Supported Audio File Formats—Play," on page 430
  - Two instances in Table 92, "Supported Audio File Formats—Record," on page 433
  - Table 94, "Supported Audio/Video File Formats—Record," on page 436

Release 8.1.5

- A new section has been added to describe how to enable IPv6 communications. See Chapter 3.
- New options were added to Table 13 to describe Resource Manager configurations. See Chapter 4.
- New options were added to Table 23 and Table 24 to describe Media Control Platform and MRCP Server configurations. See Chapter 7.
- New options were added to Table 26 to describe Call Control Platform configurations. See Chapter 9.
- Any mention of the GSM codec has been removed from this document. (It is no longer supported.)
- New options were added to Table 30 to describe Supplementary Services Gateway configurations. See Chapter 11
- A new chapter has been added to describe how to configure GVP in a multi-site environments. See Chapter 15.
- A new subsection, "Per-Call IVR Actions Report" has been added to the Historical Call Browser section. See Chapter 19.
- A new subsection, "Video Text Overlay" has been added to the VoiceXML Features section. See Appendix G.
- Service Quality reports apply to NGi VoiceXML applications, and are found in Genesys Administrator. GVP 8.1.5 and thereafter are NGi-only platforms unless you run MCP 8.1.4 to incorporate support for GVPi applications.

Release 8.1.4

- A new chapter has been added to describe how to implement and configure the Genesys Voice Platform Policy Server component. See Chapter 5.
- A new chapter has been added to describe how to implement and configure the Genesys Voice Platform MRCP Proxy component. See Chapter 5.
- New Task Summary, configuration options, and a description of ICM messages and data formats has been added to describe CTI Connector integration with Cisco ICM. See Chapter 10.
- Two new tables have been added to describe the columns and metrics for the new ASR/TTS Usage and ASR/TTS Usage Peaks reports.

Release 8.1.3

- Reporting Server has been enhanced to operate in a mode that does not require a back end persistent database. This mode of operation is optional and still allows for support of the real-time reports such as, the dashboards and the Active Call Browser report.
- Media Control Platform, Call Control Platform, and CTI Connector now support SIP static routing with an active-active Resource Manager pair.

Release 8.1.2

- PSTN Connector information has been added in this release.
- AT&T Out-of-Band transfers have been added to this release.
- Mapping IVR Profiles to DIDs has been enhanced in this release.
- The DID Group Bulk Operation Wizard is new for this release.
- Hierarchical Multi-Tenancy (HMT) information has been added in this release.
- The operational reports now include Tenant Call Arrivals and Tenant Call Peaks reports for this release.
- Service Quality reports are included for this release, apply to NGi VoiceXML applications, and are found in Genesys Administrator.
- The Voice Platform Dashboard now includes information on Supplementary Services Gateway and PSTN Connector for this release.
- The Fetching Module functionality is no longer contained in a separate component. The functionality is now part of the Media Control Platform and the Call Control Platform.

Release 8.1.1

- Supplementary Services Gateway information has been added in this release.
- Information on the HTTP interface for the Supplementary Services Gateway has been added to this release.
- For all GVP components, the option names have been changed to the display names as seen in Genesys Administrator.
- Information on how to configure client side connections for each GVP component has been added to this release.
- Information on the Media Server Markup Language (MSML) application module for the Media Control Platform has been added to this release.

- Information on the VAR API has been added to this release.
- New audio and video codecs have been added to this release.
- The VAR Call Browser report has been merged with the Historical Call Browser report for this release.
- The Voice Platform Dashboard now includes information on outbound calls for the Supplementary Services Gateway.

Chapter

1 Introduction

Genesys Voice Platform (GVP) is a software suite that integrates call processing, reporting, management, and application servers with Voice over IP (VoIP) networks, to deliver Web-driven dialog and call control services to callers.

This chapter introduces the GVP components and Genesys Administrator, the GUI for configuring and managing GVP. It contains the following sections:

About GVP

This section describes the GVP component applications and other objects in a GVP configuration:

- GVP Components
- IVR Profiles (see page 24)
- GVP MIBs (see page 24)

GVP Components

GVP comprises the following components:

- **Resource Manager**—Functions as a SIP Proxy that controls access and routing to all resources in a GVP deployment. It also functions as a SIP Registrar, and monitors the health of GVP resources in the deployment. Its functions include:
    - Allocates and monitors resources.
    - Manages sessions.
    - Selects services.

- Enforces policies.

- **Policy Server**—Provides validation and resolution of GVP-specific business rules to Genesys Administrator through an HTTP interface with Management Framework. Its functions include:
  - Manages and validates Direct Inward Dialing (DID) numbers.
  - Provides static analysis and validation of Resource Manager tenant and IVR policies.
- **Media Control Platform**—Provides media-centric services to other GVP components, and to third-party gateways that use GVP services. The Media Control Platform is responsible for the execution of supported Voice Extensible Markup Language (VoiceXML) applications. Its functions include:
  - Initiates, answers, transfers, and disconnects calls.
  - Plays audio and Text-to-Speech (TTS) prompts.
  - Handles Automatic Speech Recognition (ASR) and dual tone multi-frequency (DTMF) inputs.
  - Provides conference services.
- **MRCP Proxy**—Acts as a proxy for all MRCPv1 Real-Time Streaming Protocol (RTSP) resource traffic within a GVP deployment. Its functions include:
  - Provides resource management for the MRCPv1 speech resource traffic.
  - Provides load balancing for MRCPv1 speech resources.
  - Processes periodic updates from Management Framework for its `Applications` and resources.
  - Sends ASR and TTS usage data for tenants, IVR Profiles, or the entire deployment to the Reporting Server.
- **Call Control Platform**—Provides call control capability in accordance with the supported W3C Call Control Extensible Markup Language (CCXML) standard. The Call Control Platform is optional in a GVP deployment. It operates as a SIP Back-to-Back User Agent (B2BUA) for requests to and from GVP components. Its functions include:
  - Accepts, rejects, and redirects calls, including handling call setup information to enable intelligent routing and selective answering. Call-handling capabilities include supervised transfer, whispering, and call hold.
  - Creates outbound calls through third-party gateways.
  - Uses Media Control Platform services to initiate VoiceXML dialogs, start conferences, and perform implicit transcoding.
  - Provides multi-party conference support with moderator and floor control capabilities.
  - Provides personal assistant services, such as dialing from a personal address book or corporate directory, managing personal appointments, and managing voicemail and e-mail.

For information about creating CCXML applications that use Call Control Platform capabilities, see the *Genesys Voice Platform 8.1 CCXML Reference Manual.*

- **Reporting Server**—Stores and summarizes data and statistics submitted by Reporting Clients to provide near real-time reports by hour, day, week, and month. Reporting Clients on the Resource Manager, Media Control Platform, and Call Control Platform send call detail records (CDRs), Metrics, and Operational Reporting (OR) statistics to the Reporting Server. The Reporting Server provides an XML web services interface that is used by Genesys Administrator to obtain GVP reporting information. The XML web services interface is also accessible to any HTTP client, providing customers with access to GVP reporting outside of Genesys Administrator.

  For information on the real-time and historical reports, see "Monitoring GVP" on page 295.

- **CTI Connector**—Provides additional computer telephony integration (CTI) functionality by connecting to the IVR Server, which is part of the larger Genesys Suite, through Media Control Platform. As a result, CTI Connector remains in the call path to receive and pass call data between IVR Server and the Media Control Platform. Its functions include:
  - Routes calls to Universal Routing Server (URS).
  - Call processing treatments.
  - Transfers user data
  - Transfers through CTI
  - Remote switch transfers
  - Receives statistical data
  - Performs user/interaction data operations.

- **Supplementary Services Gateway**—Provides call processing services to the application layer through HTTP. Its functions include:
  - Outbound MCP session initiation.

- **PSTN Connector**—Provides connectivity to traditional telephony environments such as Public Switched Telephone Networks (PSTN) switches. Its functions include:
  - Media Gateway functionality (uses Dialogic boards on the TDM side).
  - Call Progress Analysis.
  - All transfers including various switch specific network transfers.

For detailed information about how the GVP components perform their functions, see Chapter 3, "How GVP Works" in the *Genesys Voice Platform 8.1 Deployment Guide*.

For information about the Genesys Voice Platform and component architecture, see Chapter 2, "GVP Architecture" in the *Genesys Voice Platform 8.1 Deployment Guide.*

IVR Profiles

Voice Extensible Markup Language (VoiceXML) and Call Control Extensible Markup Language (CCXML) are the application-level languages that are used to construct voice and call control applications that control the interaction between the external user and the GVP software.

Voice and call control applications are configured as `IVR Profile` objects in Genesys Administrator. The IVR Profiles define how requests received by the VPS are translated into concrete service requests GVP which components within the deployment can execute.

GVP MIBs

The MIB Installation Package (IP) contains the Management Information Base (MIB) files that GVP uses to support Simple Network Management Protocol (SNMP).

For general information about SNMP in a GVP deployment, see Chapter 2, "GVP Architecture" in the *Genesys Voice Platform 8.1 Deployment Guide*. For detailed information about the MIBs, see the *Genesys Voice Platform 8.1 Troubleshooting Guide.*

Genesys Administrator

Genesys Administrator is a Web-based user interface for the management and configuration of Genesys components.

Use Genesys Administrator to deploy, configure, provision, and monitor GVP.

Figure 1 shows a typical Genesys Administrator page.

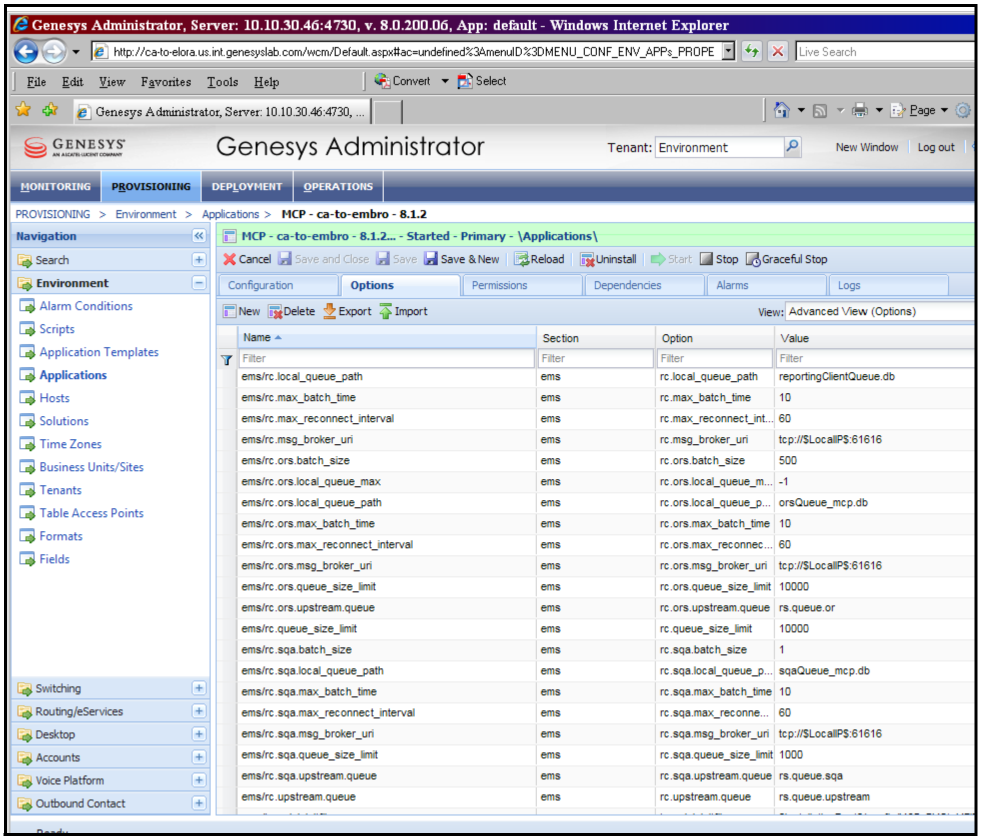


Figure 1: Genesys Administrator

To access Genesys Administrator in your Genesys deployment, go to the following URL:

`http://<Genesys Administrator host>/wcm`

More Information

- For information about installing Genesys Administrator, see the *Framework 8.1 Genesys Administrator Deployment Guide.*
- For general information about using Genesys Administrator, see the online *Framework 8.1 Genesys Administrator Help.*
- For information about using Genesys Administrator to configure and provision GVP `Application` objects and IVR Profiles, see Procedure: Viewing or modifying GVP configuration parameters, on page 34 and Procedure: Configuring logical resource groups, on page 94.
- For information about using Genesys Administrator to monitor GVP and view reports, see Part 2 of this manual, starting on page 295.

Note: The Genesys Administrator's enable and disable features for its objects has no impact on the GVP objects.

GVP Identifiers and SIP Headers

This section explains two important categories of identifiers that are used in GVP:

- Session Identifiers
- Application Identifiers (see page 27)

Session Identifiers

There are three types of session identifiers that are used to track, co-ordinate, and report on GVP sessions. Table 1 describes the session identifiers and the SIP extension headers in which the ID information is captured.

Table 1: GVP Session Identifiers and SIP Headers

Session ID	Description	SIP Header
Genesys CallUUID	The Universally Unique Identifier (UUID) that T-Server or SIP Server generates for the customer interaction.	`X-Genesys-CallUUID` The Resource Manager (SIP Proxy) and GVP components (User Agents) propagate this header, without changes, in all SIP messages.
GVP Session ID	The 128-bit Globally Unique Identifier (GUID), and other GVP specific parameters separated by semicolons that identifies a call session for a particular GVP resource. The GUID is generated by: Resource Manager when it creates a new session in response to a new SIP `INVITE` request. Media Control Platform or Call Control Platform when it initiates a new session for an outbound call. It is passed in the following format: `<GUID>[;gvp.rm.cti-call=1][;gvp.rm.datanode=<datanode>][;gvp.rm.tenant-id=<tenant info>]`	`X-Genesys-GVP-Session-ID` If the header does not already exist in the SIP request, the Resource Manager inserts the header in requests that it forwards. The Resource Manager and GVP components propagate this header in all subsequent SIP messages for the session.

Table 1: GVP Session Identifiers and SIP Headers (Continued)

Session ID	Description	SIP Header
GVP Component ID	The ID that the GVP component generates to identify the call leg. The component correlates the Component ID with the GVP Session ID, and logs the correlation for correct call detail records (CDRs). For more information on the CDR reports, see "Monitoring GVP" on page 295.	`ccxml-session-id` For the Call Control Platform only, when it sends a request to initiate an outbound session, but the Resource Manager session has not been assigned yet. This enables a newly created CCXML session to make multiple SIP requests before it has received a response to any of them.

Application Identifiers

There are two kinds of *applications* in a GVP deployment:

- The GVP components or processes, which exist as `Application` objects in the Genesys Configuration Layer.
- The VoiceXML and CCXML applications, which exist as `IVR Profile` objects in the Genesys Configuration Layer.

Table 2 describes the identifiers that GVP uses for both kinds of applications, and the SIP extension headers in which the ID information is captured.

Table 2: GVP Application Identifiers and SIP Headers

Application ID	Description	SIP Header
Application DBID	The DBID that the Configuration Layer assigns to the GVP component `Application` object. This ID is used internally by Reporting Server to link accumulated call data and summary data with specific GVP components.	Not Applicable

Table 2: GVP Application Identifiers and SIP Headers (Continued)

Application ID	Description	SIP Header
IVR Profile name	The user-defined name that was assigned to the IVR Profile when it was created. **Note:** For backwards compatibility: If gvp-tenant-id = [TenantX], Resource Manager assumes that the associated tenant for the call is TenantX. If gvp-tenant-id = IVRAppY, Resource Manager assumes that the associated IVR Profile for the call is IVRAppY. If the X-Genesys-gsw-ivr-profile-id header is also present, it is used to determining the IVR Profile and the Request URI parameter denotes the tenant. If gvp-tenant-id = [TenantX].IVRAppY, Resource Manager assumes that the associated tenant for the call is TenantX, and the IVR Profile is IVRAppY. If the X-Genesys-gsw-ivr-profile-id header is also present, it is ignored.	• gvp-tenant-id parameter in the SIP Request-URI—The Resource Manager uses this parameter, if it is present, to identify the voice or call control application for a new session. • gvp.rm.tenant-id parameter in the X-Genesys-GVP-Session-ID extension SIP header—The Resource Manager inserts this parameter in the header before it forwards the initial session request. • X-Genesys-gsw-ivr-profile-name
IVR Profile DBID	The DBID that the Configuration Layer assigns to the IVR Profile. This ID is used internally by Reporting Server to link call level and summary data to a specified IVR Profile.	X-Genesys-RM-Application-dbid The Resource Manager adds this header to the initial INVITE request to a resource. Resources log the DBID in their CDRs to the Reporting Server.
IVR Profile ID	The ID that Resource Manager uses to map a new RM session to the IVR Profile.	X-Genesys-gsw-ivr-profile-id
Campaign ID for MSML	The ID that Resource Manager uses to assign an outbound campaign-id value in order to establish a SIP session from the same campaign on the same Media Server. **Note:** The campaign-id is not mandatory for all of the outbound modes. If an msml service request is received without the campaign-id, Resource Manager will route the request to any available Media Server.	X-Genesys-gsw-session-dbid

Table 2: GVP Application Identifiers and SIP Headers (Continued)

Application ID	Description	SIP Header
Gateway Header for PSTN Connector	The header that Resource Manager uses to inform SIP Server of the identity of the PSTN Connector from which the call originated.	`X-Genesys-GVP-Trunk-Prefix`

Importing and Exporting Configuration Server Data

When data is exported from Configuration Server, and then imported back with or without modification, be aware that the DBIDs of existing configuration objects (such as GVP `Application` processes and IVR Profiles) are not preserved when imported or exported. In these cases:

- Reporting Server will not be able to correlate historical data with the new IDs, and GVP reports will not display older data.
- GVP components that use CCILib may encounter problems, because they will no longer receive updates for objects for which they registered to receive updates under the old IDs. Restarting Configuration Server will fix this problem.

Part

1 Provisioning GVP

This part of the Guide provides information about Genesys Voice Platform (GVP) configuration and provisioning that you perform on the `Provisioning` tab of Genesys Administrator.

This information appears in the following chapters:

Chapter

2 Configuration and Provisioning Overview

This chapter provides an overview of the tasks to configure Genesys Voice Platform (GVP) components and provision GVP. It contains the following sections:

Note: This guide assumes that you have deployed a basic GVP as described in the *Genesys Voice Platform 8.1 Deployment Guide*. For more information about installing GVP components and providing the basic connections, see the *Deployment Guide*

Configuring GVP

The GVP components are configured as `Application` objects in the Genesys Configuration Layer. To deploy the Voice Platform Solution (VPS), you must create and configure the required `Application` objects in Genesys Administrator. For information about creating and deploying the GVP `Applications`, see the *Genesys Voice Platform 8.1 Deployment Guide.*

To process calls in GVP 8.1, you must provision the IVR Profiles in Genesys Administrator. To trigger the execution of a particular VoiceXML or CCXML application when an incoming call is received, map the IVR Profile to a DN range. For more information about provisioning IVR Profiles and, if required, mapping them to DNs, see Chapter 6 on page 107.

Warning! Genesys recommends, that in a multi-tenancy deployment, the service provider or enterprise manager of GVP should be the only user that can manage DID groups or define Tenants. Tenants should not be given access to edit their own configurations because of a potential conflict in numbering and naming uniqueness required by GVP. However, Tenant users can be assigned to read their configurations and reports.

Configuring GVP Processes in Genesys Administrator

The following procedure describes how to configure GVP `Application` and `IVR Profile` objects in Genesys Administrator. For more information about using Genesys Administrator, see the online *Framework 8.1 Genesys Administrator Help*.

Procedure: Viewing or modifying GVP configuration parameters

Purpose: To describe the general method for using Genesys Administrator to view or modify configuration options in GVP `Application` and `IVR Profile` objects.

Prerequisites

- The `Application` or `IVR Profile` object has been created as described in the *Genesys Voice Platform 8.1 Deployment Guide*. In particular, for GVP `Application` objects, the `Application` was created from an `Application Template` into which metadata had been imported.
- You are logged in to Genesys Administrator. To access Genesys Administrator, go to the following URL:

  `http://<Genesys Administrator host>/wcm`

Start of procedure

1. In Genesys Administrator, go to the `Options` tab of the object that you want to configure:
   - For a component `Application`, go to the `Provisioning > Environment > Applications > <Component Application> > Options` tab.
   - For an IVR Profile, go to the `Provisioning > Voice Platform > IVR Profile > Options` tab.

   Figure 2 shows the `Options` tab.

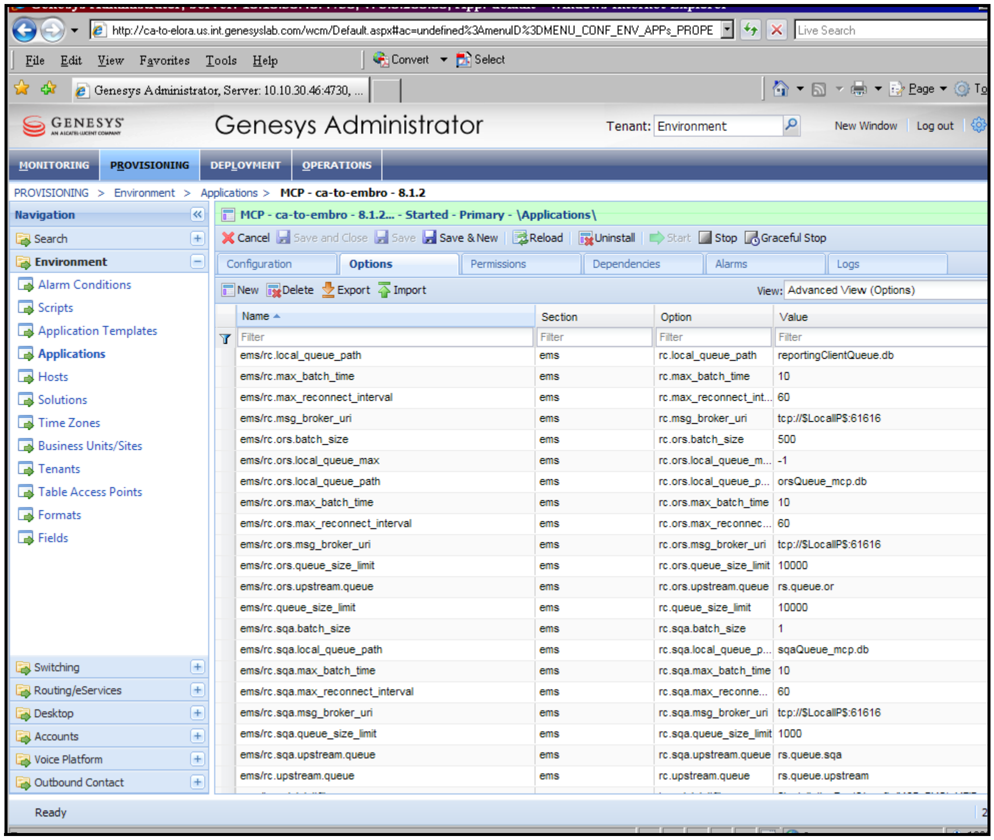


Figure 2: Genesys Administrator Options Tab

For each configurable parameter, the `Options` tab displays the following information:

- The option display name.
- The configuration section that contains the option.
- The configuration option name provided by the template.
- The current option value, either user-defined or default.

2. You can change the display in a number of ways:

- To sort the information in ascending or descending order by column, click the column header, and then select the desired sort order option from the drop-down list.
- To show or hide a column, click any column header, select the `Columns` submenu from the drop-down list, and then select or clear check boxes in the `Columns` list to show or hide columns.

3. To change an option setting:

a. Double-click the Value of the option that you want to change.

You can view the same information for all configuration options in the *Genesys Voice Platform 8.1 Configuration Options Reference.*

b. Enter the new value in the `Value` field.

4. To save your changes, click `Save` and then `Apply`.
5. To update the metadata descriptions (from an updated Templates XML file), click `Reload`. This reloads the metadata file without affecting configured option values.
6. To view the Help documentation for an option, click the blue arrow to the left of the option name.

End of procedure

Table 3 provides information about the options in the `rptui` configuration section of the `default Application` object. Table 3 provides parameter descriptions as well as the default parameter values that are preconfigured in the `default Application` object.

The `default Application` object is created automatically and is always available when you start Genesys Administrator.

Table 3: Reporting UI Configuration Options—default Application

Option Name	Description	Valid Values and Syntax
Minimum Dashboard Refreshing Interval	The minimum refresh interval, in seconds, to refresh the Voice Platform Dashboard data.	An integer greater than zero. **Default value:** `10`
Maximum Number of Items in the Dashboard	The maximum number of IVR Profiles and/or Components that can be filtered at one time.	An integer range greater than zero. **Default value:** `50`
Daylight Saving Hours	The daylight savings time difference, in hours and minutes, to apply to timestamps to adjust to local time. When specifying the value, use leading zeros if necessary.	`<HH>:<mm>` where: • `<HH>` indicates hours. • `<mm>` indicates minutes. **Default value:** `01:00`
Reporting Server HTTP Timeout	The timeout, in seconds, for communications between GVP Reports and the Reporting Server. If your deployment experiences frequent timeouts, increase this value.	Any positive integer. **Default value:** `30` (seconds)

Table 3: Reporting UI Configuration Options—default Application (Continued)

Option Name	Description	Valid Values and Syntax
Show Local Time	Specifies whether GVP Reports will display date and time values in local time, rather than in Greenwich Mean Time (GMT), which is the default format that Reporting Server returns. To display local time in reports, set this option to `true` (1) and specify a `timezone offset` (see `Timezone Offset`).	• `true (1)`—Date and time values will display in local time. • `false`—Date and time values will display in GMT. **Default value:** `true`
Timezone Offset	The time offset, in hours and minutes, that will be applied to convert GMT to local time (see `Show Local Time`), in the time zone where GVP reports will be accessed. Dates and times in all GVP reports will be converted. When specifying the value, use leading zeros if necessary.	`<s><HH>:<mm>` where: • `<s>` is either + (plus) or - (minus), to indicate whether the time should be added or subtracted. • `<HH>` indicates hours. • `<mm>` indicates minutes. **Default value:** `-08:00`

Task Summary: Configuring GVP

Task Summary: Configuring and Provisioning GVP provides an overview of the tasks to implement full GVP functionality in your deployment.

Task Summary: Configuring and Provisioning GVP

Objective	Related Procedures and Actions
Set up connections, SIP communications, and routing between the Resource Manager and all the other GVP components.	See: • "Configuring SIP Communication and Routing" on page 42. • For secure communications, see "Enabling Secure Communication" on page 46. See also component-specific requirements: • For the Media Control Platform, see Table Task Summary: on page 146. • For the Call Control Platform, see Table Task Summary: on page 213. • For the CTI Connector, see "Configuring the CTI Connector" on page 229.

Task Summary: Configuring and Provisioning GVP (Continued)

Objective	Related Procedures and Actions
Provision the resources for the Resource Manager.	See “Configuring Logical Resource Groups” on page 93.
Provision the IVR Profiles.	See Chapter 6 on page 107. • For additional specific configuration for GVPi IVR Profiles, see • To modify legacy VoiceXML applications to work with GVPi, see the *Genesys Voice Platform 8.1 Application Migration Guide*.
(Optional) Deploy Policy Server	• Enables Genesys Administrator to manage DID Groups and Resource Manager policies.
Enable GVP Reporting.	• Configure the options in the `ems` configuration section of the Resource Manager, Media Control Platform, Call Control Platform, Fetching Module, and CTI Connector `Application` objects. For more information, see “Configuring Reporting” on page 63. • Configure the Reporting Server (see Chapter 14 on page 275). • If required, configure access control for Reporting services (see “Controlling Access to Reporting Services” on page 287). • On the `Monitoring` tab of Genesys Administrator, verify that the `Voice Platform` view appears in the navigation panel. If necessary, modify the `default` (Configuration Manager) `Application` configuration to enable GVP reports to be displayed in Genesys Administrator. For more information, see “” on page 289.
(Optional) Enable Automatic Speech Recognition (ASR) and Text-to-Speech (TTS).	See “Enabling ASR and TTS” on page 150. To configure the MRCP Proxy, see Chapter 8 on page 199.
(Optional) Enable conferencing.	See “Enabling Conference Services” on page 62.

Task Summary: Configuring and Provisioning GVP (Continued)

Objective	Related Procedures and Actions
(Optional) Configure individual components to customize or enable GVP features.	In general, see the remaining chapters in the Provisioning part of this guide. More specifically, to customize: • Logging behavior, see "Service Quality Analysis (SQA)" on page 65. • Session behavior and performance, see "Configuring Session Timers and Timeouts" on page 80. • Messaging, see "Configuring SNMP" on page 72 and Table 100 on page 470. • Call Control Platform device profiles, see "Configuring Device Profiles" on page 488. • Caching behavior, see "Configuring the Squid Caching Proxy" on page 272.

Chapter

3 Configuring Common Features

This chapter describes how to implement functionality that is shared across all the components in a Genesys Voice Platform (GVP) deployment. It contains the following sections:

This chapter describes selected configuration options (parameters) that are common to GVP components. Later chapters similarly highlight important configuration options that are more component-specific.

Note: Configuration options and parameters are one in the same, and these terms are used interchangeably throughout the chapter.

The configuration option tables in this chapter provide parameter descriptions, and also the default parameter values that are preconfigured in the GVP `Application` objects. For information about all the available configuration parameters, see the *Genesys Voice Platform 8.1 Configuration Options Reference*.

Configure the options in Genesys Administrator on the `Provisioning > Environment > Applications > <GVP Component> > Options` tab. For detailed steps to configure option settings, see Procedure: Viewing or modifying GVP configuration parameters, on page 34.

Note: The configuration options for GVP processes are complex and provide a great deal of flexibility. Deploying GVP as described in the *Genesys Voice Platform 8.1 Deployment Guide* provides a fully functional, basic GVP deployment, with the minimum number of customizations required for GVP to operate in your environment. Before performing additional customizations, ensure that you review the configuration options and fully understand their implications.

Configuring SIP Communication and Routing

Task Summary: Configuring SIP Communications and Routing summarizes the steps and parameters to configure the transport and routing mechanisms for SIP messaging within the GVP deployment.

Task Summary: Configuring SIP Communications and Routing

Objective	Related Procedures and Actions
For each Resource Manager, Media Control Platform, Call Control Platform, and CTI Connector `Application` in your deployment, configure the SIP transports for the supported transport protocols.	Configure the `sip.transport.<x>` options (see page 44). Note the following: • For the Resource Manager, specify separate transports for SIP proxy, registrar, and monitoring purposes. • The lowest `<x>` in a set of `sip.transport.<x>` options indicates the preferred default protocol. By default, User Datagram Protocol (UDP) is the preferred protocol for all components (`sip.transport.0`). • To make TCP the preferred protocol, either reorder (by renaming) the respective `sip.transport` parameters, or else remove the `sip.transport.0` UDP parameter so that the `sip.transport.1` TCP parameter is the lowest numerically defined `sip.transport.<x>` and thus, becomes the default.

Task Summary: Configuring SIP Communications and Routing (Continued)

Objective	Related Procedures and Actions
For each Resource Manager, Media Control Platform, Call Control Platform, and CTI Connector `Application` in your deployment, configure the SIP transports for the supported transport protocols. (continued)	• For secure SIP (SIPS) communications, specify a transport for TLS. For more information, see "Enabling Secure Communication" on page 46. • For setting IP `DiffServ` (ToS) field in the outgoing SIP messages, specify the ToS parameter: `sip.transport.<x>.tos` **Note:** If you change the default preferred protocol for the Call Control Platform, you must complete additional steps on the CCXML application side, to ensure that the `Request-URI` specifies the correct protocol. For more information, see page 215.
Configure the route set and routing table for outbound calls.	• For each Media Control Platform and Call Control Platform `Application` in your deployment, configure the `sip.routeset` or `sip.securerouteset` option. • For each Resource Manager, Media Control Platform, and Call Control Platform `Application` in your deployment, configure the required `sip.route.dest.<n>` entries.
Verify settings that determine behavior in relation to the SIP stack.	Review and, if necessary, modify the options that control such parameters as number of threads, size of the Maximum Transmission Unit (MTU) of the network interfaces, and number of connections: • For the Resource Manager, the relevant options are in the `proxy` configuration section. • For the Media Control Platform and Call Control Platform, the relevant options are in the `sip` configuration section.

Table 4 provides information about important SIP communications and routing options. It includes parameter descriptions as well as the default parameter values that are preconfigured in various configuration sections in the Resource Manager, Media Control Platform, and Call Control Platform `Application` objects.

The default values for the `sip.transport.<n>` parameter in the Resource Manager, Media Control Platform, Call Control Platform, and CTI Connector `Applications` are summarized in Table 4.

For information about all the available configuration options, see the *Genesys Voice Platform 8.1 Configuration Options Reference.*

Table 4: Default SIP Transports

Component Application	Section.Option Name	Default Value
Resource Manager	proxy.sip.transport.0	`transport0 udp:any:5060`
	proxy.sip.transport.1	`transport1 tcp:any:5060`
	proxy.sip.transport.2	`transport2 tls:any:5061 cert=$InstallationRoot$\config\x509_certificate.pem key=$InstallationRoot$\config\x509_private_key.pem`
	Note: If host names are used in your deployment, the `proxy.sip.transport.[x]` must be configured with that hostname—for example, `proxy.sip.transport.0` = `transport0 udp:myhostname.com:5060`.	
	registrar.sip.transport.0	`transport0 udp:any:5062`
	registrar.sip.transport.1	`transport1 tcp:any:5062`
	registrar.sip.transport.2	`transport2 tls:any:5063 cert=$InstallationRoot$\config\x509_certificate.pem key=$InstallationRoot$\config\x509_private_key.pem`
	monitor.sip.transport.0	`transport0 udp:any:5064`
	monitor.sip.transport.1	`transport1 tcp:any:5064`
	monitor.sip.transport.2	`transport2 tls:any:5065 cert=$InstallationRoot$\config\x509_certificate.pem key=$InstallationRoot$\config\x509_private_key.pem`
	subscription.sip.transport.0	`transport0 udp:any:5066`
	subscription.sip.transport.1	`transport1 tcp:any:5066`
	subscription.sip.transport.2	`transport2 tls:any:5067 cert=$InstallationRoot$\config\x509_certificate.pem key=$InstallationRoot$\config\x509_private_key.pem`

Table 4: Default SIP Transports (Continued)

Component Application	Section.Option Name	Default Value
Media Control Platform **Note:** If all `sip.transport.x` values are empty, UDP, TCP, and TLS transports will all be enabled, and will listen from ports 5070, 5070, and 5071, respectively, on any network interface.	sip.transport.0	`transport0 udp:any:5070`
	sip.transport.1	`transport1 tcp:any:5070`
	sip.transport.2	`transport2 tls:any:5071 cert=$InstallationRoot$\config\x509_certificate.pem key=$InstallationRoot$\config\x509_private_key.pem`
MRCP V2 Client **Note:** If all sip.transport.x values are empty, UDP, TCP, and TLS transports will all be enabled, and will send MRCP session request from ports 7080, 7080, and 7081, respectively, on any network interface.	sip.transport.0	`transport0 udp:any:7080`
	sip.transport.1	`transport0 udp:any:7080`
	sip.transport.2	`transport2 tls:any:7081` `type=TLSv1`
Call Control Platform **Note:** If all `sip.transport.x` values are empty, UDP, TCP, and TLS transports will all be enabled, and will listen from ports 5068, 5068, and 5069, respectively, on any network interface.	sip.transport.0	`transport0 udp:any:5068`
	sip.transport.1	`transport1 tcp:any:5068`
	sip.transport.2	`transport2 tls:any:5069 cert=$InstallationRoot$/config/x509_certificate.pem key=$InstallationRoot$/config/x509_private_key.pem`
	`mediacontroller.sipsproxy`	The address of SIP Secure Proxy for outbound SIP requests. Specify in this format: `10.10.30.205:5071` Default: `$LocalIP$:5061`
	`mediacontroller.bridge_sips_server`	Address of sip secure bridge server Default: `$LocalIP$:5061`
	`mediacontroller.sipsecure`	If this flag is set to `true`, all the outbound sip requests would be in SIP Secure protocol. Note that the `hints` attribute of the CCXML elements that initiates an outbound request can overwrite this configuration. Possible Values: 1 - `true` 0 - `false` (default)

Table 4: Default SIP Transports (Continued)

Component Application	Section.Option Name	Default Value
CTI Connector **Note:** If all `sip.transport.x` values are empty, UDP, TCP, and TLS transports will all be enabled, and will listen from ports 5080, 5080, and 5081 respectively, on any network interface.	`sip.transport.0`	`transport0 udp:any:5080`
	`sip.transport.1`	`transport1 tcp:any:5080`
	`sip.transport.2`	`transport2 tls:any:5081` `cert=$InstallationRoot$/config/x509_certificate.pem key=$InstallationRoot$/config/x509_private_key.pem`
PSTN Connector	`GatewayManager.UserAgentAddr`	
	`GatewayManager.UserAgentPort`	

Enabling Secure Communication

Task Summary: Enabling SIPS, HTTPS, and SRTP in GVP summarizes the steps and parameters to set up your GVP deployment to use Secure Socket Layer (SSL) technology for secure SIP (SIPS), secure HTTP (HTTPS), and secure RTP (SRTP) communication.

Note: Although, the GVP components support SIPS, the Genesys SIP Server does not. Before you enable SIPS in your GVP deployment, contact your Genesys Sales Representative for more information.

Task Summary: Enabling SIPS, HTTPS, and SRTP in GVP

Objective	Related Procedures and Actions
Set up GVP to use SIPS for call control messaging.	1. If required, generate and deploy the SSL private key and certificate (see Procedure: Creating an SSL private key and certificate). 2. On the Resource Manager, Media Control Platform, Call Control Platform, and CTI Connector `Applications`, specify the SIP transport for TLS, including the additional parameters for the certificate and key (see information about the `sip.transport.<x>` option on pages 42, and the default values in Table 4 on page 44). 3. On the Media Control Platform and Call Control Platform Applications, specify secure routing for outbound calls (see information about the `sip.securerouteset` option on page 44 and the `sip.route.dest.<n>` option on page 44). 4. On the Call Control Platform Applications, specify secure SIP proxy to generate dialogs by using TLS calls (see information about the `mediacontroller.sipsproxy` option on page 45. 5. Modify the CCXML applications, as required, to ensure that the `Request-URI` specifies TLS as the transport protocol.
Set up the Fetching Module to use HTTPS.	1. Generate and deploy the SSL private key and certificate. For information about creating a self-signed certificate, see Procedure: Creating an SSL key and self-signed certificate for use with IIS, on page 51. 2. On the Media Control Platform or Call Control Platform `Applications` in your deployment, configure the `https_proxy`, and the `ssl_*` parameters in the `fm` section. 3. If Squid is deployed, modify the Squid configuration file, if necessary, to configure “safe” and SSL ports, and to enforce SSL (see Procedure: Modifying the Squid Configuration, on page 272). Also, if the HTTPS connection is to tunnel through Squid or another HTTP proxy, configure the `https_proxy` parameter in the `fm` section.
Verify that timeout settings are suitable for your deployment.	Given the additional processing time and lags associated with SSL encryption/decryption and handshakes, reconsider the following settings in particular: • For the Fetching Module, `iproxy.connect_timeout` (default is 5 seconds). • For the Media Control Platform, timeouts in the `sessmgr` and `sip` sections.

Task Summary: Enabling SIPS, HTTPS, and SRTP in GVP (Continued)

Objective	Related Procedures and Actions

Enable SRTP for the media channel between the Media Control Platform and the remote endpoint.	On the Media Control Platform Application:
	1. Specify the required mode (`accept-only`, `offer`, or `offer_strict`) in the `mpc.srtp.mode` parameter. By default, SRTP is not enabled.
	2. If necessary, modify the default values for the encryption and authentication algorithms (the cryptographic suites), and for the session parameters that the Media Control Platform will advertise in the SDP `crypto` attribute:
	`mpc.srtp.cryptomethods`
	`mpc.sessionparams`
	`mpc.sessionparamsoffer`

Enable SRTP for the media channel between the MRCPv2 server and the Media Control Platform.	On the MRCPv2 `Application` that represents the third-party MRCP server for ASR or TTS, verify and, if required, modify settings for the following options:
	`provision.vrm.client.TlsCertificateKey`
	`provision.vrm.client.TlsPrivateKey`
	`provision.vrm.client.TlsPassword`

Create security certificates to enable the Supplementary Services Gateway to interact with SIP Server over secure ports.	On Windows and Linux install and configure security certificates to enable interactions over TLS. See Procedure: Creating Security Certificates for TLS Interactions, on page 53.
If necessary, set up the Reporting Server to use HTTPS.	See "Enabling HTTPS for Reporting" on page 287.
Configure Genesys Administrator to use HTTPS to access Reporting Server web services, for the GVP reports that are displayed in the `Monitoring > Voice Platform` view.	• In Genesys Administrator, on the `Provisioning > Environment > Applications > default > Options tab`, set the value of `rptui.enablehttps` to `true`. • Ensure that the web server for Reporting Server is configured to enable HTTPS.

Note: Observe standard security practices to ensure that you protect the security of SSL private keys, SSL certificates, and configured user names and passwords—for example, ensure that they are stored on secure hosts, and do not create them over a network.

Configuring MCP, MRCPv2, CCP, CTIC, and RM for Secure SIP Transport

This section offers configuration examples that enable secure communication.

MCP

Inbound Inbound supports both `sips:` and `transport=TLS` schemas. Examples:

```
INVITE
sips:dialog@000.00.000.00:5071;voicexml=http://000.00.000.00/testca
se/gvp8/hellotransfer.vxml;aai=N/A SIP/2.0

INVITE
sip:dialog@000.00.000.00;transport=TLS;voicexml=http://000.00.000.0
0/testcase/gvp8/hellotransfer.vxml;aai=N/A SIP/2.0
```

Transfer Transfer can make calls with both and “transport=tls”. Examples:

```
<transfer name="newcall" dest="sips:4162245081@000.00.000.00:5071"
bridge=
sent: INVITE sips:4162245081@000.00.000.00:5071 SIP/2.0
sent: REFER sips:Genesys@000.00.000.00:5071 SIP/2.0

<transfer name="newcall"
dest="sip:4162245081@000.00.000.00:5071;transport=tls"bridge=
sent: INVITE sip:4162245081@000.00.000.00:5071;transport=tls SIP/2.0
sent: REFER sip:Genesys@000.00.000.00:5071;transport=tls SIP/2.0
```

MRCPv2

Nuance Speech Server 6 supports only the `sips:` schema. Example:
`sips:mresources@[MRCP server IP]:[port]`

Nuance Speech Server 5 supports the `transport=tls` schema. Example:
`sip:mresources@[MRCP server IP]:[port] ; transport=TLS`

Call Control Platform (CCP)

Inbound CCP supports sip secure schema `sips:` and `transport=TLS`

The example below applies to both:

```
INVITE
sips:ccxml@000.00.000.00:5069;ccxml=http://000.00.000.00/testcase/g
vp8/dialog.ccxml;

INVITE
sip:ccxml@000.00.000.00:5069;transport=TLS;ccxml=http://000.00.000.
00/testcase/gvp8/dialog.ccxml;
```

CTI Connector

When CTI Connector receives an incoming call on a secure channel, it will use only a secure channel to make the outbound call.

For example:

```
INVITE sips:1234@000.00.000.00:5080 SIP/2.0
INVITE sip:1234@000.00.000.00:5080;transport=TLS SIP/2.0
```

Note: We have not tested CTI Connector behavior by modifying the mediacontroller.sipsecure parameter. I will check with QA about this and revert back to you.

Resource Manager

When the Resource Manager receives an incoming call on a secure channel, it will use only a secure channel to make the outbound call.

Procedures that Support Enabling Secure Communication

The following procedures support the tasks outlined in Task Summary: Enabling SIPS, HTTPS, and SRTP in GVP:

- Procedure: Creating an SSL private key and certificate
- Procedure: Creating an SSL key and self-signed certificate for use with IIS, on page 51
- Procedure: Creating Security Certificates for TLS Interactions, on page 53
- Procedure: Configuring the Fetching Module for HTTPS, on page 55

Procedure:
Creating an SSL private key and certificate

Purpose: To illustrate how to create and deploy the private key and SSL certificate that are used for SIPS and HTTPS authentication.

Perform this procedure for each Resource Manager, Media Control Platform, and Call Control Platform in your deployment.

Prerequisites

- The OpenSSL Toolkit (openssl) or other SSL tool is available.

  You can download the OpenSSL Toolkit for Windows from Shining Light Productions at the following URL:

  `http://www.shininglightpro.com/products/Win32OpenSSL.html`

  For more information about OpenSSL, see `http://www.openssl.org/`.

Start of procedure

1. Generate the private key:
   - For a password-protected key, execute the following command:
     `openssl genrsa -aes128 -out x509_private_key.pem 2048`
   - For a non-password-protected key, execute the following command:
     `openssl genrsa -out x509_private_key.pem 2048`
2. Generate the certificate.

   The following example of the required command creates a certificate with file name `x509_certificate.pem,` which expires in 1095 days:

   ```
   openssl req -new -x509 -key x509_private_key.pem -out
   x509_certificate.pem -days 1095
   ```

   For information about additional supported parameters, see the *openssl Manual* page on the OpenSSL web site (`http://www.openssl.org/`).
3. Install the certificate and key.

   The default GVP configuration assumes that the file names and paths are as follows:
   - For the certificate:
     `$InstallationRoot$\config\x509_certificate.pem`
   - For the private key:
     `$InstallationRoot$\config\x509_private_key.pem`

End of procedure

Next Steps

- If required, modify the `sip.transport.<x>` configuration option for TLS to update the parameters for the certificate path, key path, and password (if applicable).

Procedure: Creating an SSL key and self-signed certificate for use with IIS

Purpose: To illustrate how to use the OpenSSL Toolkit to create a private key and self-signed SSL certificate request, to enable HTTPS connections to the IIS web server for Fetching Module communications.

Prerequisites

- The OpenSSL Toolkit (openssl) has been installed, with default settings.

  You can download the OpenSSL Toolkit for Windows from Shining Light Productions at the following URL:

`http://www.shininglightpro.com/products/Win32OpenSSL.html`

For more information about OpenSSL, see `http://www.openssl.org/`.

Start of procedure

1. Set up the openssl directories and files:
   a. (Optional, but recommended) Add `C:\OpenSSL\bin` to your system path (`Control Panel > System > Advanced > Environment Variables > System Variables`).
   b. Create a working directory—for example, `C:\ssl`.
   c. Create the directory structure and files required by openssl:
      - Directories: `keys, requests,` and `certs`
      - Files: `database.txt` and `serial.txt`—these are empty (zero-byte) text files

      To create the directories and files manually, execute the following commands at the `C:\ssl>` UNIX prompt:

      ```
      md keys
      md requests
      md certs
      copy con database.txt
      ^Z
      copy con serial.txt
      01
      ^Z
      ```
2. Set up a Certificate Authority (CA):
   a. At the `C:\ssl>` prompt, execute the following command to create a 1024-bit private key:

      ```
      openssl genrsa -des3 -out keys/ca.key 1024
      ```
   b. At the `C:\ssl>` prompt, execute the following command to create the CA certificate:

      ```
      openssl req -config openssl.conf -new -x509 -days 1001 -key
      keys/ca.key -out certs/ca.cer
      ```

      The following certificate is created:

      `c:\ssl\certs\ca.cer`
3. Create an IIS Certificate Request (`certreq.txt`).

   For more information, see the Microsoft Knowledge Base article number 228821, which is available from Microsoft Technical Support (`http://support.microsoft.com`).
4. Sign the Certificate Request.
   a. Copy the `certreq.txt` file into `C:\ssl\requests`.

b. At the `C:\ssl>` prompt, execute the following command to sign the request:

```
C:\ssl>openssl ca -policy policy_anything -config openssl.conf
-cert certs/ca.cer -in requests/certreq.txt -keyfile keys/ca.key
-days 360 -out certs/iis.cer
```

5. Install the new certificate under IIS.

   For more information, see the Microsoft Knowledge Base article number 228836, which is available from Microsoft Technical Support (`http://support.microsoft.com`).

   The secure web server is now accessible from any web browser, using SSL.

End of procedure

Next Steps

- Create security certificates for TLS interactions (if required). See Procedure: Creating Security Certificates for TLS Interactions.

Procedure: Creating Security Certificates for TLS Interactions

Purpose: To create security certificates to enable the Supplementary Services Gateway to interact with SIP Server by using TLS through secure ports.

Summary

The Security Pack on Linux provides the components, such as shared libraries and an example of a Certification Authority (CA), that are used to generate certificates and to deploy them on the installed GVP components.

Start of procedure

On Windows

1. Import the `ca_cert.pem` file to the `Trusted Root Certificate Authorities` folder.
2. Import the <serial_#>_<host_name>_cert.pfx file to the `Personal` folder of the certificate service.

On Linux

3. Install the Security Pack for Linux.
4. Configure the environment variable that corresponds to Linux and specify the path to Security Pack libraries. For example, export `LD_LIBRARY_PATH=/home/svetar/Security/Linux/SecurityPack_810`

   The certificate is generated and added to the application.

On Windows and Linux

5. In Genesys Administrator, assign the certificate to the host machine and the host application.

6. Create a secure port for the SIP Server `Application`.

   For a detailed description of the configuration that is required to assign the certificate to the host machine, application, and port, see Chapter 18 in the *Genesys 8.1 Security Deployment Guide.*

7. Assign the certificate to the secure port.

   SIP Server will listen on both the default and the secure port.

8. In Genesys Administrator, in the Supplementary Services Gateway `Application`, create a connection to SIP Server.

9. In the `Connection Info` dialog box, edit the properties to configure the ports to which the Supplementary Services Gateway will connect to SIP Server. See Figure 3 on page 54.

   For a detailed description of how to install and configure security certificates, see Chapters 16-18 in the *Genesys 8.1 Security Deployment Guide*

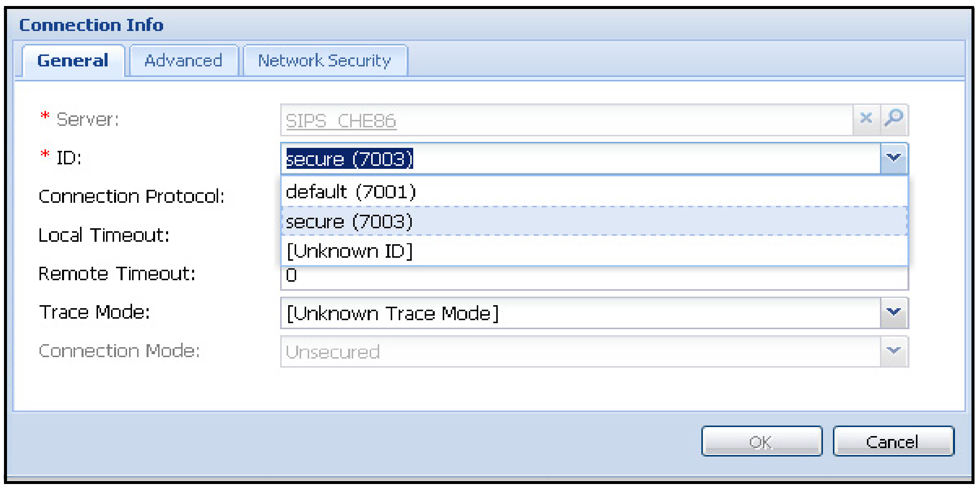


Figure 3: Connection Info Dialog Box in Genesys Administrator

End of procedure

Next Steps

- Configure the Fetching Module (`pwproxy`) to access files over HTTPS (see Procedure: Configuring the Fetching Module for HTTPS).

Note: The following procedure is applicable for pre Genesys Voice Platform 8.1.2 only deployments.

Procedure:
Configuring the Fetching Module for HTTPS

Purpose: To modify the default Fetching Module configuration to enable secure HTTPS communications.

Perform this procedure on each Fetching Module in your deployment.

Prerequisites

- The SSL certificate and key have been created and installed under IIS.

  For information about creating a self-signed certificate, see Procedure: Creating an SSL key and self-signed certificate for use with IIS, on page 51.

Start of procedure

1. Create the PEM certificate file:
   a. Using a text editor, create a new file, `proxy_client.pem`. You can store the file under any directory on the Fetching Module server.
   b. Open the `ca.key` file created by openssl (see Step 2 on page 52).
   c. Copy all the lines from `ca.key` into the new `proxy_client.pem` file.
   d. Press `Enter` to create one blank line at the end of the text.
   e. Open the `iis.cer` file created when you signed the certificate request (see Step 4 on page 52).
   f. Copy all the lines from `iis.cer` starting from `-----BEGIN CERTIFICATE----` and ending with `-----END CERTIFICATE-----`, inclusive, into the bottom of the `proxy_client.pem` file.

   The final file will look similar to Figure 4.

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,70D8F72D9BB079C3

aFnOkM5agLiG7gvcEBjZ+GIAKfFsCQKuq3cYBkng/Zlp5vgapDqx6JUycPcBs7A/
Y35h4E4HDJv40gJ3xqLc4ENrhFH4Vezc4hFDb5SfQteVQP1nklxYBE5vUY+55xwv
UCcbrpD3PjqVakWPwdz7HtA7prH/4izUytE99yEE3C5pf3QpnUvOps9OH+WN3x9L
IAWun2t2bojDjwofIREx4C0iWH/3PHi9gqpbZeRXvgwvEfw8dpKwh/oV5mCexcWt
YTJ/6Nf5fFCA2NxoaboZXIBa83ISOuceZXAb5yEiXfpe4k4wPweLHc7kzhwLiwJL
6JUnG7yjAcVxeN6gDk+oxGRkPoz7xpOVwTWRk/uCSF0umai30Mrv8CuOdya0hB/2
jBD1PeH8+1yfngH5RcU33vZJIMJtHVBiTA330YQLDqke2xvJf4uBxdawU7BSmYpT
Bo35suRc4wARf7TF8Gvxl5epFDCSx32i81rkbZhv9GlFfajiiBV3VRTMlN+ydSXb
QnLU+Oe5ln1BRbY7OUXOHLuGJRMdY1j/vkJYPbCeGhOa4S4wPQT1tPYcBpYdVhcH
DFZn55Glzlf0d4BUXeFl1lKu5FK9P0B4ozLtXwMZtaUXQ44vLjPJTWLMLpNY3AKS
zmb2boDqn5btipuxwmqXYFLfIZl6h32sLLuZex3gv9lbURsD8Zr+HgqVNzXwJTW9
kDEndj5Bd+pMUe3i/9gr0nPAVMFkFsUuHEZPNNNl2AZsWwOkPsd9o7YEHVJGovS6
AV3D6KPhOHFhg8AHmrEHcJLkN77JTSlbUJdAO+t/KNyYRs3TLwEexg==
-----END RSA PRIVATE KEY-----

-----BEGIN CERTIFICATE-----
MIICYjCCAcugAwIBAgIBATANBgkqhkiG9w0BAQQFADCBijELMAkGA1UEBhMCQ0Ex
EDAOBgNVBAgTB09udGFyaW8xEDAOBgNVBAcTB01hcmtoYW0xEDAOBgNVBAoTB0dl
bmVzeXMxCjAIBgNVBAsTATExFzAVBgNVBAMTDjEzOC4xMjAuODQuMTQ0MSAwHgYJ
KoZIhvcNAQkBFhFxYUB2b2ljZWdlbmllLmNvbTAeFw0wODA0MjUxNDUxNTBaFw0w
OTA0MjAxNDUxNTBaMGMxCzAJBgNVBAYTAkNBMQswCQYDVQQIEwJPTjEQMA4GA1UE
BxMHTWFya2hhbTEQMA4GA1UEChMHR2VuZXN5czEKMAgGA1UECxMBMTEXMBUGA1UE
AxMOMTM4LjEyMC44NC4xNDQwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMQr
f+mPgVE8Aemgbg90UocmyEJ0lh2yC3KGjC1UgLPF6TQaJ2vLIniCBSUUINgDrujo
8PKMJBXOpL4FUIqMzcX0foVCx4zXK7bPwO8mibm1DB1DDJ7dy+2n7vLRV1PM6R/r
G4l2oYeRC6tYLEg3818WJnGR49yrWPMGfHvXeHTzAgMBAAEwDQYJKoZIhvcNAQEE
BQADgYEAaVG00qOsU7L3riGomIWCnp30rtcv4lnVmUDG9BvhBWNF65EfXiSwEjqi
GIAnss9CZYw1odqo+hZsNLttwERlRn973K4GGmywQFnErei5hZeonMFm0BZjkvch
ynbPVTr/OOOt3+cKhW1Ef1osh5fFxlWlhNrwv11mpkGOOZ8pVME=
-----END CERTIFICATE-----
```


Figure 4: Sample PEM Certificate File

2. Save the file.
3. In Genesys Administrator, on the `Provisioning > Environment > Applications > <Fetching Module> > Options tab`, modify the Fetching Module configuration:
   a. Verify that the value of the `fm.https_proxy` parameter is empty (disabled—encrypted pages will not be cached).
   b. Configure the following options in the `iproxy` configuration section:
      - `ssl_key_passwd` = `<Your private key passphrase>`

- `ssl_key = <Local path to your proxy_client.pem file>`
- `ssl_cipher_list = TLSv1`

For information about other SSL-related configuration options for the Fetching Module, see the *Genesys Voice Platform 8.1 Configuration Options Reference* (options beginning with `ssl_` in the `iproxy` section).

c. Click `Save` or `Apply` to save the configuration changes.

4. Restart the Fetching Module application.

End of procedure

Next Steps

- If required, modify the Squid configuration file to identify the "safe" ports for HTTP and SSL requests, to identify the ports to be used for SSL connections, and to deny access to non-SSL connections.

  For more information, see Procedure: Modifying the Squid Configuration, on page 272

Enabling IPv6 Communication

Table 5 contains a list of the configuration options that you should be aware of if you intend to support Internet Protocol version 6 (IPv6) communication in your environment. These options can be used to customize your GVP configuration.

Notes: Although, the GVP components supports IPv6, the Genesys SIP Server and Cisco T-Servers do not. Before you enable IPv6 in your GVP deployment, contact your Genesys Sales Representative for more information.

GVP components support non-linked-local IPv6 addresses only. When using IPv6, do not use linked-local addresses.

Tip: When using IPv4-mapped IPv6 addresses, be aware that not all operating platforms function in the same way. While IPv4-mapped IPv6 addresses might function on a Windows platform, they might not function on a Linux platform.

Table 5: Configuration Options that Support IPv6

Component	Configuration option	Description
Resource Manager	• [proxy] sip.preferred_ipversion • [registrar] sip.preferred_ipversion • [monitor] sip.preferred_ipversion • [subscription] sip.preferred_ipversion	Use this option to specify the preferred IP version when a destination address resolves into multiple IP addresses that use different IP versions. The first IP address processed that matches the preferred IP version is used. However, if a sip.transport is not defined for the preferred version, a defined version that matches one of the processed IP addresses is used. Valid values are ipv4 and ipv6.
	• [proxy] sip.transport.<n> • [registrar] sip.transport.<n> • [monitor] sip.transport.<n> • [subscription] sip.transport.<n>	Use these options to define the transport layer for the SIP stack and the network interfaces that are used to process SIP requests.
	• [proxy] sip.transport.localaddress_ipv6 • [registrar] sip.transport.localaddress_ipv6 • [monitor] sip.transport.localaddress_ipv6 • [subscription] sip.transport.localaddress_ipv6	Use these options to specify the sent-by field of the Via header and the hostport part of the Contact header in an outgoing SIP message if an IPv6 transport is used The value must be a host name or domain name. (Configuration of this option is not required; it can work with the default value.)

Table 5: Configuration Options that Support IPv6 (Continued)

Component	Configuration option	Description
Resource Manager (continued)	• `[proxy] sip.route.default.udp.ipv6` • `[registrar] sip.route.default.udp.ipv6` • `[monitor] sip.route.default.udp.ipv6` • `[subscription] sip.route.default.udp.ipv6`	Use these options to specify the transport that is defined in the `sip.transport.x` configuration option, where `x` is the value of this option. These options are used when there are no IPv6 UDP routes found.
	• `[proxy] sip.route.default.tcp.ipv6` • `[registrar] sip.route.default.tcp.ipv6` • `[monitor] sip.route.default.tcp.ipv6` • `[subscription] sip.route.default.tcp.ipv6`	Use these options to specify the transport that is defined in the `sip.transport.x` configuration option, where `x` is the value of these options. These options are used when there are no IPv6 TCP routes found.
	• `[proxy] sip.route.default.tls.ipv6` • `[registrar] sip.route.default.tls.ipv6` • `[monitor] sip.route.default.tls.ipv6` • `[subscription] sip.route.default.tls.ipv6`	Use these options to specify the transport that is defined in the `sip.transport.x` configuration option, where `x` is the value of this option. These options are used when there are no IPv6 TLS routes found.
	Note: For a complete description of these Resource Manager options and their default values, see Table 13 on page 89.	
Media Control Platform	`[mpc] preferredipinterface`	Use this option to specify the preferred IP interface to use (IPv4 or IPv6) when SDP is being negotiated. This option value sets the root connection attribute in SDP answers and sets the connection attribute in SDP offers.
	• `[sip]` route.default.udp.ipv6 • `[sip]` route.default.tcp.ipv6 • `[sip]` route.default.tls.ipv6	Use these options to specify the default IPv6 route for UDP, TCP, or TLS. The number denotes the transport that is defined in the `sip.transport.x` configuration option.

Table 5: Configuration Options that Support IPv6 (Continued)

Component	Configuration option	Description
Media Control Platform (continued)	`[sip] transport.x`	Use this option to define the transport layer for the SIP stack and the network interfaces that are used to process SIP requests.
	`[sip] preferred_ipversion`	Use this option to specify the preferred IP version when a destination address resolves into multiple IP addresses that use different IP versions. The first IP address processed that matches the preferred IP version is used. However, if a `sip.transport` is not defined for the preferred version, a defined version that matches one of the processed IP addresses is used. Valid values are `ipv4` and `ipv6`.
	`[sip] transport.localaddress_ipv6`	Use this option to specify the `sent-by` field of the `Via` header and the `hostport` part of the `Contact` header in an outgoing SIP message if an IPv6 transport is used The value must be a host name or domain name. (Configuration of this option is not required; it can work with the default value.)
	Note: For a complete description of these Media Control Platform options and their default values, see Table 23 on page 160.	
Call Control Platform	`[ccxmli] basichttp.recv.host.ipv6`	Use this option to specify the IPv6 address or host name on which the basic HTTP event I/O processor will listen for HTTP requests on IPv6 network interface.
	`[ccxmli]` createsession.recv.host.ipv6	Use this option to specify the IPv6 address or host name on which the session creation event I/O processor will listen for HTTP requests on IPv6 network interface.
	`[ccxmli] createsession.recv.accessuri`	Use this option to specify the preferred IP version that will be used in the create session access URI `session.ioprocessors["createsession"]`.
	`[mediacontroller] sdp.localhost.ipv6`	Use this option to specify the `host` part of the local host IPv6 address that is used in SDP.
	`[sip] basichttp.recv.accessuri`	Use this option to specify the preferred IP version that will be used in basic HTTP access URI `session.ioprocessors["basichttp"]`.

Table 5: Configuration Options that Support IPv6 (Continued)

Component	Configuration option	Description
Call Control Platform (continued)	• `[sip]` route.default. udp.ipv6 • `[sip]` route.default. tcp.ipv6 • `[sip]` route.default. tls.ipv6	Use these options to specify the default IPv6 route for UDP, TCP, or TLS. The number denotes the transport that is defined in the `sip.transport.x` configuration option.
	`[sip] transport.x`	Use this option to define the transport layer for the SIP stack and the network interfaces that are used to process SIP requests.
	`[sip] preferred_ipversion`	Use this option to specify the preferred IP version when a destination address resolves into multiple IP addresses that use different IP versions. The first IP address processed that matches the preferred IP version is used. However, if a `sip.transport` is not defined for the preferred version, a defined version that matches one of the processed IP addresses is used. Valid values are `ipv4` and `ipv6`.
	`[sip] transport.localaddress_ipv6`	Use this option to specify that the `sent-by` field of the `Via` header and the `hostport` part of the `Contact` header in the outgoing SIP message will be set to this value if a IPv6 transport is used. (Configuration of this option is not required; it can work with the default value.)
	Note: For a complete description of these Call Control Platform options and their default values, see Table 26 on page 217	
CTI Connector	`[sip] transport.x`	Use this option to define the transport layer for the SIP stack and the network interfaces that are used to process SIP requests.
	`[sip] preferred_ipversion`	Use this option to specify the preferred IP version when a destination address resolves into multiple IP addresses that use different IP versions. The first IP address processed that matches the preferred IP version is used. However, if a `sip.transport` is not defined for the preferred version, a defined version that matches one of the processed IP addresses is used. Valid values are `ipv4` and `ipv6`.

Table 5: Configuration Options that Support IPv6 (Continued)

Component	Configuration option	Description
CTI Connector (continued)	• `[sip]` route.default.udp.ipv6 • `[sip]` route.default.tcp.ipv6 • `[sip]` route.default.tls.ipv6	Use these options to specify the default IPv6 route for UDP, TCP, or TLS. The number denotes the transport that is defined in the `sip.transport.x` configuration option.
	`[sip] transport.localaddress_ipv6`	Use this option to specify that the `sent-by` field of the `Via` header and the `hostport` part of the `Contact` header in the outgoing SIP message will be set to this value if a IPv6 transport is used. (Configuration of this option is not required; it can work with the default value.)
	Note: For a complete description of these CTI Connector options and their default values, see Table 27 on page 234.	
Supplementary Services Gateway	`[Common] enable-ipv6`	Use this option to enable an IPv6 communication between the SSG and SIP Server.
	Note: For a complete description of this Supplementary Services Gateway option and its default value, see Table 30 on page 251.	

Enabling Conference Services

Task Summary: Configuring Conferencing summarizes the steps and parameters to configure the GVP deployment to provide conference service.

Notes: You can set values for options such as conference reserve, maximums for number of conferences and participants, and conference capabilities can be set at the level of the resource group, the resource, and the IVR Profile, in order of override priority. These parameters are significant in determining how the Resource Manager handles a particular request for conference service.

Genesys recommends that, before you modify options, you carefully review the descriptions for all contexts (resource group, individual resource, and IVR Profile). For more information about the configuration options, see the remaining chapters in the Provisioning section of this guide, and the *Genesys Voice Platform 8.1 Configuration Options Reference*

Task Summary: Configuring Conferencing

Objective	Related Procedures and Actions
Assign a conference resource to a logical resource group that provides conference service. **Note:** For MCP only.	See "Configuring Logical Resource Groups" on page 93: 1. Create or modify a logical resource group for the Resource Manager, where the value of the `<logical resource group>.service-types` option includes `conference`. 2. Set the general conference maximums for the resource group (see the description of the `confmaxsize` and `confmaxcount` options). 3. If the resource has not already been added to the Resource Manager connections, add it. For more information, see the chapter about postintstallation activities in the *Genesys Voice Platform 8.1 Deployment Guide*.
Create an IVR Profile for conference service.	Set the following required parameter: • `gvp.service-prerequisite.conference-id` Also consider the following IVR Profile options, which determine whether and how conference service will be provided: • `gvp.general.application-confmaxsize` • `gvp.general.service-type` • `gvp.policy.conference-allowed` • `gvp.policy.conference-capability-requirements` • `gvp.policy.conference-usage-limit` and `conference-usage-limit-per-session` For more information, see "IVR Profile Configuration Options" on page 113.
Verify that conference-related settings on the Media Control Platform and Call Control Platform are suitable.	• For the Media Control Platform, review the options in the `conference` section. • For the Call Control Platform, verify the settings for the `Default Conference` device profile, and the options in the `mediacontroller` configuration sections.
(Optional) Customize the SIP response codes and Resource Manager behavior in the event of an error.	On the Resource Manager, customize the value of the `rm.conference-sip-error-respcode` option. For more information, see Table 100 on page 470.

Configuring Reporting

This section describes important parameters for GVP Reporting, which you configure in the `ems` section of the Resource Manager, Media Control Platform, Call Control Platform, Fetching Module, CTI Connector, UCM Connector, MRCP Proxy, Supplementary Services Gateway, and PSTN Connector `Application` objects.

For general information about Reporting in a GVP deployment, see Chapter 3, "How GVP Works" in the *Genesys Voice Platform 8.1 Deployment Guide*.

For information about additional configuration options, see the *Genesys Voice Platform 8.1 Configuration Options Reference.*

Except where otherwise indicated, all changes to `ems` configuration options take effect after you restart the component `Application`.

Table 6 summarizes the default values for the various logs and metrics filters for the component `Application` objects.

Note: Genesys recommends that you do not modify the log filter settings.

Table 6: Default Log and Metrics Filters

Component	Option Name (in ems Section)*					
	logconfig. DATAC	**logconfig. MFSINK**	**logconfig. TRAPSINK**	**metrics-config. DATAC**	**metrics-config. MFSINK**	**metricsconfig. TRAPSINK**
MCP	`0-2,4|*|*`	`*|*|*`	`0-4|*|*`	`*`	`0-16,18-41,43,52-56,72-74,76-81,127-129,130,132-141`	Not Applicable
CCP	`0-2|*|*`	`0-3,5|*|*`	`*|*|*`	`*`	`1000-1001,1003-1005,1007-1016,1019-1021,1024,1027-1036,1039-1045,1048-1050,1052-1054,1056,1058-1062`	`*`
CTIC	Not Applicable	`*|*|*`	`*|*|*`	Not Applicable	Not Applicable	Not Applicable
SSG	Not Applicable	`*|*|*`	`*|*|*`	Not Applicable	Not Applicable	Not Applicable
PSTNC	Not Applicable	`*|*|*`	`*|*|*`	Not Applicable	Not Applicable	Not Applicable

Table 6: Default Log and Metrics Filters (Continued)

Com-ponent	Option Name (in ems Section)*					
	logconfig.DATAC	logconfig.MFSINK	logconfig.TRAPSINK	metrics-config.DATAC	metrics-config.MFSINK	metricsconfig.TRAPSINK
MRCP Proxy	Not Applicable	*|*|*	*|*|*	Not Applicable	Not Applicable	Not Applicable
RM	Not Applicable	*|*|*	*|*|*	Not Applicable	Not Applicable	Not Applicable

Note: The MCP, CCP, MRCPP, and RM components must have a connection to the Reporting Server application in order to collect metrics, CDR and OR data.

Service Quality Analysis (SQA)

Table 7 describes the parameters required for Service Quality reporting in the Media Control Platform.

Note: Service Quality reports apply to NGi VoiceXML applications, and are found in Genesys Administrator. GVP 8.1.5 and thereafter are NGi-only platforms unless you run MCP 8.1.4 to incorporate support for GVPi applications.

Table 7: Service Quality Advisor Parameters

Option Name	Description	Valid Values and Syntax
SQA Enable Flag	Specifies whether to perform Service Quality Analysis.	• `True` • `False` **Default value:** `True`
Inbound Reject Failure Codes	Specifies which in call reject reason codes that, when encountered, do not mark the call as a failure.	A pipe (|)separated list. **Default value:** `decline`

Table 7: Service Quality Advisor Parameters (Continued)

Option Name	Description	Valid Values and Syntax
Outbound Reject Failure Codes	Specifies which out call reject reason codes that, when encountered, do not mark the call as a failure.	A pipe (|)separated list. **Default value:** `busy|decline|fax|noanswer|hangup`
Call Reject Latency Threshold	Specifies the maximum time, in milliseconds, to determine whether the call reject latency is considered a failure because it falls below the threshold.	Any integer. **Default value:** `3000`
Audio Gap Latency	Specifies the largest audio gap allowed while playing audio to the customer.	Any integer. **Default value:** `2000`
Cumulative Response Latency Threshold	Specifies the maximum threshold, in milliseconds, before playing a prompt after customer interaction.	Any integer. **Default value:** `4000`
Inter Prompt Latency Threshold	Specifies the maximum time, in milliseconds, before playing a prompt after playing a previous prompt when no customer interaction has taken place.	Any integer. **Default value:** `4000`
First Prompt Outbound Latency Threshold	Specifies the maximum threshold, in milliseconds, before playing a prompt on an outbound call.	Any integer. **Default value:** `3000`
First Prompt Inbound Latency Threshold	Specifies the maximum threshold, in milliseconds, before playing a prompt on an inbound call.	Any integer. **Default value:** `3000`
Call Answer Latency Threshold	Specifies the maximum time, in milliseconds, to determine whether the call answer latency is considered a failure because it falls below the threshold.	Any integer. **Default value:** `3000`
SQA Batch Size	Specifies the number of SQA messages to queue before sending them to the Reporting Server.	An integer in the range of 1–5000. **Default value:** `5000`

Configuring Logging

Table 8 describes the most commonly customized options for logging. Table 9 on page 71 summarizes the default values for these options in GVP. The options are in the `log` configuration section of each GVP component `Application`.

Configure the options for each component in Genesys Administrator on the `Provisioning > Environment > Applications > <GVP Application> > Options` tab. For the detailed steps to configure option settings, see Procedure: Viewing or modifying GVP configuration parameters, on page 34.

Changes take effect immediately.

The Application Templates do not expose all the logging parameters that are standard in Genesys applications; therefore, the `Options tab` and its metadata (which is also described in the *Genesys Voice Platform 8.1 Configuration Options Reference*) therefore do not describe all the parameters that determine the logging behavior of GVP applications. For more information about the additional, standard logging options, see the Log Section in the chapter about common configuration options in the *Framework 8.1 Configuration Options Reference Manual.*

Table 8: Selected Configuration Options—log Section

<table>
<tr><th>Option Name</th><th>Description</th><th>Valid Values and Syntax</th></tr>
<tr><td>all</td><td>A comma-separated list of the output destinations to which the `Application` (GVP process) sends all log events.
Setting `log.verbose` to `all` and this parameter (`log.all`) to `network` enables an application to send log events of the `Standard`, `Interaction`, and `Trace` levels to Message Server. With this setting, `Debug`-level log events are not sent to Message Server, and they are not stored in the Log Database.</td><td rowspan="5">• `stdout`—Log events are sent to the Standard output (`stdout`).
• `stderr`—Log events are sent to the Standard error output (`stderr`).
• `network`—Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
• `memory`—Log events are sent to the memory output on the local disk. This is the safest output in terms of application performance.
• `<filename>`—Log events are stored in a file with the specified name. The default path for the file is the working directory of the application.</td></tr>
<tr><td>standard</td><td>A comma-separated list of the output destinations to which the application (GVP process) sends log events of the Standard level.</td></tr>
<tr><td>interaction</td><td>A comma-separated list of the output destinations to which the application (GVP process) sends log events of the `Interaction` level and higher (that is, `Standard` and `Interaction` levels).</td></tr>
<tr><td>trace</td><td>A comma-separated list of the output destinations to which the application (GVP process) sends log events of the `Trace` level and higher (that is, `Standard`, `Interaction`, and `Trace` levels).</td></tr>
<tr><td>debug</td><td>A comma-separated list of the output destinations to which the application (GVP process) sends log events of the `Debug` level and higher (that is, `Standard`, `Interaction`, `Trace`, and `Debug` levels).</td></tr>
<tr><td>Enable 6.x Compatibility Log Output Priority</td><td>Specifies whether the application uses 6.x output logic.</td><td>• `true`—The log of the level specified by Log Output options is sent to the specified output.
• `false`—The log of the level specified by Log Output options and higher levels is sent to the specified output.</td></tr>
</table>

Table 8: Selected Configuration Options—log Section (Continued)

Option Name	Description	Valid Values and Syntax
expire	(Applicable only if log output is configured to be sent to a log file.) Specifies the criteria for determining when log files (segments) expire and are deleted.	• `false`—No expiration. All generated segments are stored. • `<number> file|<number>`—A number in the range of `1`–`100` that specifies the maximum number of log files to store. • `<number> day`—A number in the range of `1`–`100` that specifies the maximum number of days before log files are deleted. (This value is not applicable for Reporting Server.)
Log Message Format	The log record header format that an `Application` uses when writing logs in the log file. Using compressed log record headers improves application performance and reduces the size of the log file. When `message_format=short` (compressed headers): • A header of the log file or the log file segment contains information about the application (such as the application name, application type, host type, and time zone), whereas individual log records within the file or segment do not contain this information. • Log message priority is abbreviated to `Std`, `Int`, `Trc`, or `Dbg` (instead of `Standard`, `Interaction`, `Trace`, or `Debug`, respectively). • The message ID does not contain the prefix GCTI or the application type ID.	• `short`—The `Application` uses compressed headers when writing log records in the log file. • `full`—The `Application` uses complete headers when writing log records in the log file. **Log record examples:** • Full format: `2002-05-07T18:11:38.196 Standard localhost cfg_dbserver GCTI-00-05060 Application started` • Short format: `2002-05-07T18:15:33.952 Std 05060 Application started`

Table 8: Selected Configuration Options—log Section (Continued)

Option Name	Description	Valid Values and Syntax
Log Segmentation	(Applicable only if log output is configured to be sent to a log file.) Specifies the mode of measurement and maximum size for a log file segment. If the current log segment exceeds the size set by this option, the file is closed and a new log file is created.	• `false`—No segmentation is allowed. • `<number> KB|<number>`—The maximum segment size, in kilobytes. The minimum segment size is 100 KB. • `<number> MB`—The maximum segment size, in megabytes. • `<number> hr`—The number of hours for the segment to stay open. The minimum time period is 1 hour. (This value is not applicable for Reporting Server.)
Time Generation for Log Messages.	The system by which an application calculates the log record time when a log file is generated. The time is converted from the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970).	• `<number> local`—The time of log record generation is expressed as a local time, based on the time zone and any seasonal adjustments. Time zone information of the application's host computer is used. • `<number> utc`—The time of log record generation is expressed as Coordinated Universal Time (UTC).

Table 8: Selected Configuration Options—log Section (Continued)

Option Name	Description	Valid Values and Syntax
Time Format for Log Messages	The format in which the log file presents the time at which time the application generated the log record.	• `time`—The time string is the `HH:MM:SS.sss` format (hours, minutes, seconds, milliseconds). • `locale`—The time string is formatted according to the locale of the system. • `ISO8601`—The date in the time string is the ISO 8601 format. Fractional seconds are given in milliseconds. **Example:** `2001-07-24T04:58:10.123`
Verbose Level	Specifies the minimum level of log events that will be generated. In descending order of priority, the log event levels are: • `Standard` • `Interaction` • `Trace` • `Debug`	• `all`—All log events. • `debug`—Log events of all levels (same as `all`). • `trace`—Log events of the Standard, Interaction, and Trace levels. • `interaction`—Log events of the Standard and Interaction levels. • `standard`—Log events of the Standard level only. • `none`—No output will be generated.

Table 9 provides the default values for those options in the `log` configuration section that are commonly customized.

Table 9: Default Values for Selected log Options

Option Name	Default Value							
	RM	MCP	CCP	FM	RS	SSG	CTIC	PSTN
all	`../logs/ResourceMgr`	`../logs/MCP`	`../logs/ccp`	`../logs/fm`	Empty	`../logs/SSG`	`../logs/CTIConnector`	`../logs/PSTNConnector`
debug	`../logs/ResourceMgr`	`../logs/MCP`	`../logs/ccp`	`../logs/fm`	`/logs/rs.log`	`../logs/SSG`	`../logs/CTIConnector`	`../logs/PSTNConnector`

Table 9: Default Values for Selected log Options (Continued)

Option Name	Default Value							
	RM	MCP	CCP	FM	RS	SSG	CTIC	PSTN
expire	20 (files)	10 (files)	10 (files)	10 (files)	false	7 days	20 (files)	7 days
interaction	../logs/ResourceMgr	../logs/MCP	../logs/ccp	../logs/fm	Empty	../logs/SSG	../logs/CTIConnector	../logs/PSTNConnector
message_format	short	short	short	short	full	short	short	short
segment	10000 KB	10000 KB	10000 KB	10000 KB	10 MB	10000 KB	10000 KB	10000 KB
standard	../logs/ResourceMgr	../logs/MCP_standard	../logs/ccp_standard	../logs/fm	stdout	../logs/SSG	../logs/CTIConnector	../logs/PSTNConnector
time-format	time	ISO8601	ISO8601	time	time	ISO8601	time	ISO8601
trace	../logs/ResourceMgr	../logs/MCP	../logs/ccp	../logs/fm	Empty	../logs/SSG	../logs/CTIConnector	../logs/PSTNConnector
verbose	standard	interaction	interaction	standard	trace	standard	standard	standard

Configuring SNMP

Table 10 describes the option for `snmp task timeout`.

Table 10: Selected Configuration Options—snmp Section

Option Name	Description	Valid Values and Syntax
SNMP Task Timeout	Specifies the maximum time interval, in milliseconds, that SNMP waits for a new task.	Any integer greater than zero. **Default Value:** `100`

Configuring Client-Side Connections

GVP components provide the ability to configure the server-side ports in a UDP or a TCP connection. This includes situations where the GVP components act as the server (for example, Call Control Platform’s SIP service port), so that modification of the corresponding configuration results in a different listening port. It also includes situations where the GVP components acts as the client (for example, Call Control Platform's connection to Reporting Server), so that modification of the port option results in an attempt to connect to a different remote port.

Table 11 describes the connections for each GVP component.

Table 11: Client Connections

Component	Connection	Purpose
Call Control Platform (CCP)	Configuration Server —TCP dynamically configured by the OS	Allows the CCP to receive configuration data and updates from the Configuration Server.
	Message Server—TCP dynamically configured by the OS	Sends logs to the Message Server if sink logging is turned on.
	Local Control Agent (LCA)—TCP dynamically configured by the OS	Allows CCP to send status information to the Solution Control Server (SCS).
	SIP—UDP option `sip.transport.x` **Default value:** `5068` SIP—UDP/TCP dynamically configured by the OS	Allows the CCP to provide SIP Service
	Reporting Server—TCP dynamically configured by the OS	Allows the CCP to send sink, OR and CDR information to the Reporting Server.
Call Control Platform (CCP) (continued)	HTTP—TCP dynamically configured by the OS	Allows the CCP to fetch pages and HTTP messaging.
	SNMP—TCP dynamically configured by the OS	Allows the CCP to connect to SNMP Master Agent

Table 11: Client Connections (Continued)

Component	Connection	Purpose
Media Control Platform (MCP)	Configuration Server—TCP dynamically configured by the OS	Allows the MCP to receive configuration data and updates from the Configuration Server.
	Message Server—TCP dynamically configured by the OS	Sends logs to the Message Server if sink logging is turned on.
	Local Control Agent (LCA)—TCP dynamically configured by the OS	Allows the MCP to send status information to the Solution Control Server (SCS).
	SIP—UDP option `sip.transport.x` **Default value:** `5070` SIP—TCP dynamically configured by the OS	Allows the MCP to provide SIP Service.
	Reporting Server—TCP dynamically configured by the OS	Allows the MCP to send logging, OR, and CDR information to the Reporting Server.
	HTTP—TCP dynamically configured by the OS	Allows the MCP to fetch pages and HTTP messaging.
	PageCollector—TCP dynamically configured by the OS with ports ranging from 1024 to 65535.	Allows the legacy interpreter (GVPi) to fetch pages from the Web Server.
	SNMP—TCP dynamically configured by the OS	Allows the MCP to connect to SNMP Master Agent
	MRCPv2—TCP/UDP option: `mrcpv2client.sip.localport` **Default value:** `7080`	Allows local non-security SIP messages.

Table 11: Client Connections (Continued)

Component	Connection	Purpose
Media Control Platform (MCP) (continued)	RTP—UDP options: • `mcp.rtp.portlow` • `mcp.rtp.porthigh` **Default value:** `10000-65535`	Allows the MCP to receive media data.
	RTP for ASR/TTS UDP options: • `mtinernal.rtp_min_port` • `mtinternal.rtp_max_port` **Default value:** `20000-30000`	Allows the MCP to perform recognition and play synthesized text.
Reporting Server (RS)	Configuration Server—TCP dynamically configured by the OS	Allows the RS to receive configuration data and updates from the Configuration Server.
	Message Server—TCP dynamically configured by the OS	Sends logs to the Message Server if sink logging is turned on.
	Local Control Agent (LCA)—TCP dynamically configured by the OS	Allows the RS to send status information to the Solution Control Server (SCS), and to receive High Availability instructions.
	DBMS (Oracle or MSSQL)	Allows the RS to connect to an Oracle or MSSQL database for data storage. There can be up to seven connections.
Resource Manager (RM)	Configuration Server—TCP dynamically configured by the OS	Allows the RM to receive configuration data and updates from the Configuration Server.
	Message Server—TCP dynamically configured by the OS	Sends logs to the Message Server if sink logging is turned on.
	Local Control Agent (LCA)—TCP dynamically configured by the OS	Allows the RM to send status information to the Solution Control Server (SCS).
	SIP—TCP/UDP dynamically configured by the OS	Allows the RM to provide SIP Service.

Table 11: Client Connections (Continued)

Component	Connection	Purpose
Resource Manager (RM)	Reporting Server—TCP dynamically configured by the OS	Allows the RM to send logging, OR, and CDR information to the Reporting Server.
	Cluster—TCP dynamically configured by the OS	Allows RM to monitor the primary and backup connections for High Availability configuration.
Supplementary Services Gateway (SSG)	Configuration Server—TCP dynamically configured by the OS	Allows the SSG Connector to receive configuration data and updates from the Configuration Server.
	Message Server—TCP dynamically configured by the OS	Sends logs to the Message Server if sink logging is turned on.
	Local Control Agent (LCA)—TCP dynamically configured by the OS	Allows the SSG to send status information to the Solution Control Server (SCS).
	HTTP Client port—TCP option `[fm]portrange` **Default value**: Empty	Allows the SSG to post the call results to the Notification URL that is specified in the call request sent by the Trigger Application.
	T-Lib Port—TCP dynamically configured by the OS	Allows the SSG to send and receive T-lib messages from SIP Server.

Table 11: Client Connections (Continued)

Component	Connection	Purpose
CTI Connector (CTIC)	Configuration Server—TCP dynamically configured by the OS	Allows the CTI Connector to receive configuration data and updates from the Configuration Server.
	Local Control Agent (LCA)—TCP dynamically configured by the OS	Allows the CTI Connector to send status information to the Solution Control Server (SCS).
	SIP—UDP option `sip.transport.x` **Default value:** `5080` SIP—TCP dynamically configured by the OS	Allows the CTI Connector to provide SIP Service.
	IVR Server—TCP dynamically configured by the OS with ports in a range is determined by the configuration parameter `IVRSCClientPortRange` in the corresponding IVR Server section.	Allows the CTI Connector to receive messages from IVR Server.
PSTN Connector (PSTNC)	Configuration Server—TCP dynamically configured by the OS	Allows the PSTN Connector to receive configuration data and updates from the Configuration Server.
	Message Server—TCP dynamically configured by the OS	Sends logs to the Message Server if sink logging is turned on.
	Local Control Agent (LCA)—TCP dynamically configured by the OS	Enables the PSTN Connector to send status information to the Solution Control Server (SCS).
	SIP—UDP option: • `GatewayManager.LocalSIPPort` **Default value:** `5060`	Enables the PSTN Connector to provide SIP Service

Table 11: Client Connections (Continued)

Component	Connection	Purpose
MRCP Proxy (MRCPP)	Configuration Server—TCP dynamically configured by the OS	Enables the MRCPP to receive configuration data and updates from the Configuration Server.
	Message Server—TCP dynamically configured by the OS	Sends logs to the Message Server if sink logging is turned on.
	Local Control Agent (LCA)—TCP dynamically configured by the OS	Enables the MRCPP to send status information to the Solution Control Server (SCS).
	Reporting Server—TCP dynamically configured by the OS.	Enables the MRCPP to send OR log information to the Reporting Server.
	ASR and TTS speech engines—TCP dynamically configured by the OS.	Enables MRCPP to send RTSP requests and receive responses.
Policy Server (PS)	Configuration Server—TCP dynamically configured by the OS	Enables the PS to receive configuration data and updates from the Configuration Server.
	Message Server—TCP dynamically configured by the OS	Enables PS to send logs to the Message Server if sink logging is turned on.
	Local Control Agent (LCA)—TCP dynamically configured by the OS	Enables PS to send status information to the Solution Control Server (SCS).
	HTTP—TCP dynamically configured by the OS **Default value**: 8090	Enables PS to receive incoming policy queries from Genesys Administrator.

Customizing SIP Responses

This section lists the configuration options that enable you to customize the SIP responses and alarms that the Resource Manager, Media Control Platform, Call Control Platform, and CTI Connector signal for certain events and conditions.

For more information about the SIP response codes that are generated, and how the following configuration options relate to them, see Table 100 on page 470. For more information about all the configuration options, see the *Genesys Voice Platform 8.1 Configuration Options Reference.*

To customize GVP behavior in response to error conditions and other events, consider the following options.

- **Resource Manager:**
  - rm.conference-sip-error-respcode
  - rm.options_response_contenttype
  - rm.options_response_msg_body
  - rm.resource-no-match-respcode
  - rm.resource-unavailable-respcode
  - rm.suspend-mode-respcode
  - <gateway resource group>.noresource-response-code
- **IVR Profile:**
  - gvp.policy.announcement-call-forbidden-response-code
  - gvp.policy.announcement-call-forbidden-set-alarm
  - gvp.policy.announcement-usage-limit-exceeded-set-alarm
  - gvp.policy.ccxml-usage-limit-exceeded-respcode
  - gvp.policy.ccxml-usage-limit-exceeded-set-alarm
  - gvp.policy.conference-forbidden-respcode
  - gvp.policy.conference-forbidden-set-alarm
  - gvp.policy.conference-usage-limit-exceeded-respcode
  - gvp.policy.conference-usage-limit-exceeded-set-alarm
  - gvp.policy.dialing-rule-forbidden-respcode
  - gvp.policy.dialing-rule-forbidden-set-alarm
  - gvp.policy.external-sip-forbidden-respcode
  - gvp.policy.external-sip-forbidden-set-alarm
  - gvp.policy.external-sip-usage-limit-exceeded-set-alarm
  - gvp.policy.inbound-usage-limit-exceeded-set-alarm
  - gvp.policy.inbound-call-forbidden-response-code
  - gvp.policy.msml-call-forbidden-response-code
  - gvp.policy.msml-call-forbidden-set-alarm
  - gvp.policy.msml-usage-limit-exceeded-set-alarm
  - gvp.policy.outbound-call-forbidden-respcode
  - gvp.policy.outbound-call-forbidden-set-alarm
  - gvp.policy.outbound-usage-limit-exceeded-set-alarm
  - gvp.policy.outbound-call-forbidden-response-code
  - gvp.policy.transfer-forbidden-respcode
  - gvp.policy.transfer-forbidden-set-alarm
  - gvp.policy.usage-limit-exceeded-respcode
  - gvp.policy.usage-limit-exceeded-set-alarm
  - gvp.policy.voicexml-dialog-forbidden-respcode
  - gvp.policy.voicexml-dialog-forbidden-set-alarm
  - gvp.policy.voicexml-usage-limit-exceeded-respcode
  - gvp.policy.voicexml-usage-limit-exceeded-set-alarm

- **Media Control Platform:**
  - `sip.sendalert`
  - `sip.copyunknownheaders`
- **Call Control Platform:**
  - `ccpccxml.defaultrejectcode`
  - `ccpccxml.sip.send_progressing`
  - `session.copy_unknown_headers`
  - `sip.copyunknownheaders`
  - `sip.OPTIONS.header.Accept`
  - `sip.OPTIONS.header.Accept-Encoding`
  - `sip.OPTIONS.header.Accept-Language`
  - `sip.OPTIONS.header.Allow`
- **CTI Connector:**
  - `sip.copyunknownheaders`

To further customize the SIP response codes for specific situations, use the `<hints>` attribute of the `<redirect>` and `<reject>` tags—the `responseCode` property of the `hints` object specifies the response code to be used.

Configuring Session Timers and Timeouts

This section describes two kinds of timeouts that determine the length of a session and affect responses to service requests:

- The session inactivity timers, expiry timers, and timeouts that the Resource Manager uses to manage sessions (see "Resource Manager Session Timers").
- Additional timeouts that are set at the resource level, or specified in SIP or HTTP requests (see "Additional Timeouts" on page 83).

Resource Manager Session Timers

Table 12 summarizes the configuration options that determine the session timers that the Resource Manager uses to manage sessions, in the order in which the Resource Manager applies them. Configure these options on the `IVR Profile`, `Tenant`, or `Application` (Media Control Platform, Call Control Platform, Resource Manager) objects, as applicable for your deployment.

The Resource Manager adds a `Session-Expires` header to initial `INVITE` requests if one is not present, and if the request does not contain the `timer` option in the `Supported` header. The value of the `Session-Expires` header is the configured value of the applicable session inactivity timer.

Table 12: Session Timer Configuration Options

Option Name	Description	Valid Values and Syntax
Environment Tenant/ IVR Profile		
gvp.general. sip.sessiontimer	The timeout interval, in seconds, for the SIP session that is executed for the IVR Profile. If the Resource Manager receives no SIP messages associated with this call leg within the timeout interval, it considers the call leg to have ended. For the call leg associated with the IVR Profile for this tenant, the value that you configure for this `sip.sessiontimer` option overrides session expiry timeouts that are set at the level of the resource and the Resource Manager.	Any positive integer. **Default value:** Empty
PSTN Connector `GatewayManager` **Section**		
Enable Session Timer	Specifies whether the session timers is enabled for a call session.	• `True` • `False` **Default value:** `True`
Session Timer Interval (secs)	The time interval, in seconds, for which a call session must be refreshed before it expires.	An integer in the range of 90 to 86400. **Default value:** `1800`
Session Minimum Timer Interval (secs)	The minimum time interval, in seconds, for which a call session must be refreshed.	An integer in the range of 90 to 1800. **Default value:** `90`
Session Timer Refresh	Specifies which user agent is to initiate the refresh of a call session.	• `0`—Local • 1—Remote **Default value:** `0`

Table 12: Session Timer Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Resource Manager `proxy` **Section**		
Min SE	The minimum value of the `Session-Expires` header, in seconds, that the Resource Manager will accept. If an incoming SIP request contains a `Session-Expires` header with a value that is less than `sip.min_se`, the Resource Manager rejects the `INVITE` request with a `422 (Session interval too small)` response. **Changes take effect:** After restart.	Any unsigned integer. **Default value:** `90`
Session Expires	The default timeout interval, in seconds, for Media Control Platform or Call Control Platform sessions. If no re-`INVITES` are sent or received within the timeout period, the session expires. If a different timeout has been set for a particular VoiceXML or CCXML application, it overrides the value of this `sip.sessionexpires` option.	An integer in the range of `90–3600`. **Default value:** `1800`
Resource Manager `proxy` **Section**		
Min SE	The minimum value of the `Session-Expires` header, in seconds, that the Resource Manager will accept. If an incoming SIP request contains a `Session-Expires` header with a value that is less than `sip.min_se`, the Resource Manager rejects the `INVITE` request with a `422 (Session interval too small)` response. **Changes take effect:** After restart.	Any unsigned integer. **Default value:** `90`
Session Timeout	The timeout interval, in seconds, for each SIP session (call leg) that the Resource Manager handles. If a different timeout has been set for a particular resource or XML application, it overrides the Resource Manager session expiry timeout for the applicable session. **Changes take effect:** After restart.	Any unsigned integer. **Default value:** `1800`

Table 12: Session Timer Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Resource Manager `registrar` Section		
Registrar Max Expiry Time	The maximum expiry time, in seconds, of this registrar. If the client requests an expiry time greater than this value, this `sip.registrar.maxexpirytime` value is returned.	An integer in the range of `60–7200`. **Default value:** `60`
Registrar Min Expiry Time	The minimum expiry time, in seconds, of this registrar. If the client requests an expiry time smaller than this value, the request is rejected, with this value in the `Min-Expires` header.	An integer in the range of `60–7200`. **Default value:** `60`

Additional Timeouts

The following timeouts and timers are also important for GVP behavior:

- Resource Manager, Media Control Platform, and Call Control Platform:
  - `dproxy.sip.timer.ci_proceeding`—The timeout for the client INVITE. The timer starts after a 1xx response is received for a client INVITE. If a final response is not received before the timer expires, the SIP session and dialog is destroyed without further notice to the UAS. This timer should be greater than the connect timeout of the outbound call (depending on how the outbound call is initiated, the connect timeout can be specified in the transfer tag, or in the remdial command). Otherwise, the `dsip.timer.ci_proceeding` timer will trigger before the connect timeout occurs, which overrides the connect timeout. (The default is 120000 ms, or 120 seconds.)
- Resource Manager:
  - `sip-timer_B`—The timeout Resource Manager uses when selecting a resource for messaging. If no `1xx` provisional response is received with this timeout, Resource Manger considers this resource unreachable, and attempts to select another resource.
- Media Control Platform:
  - `mpc.rtp.timeout`—The timeout for the RTP/RTCP stream. (The default is 60000 ms.)
  - `msml.cpd.beeptimeout`—The timeout for the default beep. (The default is 30 seconds.)
  - `msml.cpd.postconnecttimeout`—The timeout for the CPD postconnect. (The default is 30 seconds.)
  - `msml.cpd.preconnecttimeout`—The timeout for CPD preconnect. (The default is 30 seconds.)

- `sessmgr.acceptcalltimeout`—The timeout for the platform to accept inbound calls, after alerting is issued. (The default is 30000 ms.)
- `sessmgr.maxincalltime`—The maximum call time for inbound calls. (The default is 0—disabled.)
- `sip.hfdisctimer`—The timeout to terminate a SIP hookflash transfer. (The default is 5000 ms.)
- `sip.referxferwaitbye` - After a REFER transfer, timeout to wait for BYE message from the remote end before sending BYE to disconnect the call. If it is zero, it will send BYE right after a NOTIFY/200 is received. If it is non-zero, it will wait for the configured timeout (in milliseconds) before sending the BYE. Values are specified in millisecond. (The default is 0 - send BYE immediately).
- `stack.connection.timeout`—The connection timeout for the MRCP Client stack to establish a TCP connection to the MRCP server. (The default is 10000 ms.)
- `vrm.client.timeout`—The connection timeout for the MRCP Client to receive a response from the MRCP server. (The default is 10000 ms.)
- `vxmli.default.connecttimeout`—The default value of the `connecttimeout` attribute for bridge or consultation transfers, if not provided. (The default is 30000 ms.)
- `vxmli.initial_request_fetchtimeout`—The fetch timeout for the initial VoiceXML page. (The default is 30000 ms.)
- `vxmli.max_script_time`—The maximum time allowed for each script or ECMAScript expression to be executed. (The default is 2000 ms.)

- Call Control Platform:
  - `ccxmli.fetch.timeout`—The default timeout for the fetch of the initial page to be completed. (The default is 30 seconds.)
- Supplementary Services Gateway:
  - `ssg.ReqAccOnResourceDNErrTimeoutSecs`—The timeout for the Supplementary Services Gateway to reject new requests from tenant applications if the Supplementary Services Gateway fails to register a Resource DN with SIP Server. (The default is 900 seconds).
  - `ssg.ReqAccOnSIPSConnErrTimeoutSecs`—The default timeout for the Supplementary Services Gateway to reject new requests from tenant applications if the Supplementary Services Gateway fails to connect to SIP Server. (The default is 900 seconds).
- CTI Connector:
  - `ctic.connectcalltimeout`—The default timeout that CTI Connector waits for an outbound call to connect. (The default is 6000 ms)
  - `IServer_Sample.keepaliveresptimeout`—The timeout that CTI Connector waits for a response from IVR Server. (The default is 3 seconds.)
  - `IVRSC.scriptidfetchtimeout`—The time to wait for a response to fetch the script id from URS. (The default is 5000 ms.)

- PSTN Connector:
  - GatewayManager.XferConnectTimeoutMSec—The time to wait for the transfer result after issuing a blind or consult transfer request. (The default is 60000 ms)
- MRCP Proxy:
  - timeout.back_in_service—The time to wait for the server to be put back into service after it encounters errors such as timeout or TCP connection error.
  - timeout.barge_in_occurred—The timeout for a barge-in to occur.
  - timeout.clean_loop—The time interval in which idle sessions are cleaned, as determined by timeout.max_idle configuration option.
  - timeout.close_session—The time it takes for a Close-Session request to expire.
  - timeout.control—The time it takes for a CONTROL message to expire.
  - timeout.define_grammar—The time it takes for a DEFINE-GRAMMAR message to expire.
  - timeout.get_params—The time it takes for a GET-PARAMS message to expire.
  - timeout.get_result—The time it takes for a GET-RESULT message to expire.
  - timeout.get_server_info—The time to wait to get a response to a Get-Server-Info request (ping) before timing out.
  - timeout.lca_calibrate—The time to wait (at connect or re-connect) before an application mode query is sent to the LCA. After this timeout expires the query can be sent.
  - timeout.max_idle—The maximum amount of time a session can be idle before it is terminated.
  - timeout.open_session—The time it takes for an Open-Session to expire.
  - timeout.pause—The time it takes for a PAUSE message to expire.
  - timeout.recog_start_timers—The time it takes for a RECOGNITION-START-TIMERS message to expire.
  - timeout.recognize—The time it takes for a RECOGNIZE message to expire.
  - timeout.reconnect_interval—The time to wait before a reconnect attempt is made, if the TCP connection is not yet established with the MRCP server.
  - timeout.resume—The time it takes for a RESUME message to expire.
  - timeout.set_params—The time it takes for a SET-PARAMS message to expire.
  - timeout.speak—The time it takes for a SPEAK message to expire.
  - timeout.stop—The time it takes for a STOP message to expire.

- `connection.timeout`—The time it takes before a connection times out when the SRM MRCPv1 and MRCPv2 stack is attempting to establish a TCP connection to the server.
- `timeout`—The maximum amount of time that SNMP can wait for a new task.

Chapter

4 Configuring the Resource Manager

This chapter describes the Resource Manager configuration requirements for your Genesys Voice Platform (GVP) deployment. It contains the following sections:

Task Summary: Configuring the Resource Manager

Task Summary: Configuring the Resource Manager summarizes the configuration steps and options to implement Resource Manager functionality in your GVP deployment.

Task Summary: Configuring the Resource Manager

Objective	Related Procedures and Actions
Set up the Resource Manager to function as SIP Proxy, SIP Registrar, SIP Notifier, and resource monitor and manager.	See "Configuring SIP Communication and Routing" on page 42. To secure SIP communications between the Resource Manager and the other GVP components, ensure that you specify a transport for the Transport Layer Security (TLS) protocol and a secure routeset for outbound calls.
Provision GVP resources.	See "Configuring Logical Resource Groups" on page 93.

Task Summary: Configuring the Resource Manager (Continued)

Objective	Related Procedures and Actions
Configure the IP DiffServ (ToS).	Set the SIP packet's ToS using `[sip]transport.[n].tos` See "Configuring SIP Communication and Routing" on page 42".
Provision IVR Profiles.	See Chapter 6 on page 107.
Configure conferencing.	See "Enabling Conference Services" on page 62.
Configure reporting.	See "Configuring Reporting" on page 63.
Customize logging.	See "Configuring Logging" on page 66.
Customize session management behavior and performance.	See "Configuring Session Timers and Timeouts" on page 80. See also the parameters in the `proxy` section that specify parameters such as the number of threads and connections.
Customize client-side communication ports.	See "Configuring Client-Side Connections" on page 72.
Customize Resource Manager messaging.	See "Configuring SNMP" on page 72 and Table 100 on page 470.

Important Resource Manager Configuration Options

This section describes the key configuration options that you either must or may want to customize.

Configure the options in Genesys Administrator on the `Provisioning > Environment > Applications > <Resource Manager> > Options` tab. For the detailed steps to configure option settings, see Procedure: Viewing or modifying GVP configuration parameters, on page 34.

Except where otherwise indicated, all changes to Resource Manager parameters take effect after you restart the Resource Manager.

The Resource Manager configuration options are in the following configuration sections:

- `cluster`—Parameters that determine the High Availability behavior.
- `gvp`—Parameters that monitor the health and status of the network interfaces and bonding drivers.
- `ems`—Parameters that the determine Reporting behavior. (See Table 6 on page 64.)

- `log`—Parameters the determine the behavior for Management Framework logging. (See “Configuring Logging” on page 66.)
- `monitor`—Parameters that support the Resource Manager in its role as manager of GVP resources.
- `proxy`—Parameters that determine the behavior of the Resource Manager in its role as SIP Proxy and session manager.
- `registrar`—Parameters that determine the behavior of the Resource Manager in its role as SIP Registrar.
- `rm`—Parameters that determine the behavior of the Resource Manager in its role as manager of GVP services.
- `snmp`—Parameters that determine the behavior of SNMP. (See “Configuring SNMP” on page 72.)
- `subscription`—Parameters that control the `SUBSCRIBE/NOTIFY` behavior.

Table 13 provides information about important Resource Manager parameters that are not described in Chapter 3 on page 41. Table 13 provides parameter descriptions, and also the default parameter values that are preconfigured in the Resource Manager `Application` object.

For information about all the available configuration options for the Resource Manager, see the *Genesys Voice Platform 8.1 Configuration Options Reference*.

Table 13: Selected Resource Manager Configuration Options

Option Name	Description	Valid Values and Syntax
	`cluster` Section	
Election Timer	Specifies the interval, in milliseconds, in which this node waits for a response from its remote members. If there is no response within this time, the local Resource Manger becomes the active node.	An integer in the range of 1000–10000 **Default value:** `3000`
FailOver Batch Script	Specifies the path to the fail over script.	`Install dir /bin/nbl.bat` **Default value:** `$InstallationRoot$/bin/nbl.bat`
Heartbeat Interval	Specifies the interval, in milliseconds, for which the members of the cluster check each other’s health status.	An integer in the range of 2000–60000 **Default value:** `2000`

Table 13: Selected Resource Manager Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Cluster Standby Mode	This parameter has effect only if cluster.virtual-ip parameter is non-empty. If `true`, this parameter mandates that the RM cluster run in hot-standby redundancy mode where call data is replicated to other RM node for SIP dialog redundancy. Otherwise, this parameter mandates that the RM cluster be run in warm-standby redundancy mode where SIP dialog redundancy is not supported, but the new requests will be handled by the other healthy RM node	• `True` • `False` **Default value:** `False`
Members	List of ID's of the members in the cluster (unsigned integers 1 to 32 delimited by space). For NLB, the ID's correspond to the unique host identifier (priority) number specified for each of the NLB cluster machines.	**Default value:** `1 2`
Members 1	Describes the IP and TCP port on which the member ID 1 can be reached. The format is IP:Port where IP and Port specifies the IP and port where this RM node can be reached for cluster communication	`<first member IP address>:<first member cluster communication port>` **Default value:** Empty
Members 2	Describes the IP and TCP port on which the member ID 2 can be reached. The format is IP:Port where IP and Port specifies the IP and port where this RM node can be reached for cluster communication	`<second member IP address>:<second member cluster communication port>` **Default value:** Empty
My Member ID	Indicates this cluster manager instance's member ID (select one among the ID's listed in cluster.members	An integer. **Default value:** Empty
Cluster Virtual IP Address	If non-empty, this parameter indicates that this RM node is part of a RM cluster and the value specifies the virtual IP address of the RM cluster this RM node is part of. In stand-alone mode, this parameter must be left empty.	An integer. **Default value:** Empty
Cluster HA Mode	Specifies which cluster the Resource Manager instances are configured.	• `none`—stand alone RM • `active-standby`—RM in cluster • `active-active`—external load balancer **Default value:** `none`

Table 13: Selected Resource Manager Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
`rm` Section		

FIPS Mode Enabled	Enables FIPS mode in RM.	`True`
		`False`
		Default value: `False`
		Changes take effect: start/restart

SIP Header for DNIS	The header from which the Resource Manager will retrieve the DNIS, to identify which IVR Profile to use. Ensure that the value you specify is consistent with Media Gateway behavior, so that the `INVITE` messages that SIP Server forwards to the Resource Manager have the DNIS information in the expected header. • If the value of this parameter is `History-Info` but there is no `History-Info` header in the SIP `INVITE`, the Resource Manager picks up the DNIS from the `To` header. • If the value of the specified header in the SIP `INVITE` is not a valid DNIS, the Resource Manager cannot map the SIP request to an IVR Profile, and it defaults to the next behavior to select the IVR Profile. • If the P-Called-Party-ID header is present, SIP Header for DNIS is ignored. **Changes take effect:** Immediately.	• `To` • `Request-Uri` • `History-Info` **Default value:** `History-Info`
Default Resource Port Capacity	Specifies the port capacity that is assigned to each physical resource. **Note:** This parameter can be overridden by the `MaxPorts` configuration during post installation activities. For more information, see the *Genesys Voice Platform 8.1 Deployment Guide*.	Any unsigned integer. **Default value:** `500`
`monitor` Section		
SIP Resource OPTIONS Interval	The interval, in milliseconds, at which the Resource Manager sends `OPTIONS` messages to a healthy resource, to determine whether the resource is alive.	Any unsigned integer. **Default value:** `5000`

Table 13: Selected Resource Manager Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
SIP Unavailable Resource OPTIONS Interval	The interval, in milliseconds, at which the Resource Manager sends `OPTIONS` messages to a dead resource to determine whether the resource is alive.	Any unsigned integer. **Default value:** `5000`
SIP Release Conference Resource on Failure	Specifies how to handle new incoming calls that are joining the conference if the conference resource goes offline. If set to `True`, all conference sessions are released, and the new incoming calls are routed to the next available resource. If set to `False`, all conference sessions are released, and the new incoming calls will receive an error.	• `True` • `False` **Default value:** `True`
	`proxy` Section	
Preferred IP Version Used in SIP Proxy	Specifies the preferred IP version when multiple IP addresses with different IP versions are resolved from a destination address. The first address from the list with the preferred IP version is used. However, if there the sip.transport.x configuration option is not defined with the preferred version, other version are used.	• `ipv4` • `ipv6` **Default value:** `ipv4`

Table 13: Selected Resource Manager Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
IP Type of Service for SIP Transport	Specifies the IP differentiated services field (ToS) to set in all outgoing SIP packets over the SIP transport. **Notes:** • For Windows Server 2003, the ToS must be enabled in the registry. See `http://support.microsoft.com/kb/248611` • For Windows Server 2008, the ToS configuration is not supported. It must be configured at the OS level. You can define per executable and per port, and what type of `DiffServ` bits to set on the outgoing packets using the QoS policy defined in the following article. `http://technet.microsoft.com/en-us/library/cc771283.aspx` • For all Operating Systems, when the SIP/RTP packets are sent across different subnets, the router may reset the DiffServ bits in the IP header even though it was set by MCP.	Range: `0-255` Examples: • `0`—Disabled • `16`—IPTOS LOWDELAY (0x10) • `32`—IPTOS PREC PRIORITY (0x20) • `64`—IPTOS PREC CRITICAL (0x40) • `184`—DiffServ EF (Expedited Forward 0xBB) **Default value:** `0`
Record Route Host	Specifies the host to be used for the Record-Route when an INVITE is forwarded when Resource Manager is in stand-alone mode. The value specified can either be configured as an IP address, or FQDN. If the value is empty, the IP address of the outgoing transport is used.	`<hostname or IP address>`

Configuring Logical Resource Groups

For each type of service that GVP provides (VoiceXML, CCXML, Conference, MSML, Announcement, Recording Server), you must create and configure a logical resource group (LRG) that the Resource Manager will use as its resource pool. You must create a resource group for each type of service, even if there is only one resource available to provide that service (in other words, if the group has a single member).

Use the Genesys Administrator Resource Group Wizard (`Provisioning > Voice Platform > Resource Groups`) to create, modify, or view settings for the resource group and to specify the resources that belong to each group. For detailed information about using the Resource Group Wizard, see Procedure: Configuring logical resource groups.

You identify the actual resource hosts and applications in the connections that you configure for the Resource Manager. In addition, you must create a remote access point for the `recordingserver` resource that points to the Resource Manager. See Procedure: Creating the resource access point for Recording Server, on page 99.

For more information about logical resource groups, see the chapter about post-installation activities in the *Genesys Voice Platform 8.1 Deployment Guide*.

The following procedure provides the detailed steps to use the Resource Group Wizard.

Procedure: Configuring logical resource groups

Purpose: To create or modify the property information that is shared by a logical group of GVP resources managed by a particular Resource Manager.

Unless otherwise indicated, changes to logical resource group configurations take effect immediately.

Prerequisites

- The GVP `Application` objects have been installed, as described in the *Genesys Voice Platform 8.1 Deployment Guide.*
- You are logged in to Genesys Administrator. To access Genesys Administrator, go to the following URL:

  `http://<Genesys Administrator host>/wcm`

Start of procedure

1. In Genesys Administrator, go to the `Provisioning > Voice Platform > Resource Groups` panel (see Figure 5).

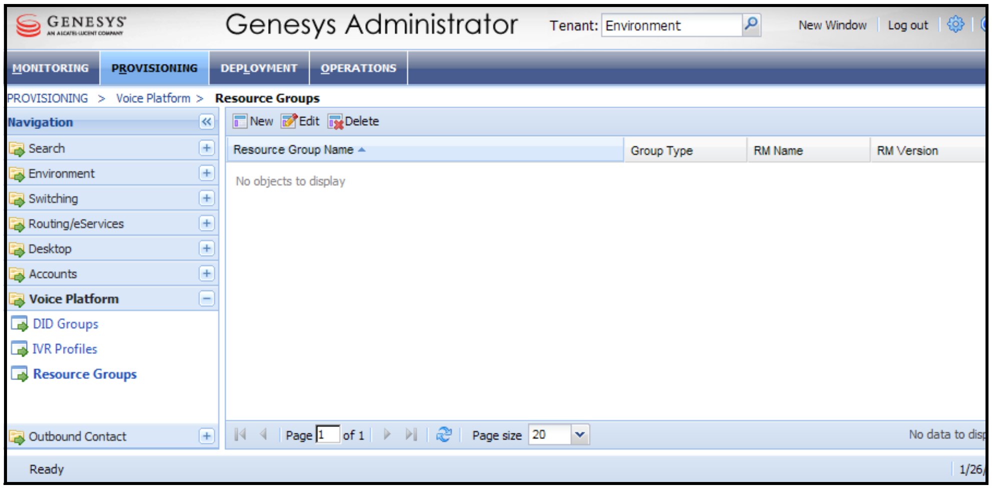


Figure 5: The Resource Groups Panel

2. Do one of the following to invoke the wizard:
   - To create a new group, click `New`.
   - To modify the configuration parameters for an existing group, select the group name and click `Edit`.
   - To delete an existing group, click `Delete`.

   For more information on how to use the Resource Groups Wizard, see the *Genesys Voice Platform 8.1 Deployment Guide*.

End of procedure

Next Steps

- If required, configure the `noresource-response-code` option in the `<gateway resource group>` section of the Resource Manager `Application` object.

  The default behavior for the Resource Manager with regard to `gateway` resources is not to retry failed requests. To configure the Resource Manager to automatically retry other resources in a `gateway` resource group, specify the required SIP failure response codes in the `noresource-response-code` option. This option does not appear in the Resource Management Wizard, you configure it on the `Provisioning > Environment > Applications > <Resource Manager> > Options` tab.

- Manually set the capabilities and the preferences for the Logical Resource Group (see page 98).

  Table 14 provides information about the Resource Manager parameters for logical groups.

Table 14: Logical Group Section Configuration Options

Option Name	Description	Valid Values and Syntax
Group Type	Specifies the type of logical resource group.	• `Media Control Platform` • `Call Control Platform` • `Gateway` • `CTI Connector` • `Recording Server` **Default value:** Empty
CTI Usage	Specifies whether the Resource Manager will use CTI Connector for a Gateway logical group resource.	• `Always off`—Resource Manager does not use CTI Connector, and proceeds with the call using DNIS-IVR Profile mapping. • `Always On`—Resource Manager will not map the call. • `Based on DID lookup`—Resource Manager performs the IVR Profile lookup for the call and forwards it to CTI Connector with the CTI service parameters configured in the IVR Profile. **Default value:** Empty
Port Capacity	Specifies the port capacity of all resources in this logical resource group combined. Individual resource port capacity will be ignored. **Note:** The port capacity option is available for the Recording Server resource group only when parallel-forking is used as the load balancing scheme.	• **Default value:** `500`
Load Balancing Scheme	The distribution algorithm that the Resource Manager will use to select a resource within this logical resource group. **Note:** The parallel forking option value is available for the Recording Server LRG only.	• `Round-robin` • `Least used` • `Least percent` • parallel forking **Default value:** `Round-robin`

Table 14: Logical Group Section Configuration Options (Continued)

<table>
<tr><th>Option Name</th><th>Description</th><th>Valid Values and Syntax</th></tr>
<tr><td>Max Conference Size
(For MCP)</td><td>The maximum number of participants allowed in a conference.</td><td>An integer in the range of 0–32.</td></tr>
<tr><td>Max Conference Count
(For MCP)</td><td>The maximum number of conferences allowed for a resource.</td><td>Any integer.</td></tr>
<tr><td>Monitoring Method</td><td>The method that the Resource Manager will use to determine whether the physical resources that belongs to the logical resource group are alive and healthy.</td><td>• SIP OPTIONS—Resource Manager will use SIP OPTIONS messages.
• None—Resource Manager will not monitor resource health. It assumes that resources in this group are always alive.
Default value: SIP OPTIONS</td></tr>
<tr><td>Geo Location
(Optional)</td><td>The geographical location of the resource.</td><td>A character string.</td></tr>
<tr><td>Capability
(Manually configured)</td><td>The list of values supported by the resources within the logical resource that corresponds to the SIP INVITE capability. This is read from the configuration parameter capability of a logical resource, and the parameter value has the following syntax:
[cap_NameA]=[cap_ValueA1],…,[cap_ValueAm];
[cap_NameB]=[cap_ValueB1],·,[cap_ValueBn];…;
[cap_NameM]=[cap_ValueM1],·,[cap_ValueMi]
Note: The set of [cap_NameX] must be unique.</td><td>A character string.</td></tr>
<tr><td>Preference
(Manually configured)</td><td>Specifies which logical resource group has preference. If there are two resource groups with difference preference numbers, the logical resource group with lower number is considered first for all calls.</td><td>Any positive integer.</td></tr>
<tr><td>Address of Record (AOR)
(Manually configured)</td><td>Specifies the list of contacts for the physical resource. This parameter must be configured manually in order for Resource Manager to support static routing (see Procedure: Configuring Resource Group capabilities, preferences, and AOR, on page 98)</td><td>[sip|sips]:<address>:<port>.</td></tr>
</table>

The following procedure provides the detailed steps to manually configure the Logical Resource Group capabilities, preferences and AOR.

Procedure: Configuring Resource Group capabilities, preferences, and AOR

Purpose: To configure capabilities, preferences, and AOR for the logical resource group.

Prerequisites

- The GVP `Application` objects have been installed, as described in the *Genesys Voice Platform 8.1 Deployment Guide.*
- The Logical Resource Group has been created.
- You are logged in to Genesys Administrator. To access Genesys Administrator, go to the following URL:
  `http://<Genesys Administrator host>/wcm`

Start of procedure

1. In Genesys Administrator, go to the `Provisioning > Environment > Business Units/Sites`
2. Select the appropriate tenant.
3. Select the MCP `Application`, and click `Edit`.
4. Add the `Capability`, `Preference`, and `AOR` parameters to the `gvp.lrg` section. For more information on how to add options using Genesys Administrator, see the *Framework 8.1 Genesys Administrator Help*.

   See Table 14 for descriptions of these parameters.

End of procedure

Note: When a SIP Server HA pair is configured as a gateway resource for the Resource Manager, the `sip-address` configuration option in the T-Server section of both SIP Servers in the pair must be configured to point to the virtual-ip for the HA pair. This value is used by SIP Server to build the `Via` and the `Contact` headers in SIP messages.

Procedure:
Creating the resource access point for Recording Server

Purpose: To provide an overview of the steps to configure logical recording server `Application`, to provide a presence for the recording server in the Genesys Configuration Layer.

Summary

You must create a separate `Application` for each recording server in your deployment. The `Application` type is `Resource Access Point`.

For detailed information about importing `Application Templates` and metadata, and creating `Applications` from the templates, see Appendix A in the *Genesys Voice Platform 8.1 Deployment Guide,* which describes the pre-installation activities

Prerequisites

- You are logged in to Genesys Administrator. To access Genesys Administrator, go to the following URL:
  `http://<Genesys Administrator host>/wcm`
- The Resource Manager Installation Package (IP) is available.

Start of procedure

1. Create the Recording Server `Application` object.
   a. Import the required `Application Template` from the Resource Manager Installation Package (IP). For example, `VP_CallRecordingServer_814.apd`.
   b. On the `Provisioning > Environment > Applications` tab, create and name the new `Resource Access Point`, based on the `Application Template`.

Configure Resource Access Points:

2. In the `gvp.rm` section, on the `Provisioning > Environment > Applications > <Recording Server> > Options` tab, configure the following options:
   - `aor=sip[s]:<host|ip>:<port>`
   - `port-capacity=500`
   - `redundancy-type=active`

Note: The host and port number in the aor configuration option, is populated automatically when the LRG group wizard is used to create the `recordingserver` resource group.

3. In the `provision` section, ensure the default value of `1` is retained for the `recording-server` configuration option.
4. Save the configuration.

End of procedure

Next Steps

- No further steps are required.

Chapter

5 Configuring Policy Server

The Genesys Voice Platform (GVP) Policy Server component is used by Genesys Administrator for the validation and resolution of GVP-specific business rules. It is a stand-alone Java process that connects to Management Framework through an HTTP interface.

This chapter provides information about configuring Policy Server in the following sections:

Task Summary: Configuring Policy Server

Task Summary: Configuring Policy Server summarizes the tasks that are required to implement Policy Server functionality in your GVP deployment.

Task Summary: Configuring Policy Server

Objective	Related Procedures and Actions
Create a connection to Policy Server in the Configuration Manager Application.	In Genesys Administrator, open the Configuration Manager `Application` and on the `Configuration` tab, add Policy Server to the `Connections`. **Note:** Policy Server can be deployed with Genesys Administrator and Configuration Manager 8.1 and later releases only.
Customize client side communication ports.	See "Configuring Client-Side Connections" on page 72.

Important Policy Server Configuration Options

This section describes the key configuration options that you either must or may want to customize.

Configure the options on Genesys Administrator on the `Provisioning > Environment > Applications > <Policy Server> > Options` tab. For the detailed steps to configure option settings, see Procedure: Viewing or modifying GVP configuration parameters, on page 34.

The configurable Policy Server parameters are in the following configuration sections:

- `https`—Parameters that determine how security is implemented for Policy Server.
- `https_key`—Parameters that specifies the optional security key password.
- `log` —Parameters that determine the logging behavior.
- `reporting`—Parameters that determines how data from Policy Server is reported.

Table 15 provides information about important Policy Server parameters that are not described in Chapter 3 on page 41. Table 15 provides parameter descriptions as well as the default parameter values that are preconfigured in the Policy Server `Application` object.

Unless indicated otherwise, all changes take effect immediately.

For a complete list of Policy Server configuration options and their descriptions, see the *Genesys Voice Platform 8.1 Configuration Options Reference.*

Table 15: Selected Policy Server Configuration Options

Option Name	Description	Valid Values and Syntax
	`https` Section	
SSL Keystore Path	Specifies the path to the keystore file, which will be used for all the HTTPS connectors.	Any string of characters. **Default value:** `${user.home}/.keystore` Changes take effect: start/restart
SSL Keystore Password	Specifies the password for the keystore file.	Any string of characters. **Default value:** Empty Changes take effect: start/restart

Table 15: Selected Policy Server Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
SSL Keystore Type	Specifies the type of keystore, which defines the supported file format for the security implementation.	Any string of characters. **Default value:** `JKS` Changes take effect: start/restart
SSL Certificate Algorithm	Specifies the name of the SSL algorithm that will be used for the configured keystore.	Any string of characters. **Default value:** `SunX509` Changes take effect: start/restart
HTTPS Protocol	Specifies the cryptographic protocol to use.	Select one of five option values—`SSL`, `SSLv2`, `SSLv3`, `TLS`, or `TLSv1` **Default value:** `TLS` Changes take effect: start/restart
Secure Random Algorithm	Specifies the name of the RNG (Random Number Generator) algorithm. For more information about the RNG, see the JDK JavaDoc for class `java.security.SecureRandom`	Any string of characters. **Default value:** Empty Changes take effect: start/restart
Security Provider	Specifies the name of Java security provider. For more information about the security provider, see the JDK JavaDoc for class `java.security.Provider`	Any string of characters. **Default value:** `None` Changes take effect: start/restart
Client Authentication Requirement	Specifies the HTTPS client authentication requirements. If this option is set to: • `none`—No certificate is requested; Client-side authentication is disabled. • `required`—A certificate is requested and the server will require a valid, non-empty certificate response to establish the connection. (Works for BIO connector type only.) • `preferred`—A certificate is requested, but the server will still establish the connection if the certificate response is empty.	Select one of three option values—`none`, `required`, or `preferred`. **Default value:** Empty Changes take effect: start/restart

Table 15: Selected Policy Server Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
HTTPS Connector Type	Specifies the type of Jetty connector that will be used. If this option is set to: • 1 <NIO>—Non-blocking NIO connector. • 2 <BIO>—Blocking BIO connector. For more information about these connectors, see Jetty's JavaDoc for class `org.mortbay.jetty.security.SslSelectChannelConnector`.	Select one of two option values—`1` or `2`. **Default value:** `2` Changes take effect: start/restart
`https_key` Section		
SSL Key Password	Specifies the optional key password for the HTTPS configuration.	Any string of characters. **Default value:** Empty Changes take effect: start/restart
`log Section`		
Verbose Level	Determines whether or not a log output is created. If it is, this option specifies the minimum level of log events that are generated. Any one of the following log event levels can be selected as the value for this option (starting with the highest priority level): `standard`, `interaction`, `trace`, `debug`, `all`, or `none`. For a description of the log events that are logged for each level, see Table 8 on page 68.	Select one of several log event levels. **Default value:** `standard`
Output for Level All	Specifies the outputs to which an application sends all log events. The log output types must be separated by a comma when more than one output is configured.	A string of characters. **Default value:** `Empty`
Output for Level Standard	Specifies the outputs to which an application sends the log events of the Standard level.	A string of characters. **Default value:** `stdout`

Table 15: Selected Policy Server Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Output for Level Interaction	Specifies the outputs to which an application sends the log events of the Interaction level and higher, which means, more than one output is configured—`standard` and `interaction` levels.	A string of characters. **Default value:** `Empty`
Output for Level Trace	Specifies the outputs to which an application sends the log events of the Trace level and higher, which means, more than one output is configured—`standard`, `interaction`, and `trace` levels.	A string of characters. **Default value:** `Empty`
Output for Level Debug	Specifies the outputs to which an application sends the log events of the Debug level and higher, which means, more than one output is configured—`standard`, `interaction`, `trace`, and `debug` levels.	A string of characters. **Default value:** `logs/ps.log`
Log Segmentation	Specifies the segmentation limit for a log file. Sets the mode of measurement, along with the maximum size. If the current log segment exceeds the size set by this option, the file is closed and a new one is created. For a complete description of the option values for log segmentation, see Table 8 on page 68.	A string of characters. **Default value:** `10MB`
Log Expiration	Determines whether or not the log files expires. If they do, this option sets the measurement for determining when they expire, along with the maximum number of files (segments) or days before the files are removed. For a complete description of the option values for log expiration, see Table 8 on page 68.	A string of characters. **Default value:** `false`
Log Messages Format	Specifies the format of log record headers that an application uses when writing logs in the log file. Using compressed log record headers improves application performance and reduces the log file's size. For a complete description of each option value, see Table 8 on page 68.	Select one of two option values—`short` or `full`. **Default value:** `full`

Table 15: Selected Policy Server Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Time Format for Log Messages	Specifies how to represent the time when an application generates log records in a log file. For a complete description of each option value, see Table 8 on page 68.	Select one of three option values—`time`, `locale`, or `ISO8601`. **Default value:** `time`
	`reporting` **Section**	
PS Service IP Address	Specifies the interface IP address that will be used to bind the Policy Server service.	A string of characters. **Default value:** Empty Changes take effect: start/restart
PS Service Hostname	Specifies the hostname that will be used to access Policy Server.	A string of characters. **Default value:** Empty Changes take effect: start/restart
PS Service Port	Specifies the port on which Policy Server will receive reporting requests.	An integer greater than 0. **Default value:** `8090` Changes take effect: start/restart
PS Service Protocol	Specifies the type of communication protocol that Policy Server will use to service reporting requests.	Select one of two option values—`http` or `https`. **Default value:** `http` Changes take effect: start/restart
Basic HTTP Authentication User Name	Specifies the user name that Policy Server uses to perform basic HTTP authentication.	A string of characters. **Default value:** Empty Changes take effect: start/restart
Basic HTTP Authentication Password	Specifies the password that Policy Server uses to perform basic HTTP authentication.	A string of characters. **Default value:** Empty Changes take effect: start/restart

Chapter

6 Provisioning IVR Profiles

IVR Profiles are the Voice Extensible Markup Language (VoiceXML), Call Control Extensible Markup Language (CCXML), Conference, and Announcement applications that control interactions with external customers. This chapter describes how to provision IVR Profiles for Genesys Voice Platform (GVP). It contains the following sections:

Provisioning IVR Profiles for GVP

The summary procedure in this section provides an overview of the steps to provision IVR Profiles for GVP.

Procedure:
Provisioning IVR Profiles

Purpose: To set up GVP so that it uses specified VoiceXML or CCXML applications.

Prerequisites

- You are logged in to Genesys Administrator. To access Genesys Administrator, go to the following URL:

  `http://<Genesys Administrator host>/wcm`

Start of procedure

1. Use the IVR Profile Wizard in Genesys Administrator to create or edit an IVR Profile object. For more information on how to use the wizard, see the *Genesys Voice Platform 8.1 Deployment Guide*.
2. On the `Provisioning > Voice Platform > IVR Profile` tab of Genesys Administrator, launch the IVR Profile Wizard.
3. On the `Service Type` page:
   a. Enter the name of the IVR Profile.
   b. Select the type of service that the IVR Profile requires. This sets the `gvp.general.service-type` parameter. The possible values are:
      - `VoiceXML`
      - `CCXML`
      - `Conference`
      - `Announcement`

   c. Complete the remainder of the wizard as applicable for your deployment. Depending on the type of service you have selected in Step b, the wizard presents different subsequent pages.
      - For VoiceXML, go to Step 4 on page 108.
      - For CCXML, go to Step 5 on page 110.
      - For Conference, go to Step 6 on page 110.
      - For Announcement, go to Step 7 on page 111.
4. For VoiceXML:
   a. On the `Service Properties` page:
      i. Enter the Initial Page URL. This sets the `initial-page-url` parameter.
      ii. Enter the Alternate Page URL. This sets the `alternatevoicexml` parameter.
      iii. Enter the Default Properties Page URL. This sets the `default-properties-page` parameter.
      iv. Select the type of VoiceXML interpreter. This sets the `voicexml.gvp.appmodule` parameter. For more information on the interpreters, see Chapter 7, "Configuring the Media Control Platform," on page 145.
      v. Enter the Toll Free Number. This sets the `toll-free-number` parameter.

   vi. Enter the Virtual Reporting Object 1 and Virtual Reporting Object 2. This sets the `VirtualReportingTag1`, and `VirtualReportingTag2` parameters.

b. On the `Usage Limits` page, if required, enter the maximum number of concurrent sessions. This sets the `usage-limits` parameter.

c. On the `IVR Capabilities` page:

   i. If you want outbound calls with bridged or consultation transfers, allow outbound calls. This sets the `outbound-call-allowed` parameter.

   ii. If you want midcall transfers (blind or consultation), allow transfers. This sets the `transfer-allowed` parameter.

   iii. If MSML requests are not to be processed, set `msml-allowed` to `false`.

   iv. Select how to use the gateway. This sets the `use-same-gateway` parameter.

d. On the `CTI Parameters` page:

   i. If CTI Connector is required to interface with IVR Server, select `Require CTI Interaction`. This sets the `cti-allowed` parameter.

   ii. If midcall blind transfers (starts in the application) are allowed (`transfer-allowed` is set to `true`) and you want the transfers to be performed through the IVR Server, select `Transfer on CTI`. This sets the `cti.TransferOnCTI` parameter.

   iii. Enter the default agent. This must be a DN that is configured in the Configuration Server database. This sets the `cti.DefaultAgent` parameter.

e. If you allowed outbound calls and/or transfers, on the `Dialing Rules` page:

   i. Select to accept or reject the rule expression.

   ii. Enter the Regular Expression.

   iii. Click `Add as New Rule`.

Repeat Steps i to iii for each Dialing Rule that you require. This sets the `gvp.policy.dialing-rules` parameter.

For more information on Dialing Rules, see the “`gvp.policy.dialing-rules` Section” on page 123.

f. On the `Policies` page, add the SQ Notification Threshold. This sets the `error.notification.threshold` parameter.

For more information on VoiceXML IVR Profile parameters, see Table 16 on page 113.

5. For CCXML:
   a. On the `Service Properties` page:
      i. Enter the Initial Page URL. This sets the `initial-page-url` parameter.
      ii. Enter the Virtual Reporting Object 1 and Virtual Reporting Object 2. This sets the `VirtualReportingTag1`, and `VirtualReportingTag2` parameters.
   b. On the `Usage Limits` page, if required:
      i. Enter the maximum number of concurrent sessions. This sets the `usage-limits` parameter.
      ii. Enter the usage limits for each service. This sets the `<service>-usage-limit` parameter.
      iii. Enter the usage limits per session. This sets the `<service>-usage-limit-per-session` parameter.
   c. On the `IVR Capabilities` page, if required:
      i. Enable conferencing. This sets the `conference-allowed` parameter.
      ii. Enable outbound calling. This sets the `outbound-call-allowed` parameter.
      iii. Enable transfers. This sets the `transfer-allowed` parameter.
      iv. Enable VoiceXML dialogs. This sets the `voicexml-dialog-allowed` parameter.
      v. Select how to use the gateway. This sets the `use-same-gateway` parameter.
   d. If you allowed outbound calls and/or transfers, on the `Dialing Rules` page:
      i. Select to accept or reject the rule expression.
      ii. Enter the Regular Expression.
      iii. Click `Add as New Rule`.

   Repeat Steps i to iii for each Dialing Rule required. This sets the `gvp.policy.dialing-rules` parameter.

   For more information on Dialing Rules, see “`gvp.policy.dialing-rules` Section” on page 123.

   e. On the `Policies` page, add the SQ Notification Threshold. This sets the `error.notification.threshold` parameter.

   For more information on CCXML IVR Profile parameters, see Table 16 on page 113.

6. For Conference:
   a. On the `Service Properties` page:
      i. Enter the conference ID. This sets the `conference-id` parameter.
      ii. Enter the maximum conference size, if required. This sets the `application-confmaxsize`.

iii. Enter the Virtual Reporting Object 1 and Virtual Reporting Object 2. This sets the `VirtualReportingTag1`, and `VirtualReportingTag2` parameters.

b. On the `Usage Limits` page, if required, enter the maximum number of concurrent sessions. This sets the `usage-limits` parameter.

c. On the Policies page, add the SQ Notification Threshold. This sets the `error.notification.threshold` parameter.

For more information on Conference IVR Profile parameters, see Table 16 on page 113.

7. For Announcement:

a. On the `Service Properties` page;

i. Enter the `VoiceXML` page that is used to play announcements. This sets the `announcement-url` parameter.

ii. Enter the content-type, if required.

iii. Enter the repeat count, if required.

iv. Enter the delay amount, if required.

v. Enter the duration, if required.

vi. Enter the Virtual Reporting Object 1 and Virtual Reporting Object 2. This sets the `VirtualReportingTag1`, and `VirtualReportingTag2` parameters.

b. On the `Usage Limits` page, if required, enter the maximum number of concurrent sessions. This sets the `usage-limits` parameter.

c. On the Policies page, add the SQ Notification Threshold. This sets the `error.notification.threshold` parameter.

For more information on Announcement IVR Profile parameters, see Table 16 on page 113.

8. On the `Context Services` page, enter a username and password for context services authentication.

The password is masked when it is entered into the password field.

9. When you have completed the required configuration in the wizard, click `Finish`.

The IVR Profile object now displays in the list on the `Provisioning > Voice Platform > IVR Profile` tab in Genesys Administrator.

10. Modify the IVR Profile to capture the required configuration parameters that are not set with the wizard.

    The Resource Manager uses the IVR Profile name that you specify to identify the context of the session. For more information, see "Application Identifiers" on page 27.

    For detailed information about the IVR Profile configuration options, see "IVR Profile Configuration Options" on page 113.

    For detailed information about configuring IVR Profiles for GVPi, see "IVR Profile Configuration for GVPi" on page 127.

End of procedure

IVR Profile Configuration Options

The IVR Profile configuration options determine the type of service the IVR Profile will provide, and also its operating parameters.

Configure the IVR Profile parameters in Genesys Administrator:

- Configure the parameters in the `gvp.service-parameters` section and, if applicable, in the `dbmp` section on the `Provisioning > Voice Platform > IVR Profile > <IVR Profile> > Options` tab.
- Configure all the other IVR Profile parameters on the `Provisioning > Voice Platform > IVR Profile > <IVR Profile> > Option` tab.

For more information about using Genesys Administrator to add or modify configuration sections and options, see Procedure: Viewing or modifying GVP configuration parameters, on page 34.

Table 16 describes the IVR Profile configuration options.

Notes: All changes to IVR Profile configuration options take effect with the next session that uses the IVR Profile.

The alarm and response codes are not independent policies. Resource Manager will look for the corresponding alarm and/or response code parameters from the matched tenant/profile only.

Table 16: IVR Profile Configuration Options

Option Name	Description	Valid Values and Syntax
	`gvp.general` **Section**	
Conf Max Size	(For Conference only) The maximum number of participants in the conference. This setting does not override conference size maximums that are configured for the Resource Manager logical group or the conference resource itself.	Any unsigned integer. **Default value:** `20`
Service Type	The default type of service that the IVR Profile provides. The default service type does not preclude the use of other service types within the application as well.	• `ccxml` • `conference` • `voicexml` • `announcement` **Default value:** Empty
Toll Free Number	The toll free number that is used by this IVR Profile.	A string of characters. **Default value:** Empty

Table 16: IVR Profile Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
SIP Session Timer Interval	The timeout value, in seconds, for the SIP session that it executed for this IVR Profile. If the Resource Manager receives no SIP messages associated with this call leg within the timeout interval, it considers the call leg to have ended. For the call leg that is associated with this IVR Profile, the value of this `sip.sessiontimer` parameter overrides session expiry timeouts that are set at the level of the tenant, the Resource Manager, and the resource.	Any unsigned integer. **Default value:** Empty
Virtual Reporting Tag 1 Virtual Reporting Tag 2	Specifies the Virtual Reporting Object 1 and Virtual Reporting Object 2. These parameters enable you to query and correlate call data with custom parameters based on business needs. For example, if you are running a certain campaign, you may want to associate calls with a virtual reporting object value of "Last Campaign 2009" and later query call data by that, not having to deal with which DIDs, IVR Profiles or platform instances were utilized for that campaign.	A string of characters. **Default value:** Empty
	`gvp.log` Section	
metricsfilter	The filter that determines which metrics (for calls that are made to this IVR Profile) will be forwarded to the Reporting Server. If this parameter is set, the value will override the default `DATAC` filter for the component for sessions that execute under this IVR Profile. This overrides the default parameter that is set at the platform level. The Resource Manager passes this property value to the component in a SIP custom header.	`<FilterID1>[,<FilterID2>, ...]` where: • `<FilterID>` is a single Metric ID or a range of Metric IDs. For the valid Metric IDs, see the *Genesys Voice Platform 8.1 Metrics Reference.* The wildcard character (`*`) means "all". **Default value:** Empty

Table 16: IVR Profile Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
	`gvp.policy` **Section**	
<Service> Allowed	Specifies whether a Resource Manager session is allowed to use the selected service. Valid values for <Service> are: • `Announcement` • `ccxml` • `Conference` • `cpd` • `MSML` • `Outbound Call` • `Treatment` • `VoiceXML Dialog`	• `True` • `False` **Default value:** `True`
CTI Allowed	Specifies whether calls that use this IVR Profile use CTI Connector to interface with IVR Server, or send directly to Media Control Platform (MCP) or Call Control Platform (CCP).	• `True`—Use CTI Connector • `False`—Use MCP or CCP **Default value:** `True`
Transfer Allowed	Specifies whether a Resource Manager session is allowed to perform a transfer by using a SIP `REFER` request within the existing SIP session.	• `True` • `False` **Default value:** `True`
Allow Burst Usage.	Specifies whether burst usage for an application is allowed for various usage-based policies. **Note:** This parameter applies to individual objects (tenant/profile) only; it's not applicable for hierarchical use.	• `True` • `False` **Default value:** `False`
Raise Alarm for Exceeding Burst Limit	Specifies whether to raise an alarm when the burst limit has been exceeded.	• `True` • `False` **Default value:** `False`
Dialing Rule Based Rejection Response Code	The SIP response code that is sent in the SIP response when a call is rejected because of a dialing rule (`gvp.policy.dialing-rules.rule-<n>`).	• `<sipcode>;<desc>` • `<sipcode>` where: • `<sipcode>` is an integer in the range of 400–699. • `<desc>` is any string. **Default value:** `403`

Table 16: IVR Profile Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Raise Alarm for Dialing Rule Based Rejection	Specifies whether an alarm will be raised for the corresponding policy violation.	• `True` • `False` **Default value:** `False`
Usage Limits Note: GVP has the ability to set policy limits for Level 1, Level 2 and Level 3 simultaneous use of specific services, and also usage limit alarms as well.		
Usage Limits	The number of times that a Resource Manager session can be concurrently in use in the context of any IVR Profile.	Any unsigned integer. **Default value:** Empty
<service> level2 Usage Limit (burst limit)	Specifies the number of times a Resource Manager session can concurrently be in use in the context of the IVR Profile for level2 burst. Valid values for `<service>` are: • `announcement` • `ccxml` • `conference` • `inbound` • `msml` • `outbound` • `voicexml`	Any integer **Default value:** Empty
<service> level3 Usage Limit (burst limit)	Specifies the number of times a Resource Manager session can concurrently be in use in the context of the IVR Profile for level3 burst. Valid values for `<service>` are: • `announcement` • `ccxml` • `conference` • `inbound` • `msml` • `outbound` • `voicexml`	Any integer **Default value:** Empty

Table 16: IVR Profile Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
<service> Usage Limit	The maximum number of concurrent service sessions that are permitted for this IVR Profile Valid values for `<service>` are: • `announcement` • `ccxml` • `conference` • `inbound` • `msml` • `outbound` • `voicexml`	Any unsigned integer. **Default value:** Empty
<service> Usage Limit per session	The number of times that the specified service may be invoked in the context of this instance of a Resource Manager session. Valid values for `<service>` are: • `announcement` • `ccxml` • `conference` • `msml` • `voicexml`	Any unsigned integer. **Default value:** Empty
<service> Usage Limit Exceeded Response Code	The SIP response code that is sent in the SIP response when a request for a service is rejected because the usage limits for that service (`gvp.policy.<service>-usage-limit` or `gvp.policy.<service>-usage-limit-per-session`) have been reached. Valid values for `<service>` are: • `ccxml` • `conference` • `voicexml` • `outbound` • `inbound` • `announcement` • `msml`	• `<sipcode>;<desc>` • `<sipcode>` where: • `<sipcode>` is an integer in the range of 400–699. • `<desc>` is any string. **Default value:** `503`

Table 16: IVR Profile Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Usage Limit Exceeded Response Code	The SIP response code that is sent in the SIP response when a call is rejected because the usage limits that are specified in the following configuration options have been reached: • `gvp.policy.usage-limits` • `gvp.policy.outbound-usage-limit` • `gvp.policy.inbound-usage-limit`	• `<sipcode>;<desc>` • `<sipcode>` where: • `<sipcode>` is an integer in the range of 400–699. • `<desc>` is any string. **Default value:** `480` (Temporarily unavailable)
Raise Alarm for <service> Not Allowed	Specifies whether an alarm will be raised for the corresponding policy violation. Valid values for `<service>` are: • `ccxml` • `conference` • `outbound-call` • `transfer` • `voicexml-dialog` • `announcement` • `msml`	• `True` • `False` **Default value:** `False`
Raise Alarm for <service> Usage Limit Exceeded	Specifies whether an alarm will be raised for the corresponding policy violation. Valid values for `<service>` are: • `ccxml` • `conference` • `inbound` • `outbound` • `voicexml` • `announcement` • `msml`	• `True` • `False` **Default value:** `False`

Table 16: IVR Profile Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax

<service> Forbidden Response Code	Specifies the error response code to send when the `<service>` is not allowed. Valid values for `<service>` are:
	• `ccxml`
	• `conference`
	• `inbound`
	• `outbound`
	• `voicexml`
	• `announcement`
	• `msml`
		`<sipcode>;description`
		or
		`<sipcode>`
		Default value: `403`
		For more information on SIP Response Codes, see Appendix D, “SIP Response Codes,” on page 469.

Disable Video	Specifies whether to disable video.	• `True`—Disable video • `False`—Enable video **Default value:** `False`
Disable <Codec>	Specifies whether to disable <Codec> transcoding. The values for <Codec> are: • G729 • AMR-NB • AMR-WB	• `True`—Disable transcoding • `False`—Enable transcoding **Default value:** `False`

Table 16: IVR Profile Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
mcp-asr-usage-mode	Specifies whether there will be one Voice Resource Management (VRM) session for the entire call, or whether a separate VRM session will be opened for each recognition request. A single session for the entire call (`mcp-asr-usage-mode` = `per-call`) means that each call may have multiple recognition sessions. If this parameter is set not to enable a single session for the entire call (`mcp-asr-usage-mode` = `per-utterance`), each VRM session is closed when the recognition request completes, either successfully or unsuccessfully (such as no match). Therefore, each call may have multiple VRM sessions. The Resource Manager passes this value to the Media Control Platform in a `Request-URI` parameter. The value of this parameter overrides a similar parameter that is set for the Media Control Platform overall (`asr.load_once_per_call`), if the settings are not consistent. See the description of the `asr.load_once_per_call` parameter on page 160 for more information about the implications of this setting.	• `per-call` • `per-utterance` **Default value:** `per-call`
MCP Send/Receive Enabled	Specifies whether a Media Control Platform is allowed to perform `<send>` and `<receive>` extensions. The Resource Manager passes this value to the Media Control Platform in a `Request-URI` parameter.	• `True` • `False` **Default value:** `True`
MCP max-subdialog-depth	Specifies the number of subdialogs allowed in a VoiceXML call.	An unsigned integer. **Default value:** Empty

Table 16: IVR Profile Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax

<service> Capability Requirement	A list of name-value pairs that specify the capabilities that are required when the specified `<service>` is invoked in the context of this IVR Profile.
	Valid values for `<service>` are:
	• `ccxml`
	• `conference`
	• `voicexml`
	• `msml`
	• `announcement`
	Items in the name-value pair list are separated by a semicolon (;). The value side of each name-value pair can itself be a comma-separated list of capabilities. Each set of values must be unique.
	The Resource Manager will direct interactions to a resource group only if the resource group capabilities exactly match the capability requirements that are specified in this option (see the `<Logical Group>.capability` option in Table 14 on page 96).
		`<cap_NameA>=<cap_ValueA>[;<cap_NameB>=<cap_ValueB>;...]`
		where:
		• `<cap_NameX>` is the name of the capability.
		• `<cap_ValueX>` is a comma-separated list of values.
		Example:
		`lang=en-US;grammar=grxml,gsl`
		Default value: Empty

Table 16: IVR Profile Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Use Same Gateway	(For `gateway` service only) Specifies whether outbound calls to a gateway must use the same gateway that the Resource Manager session is currently using. A gateway resource becomes associated with a Resource Manager session when (a) the Resource Manager session is not already associated with another gateway resource and (b) one of the following occurs: • The Resource Manager receives a request from a gateway resource. • The Resource Manager receives a request for a gateway service and allocates it in accordance with the load-balancing scheme for the group.	• `always`—The Resource Manager must forward the request to exactly the same gateway resource already associated with the Resource Manager session, or else the request fails. • `preferred`—The Resource Manager first tries to forward the request to the gateway resource already associated with the Resource Manager session, but tries other gateways if the first request fails. • `indifferent`—The Resource Manager chooses a gateway in accordance with load-balancing scheme for the group. **Default value:** `always`
Prediction Factor	Specifies the ratio of agent calls to customer calls from Outbound Contact Server (OCS) for a campaign to minimize bridging when multiple MCPs are present in the environment.	Any integer range between 0.33—1.0. **Default value:** `0.5`

Table 16: IVR Profile Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
	`gvp.policy.dialing-rules` **Section**	

rule-<n>	For each `<n>`, this parameter specifies a dialing rule that the Resource Manager will use to determine if an address towards a gateway is allowed.
	`<n>` is a positive integer in the range of 1–10000.
	The rules are applied in rule number order.
	Example:
	To reject outbound calls to 911, allow calls to toll-free numbers, and reject calls to long-distance numbers, specify the following set of rules:
	`gvp.policy.dialing-rules.rule-1: r,911`
	`gvp.policy.dialing-rules.rule-2: a,1800*`
	`gvp.policy.dialing-rules.rule-3: a,1888*`
	`gvp.policy.dialing-rules.rule-4: a,1877*`
	`gvp.policy.dialing-rules.rule-5: a,1866*`
	`gvp.policy.dialing-rules.rule-6: r,1*`
	Note: Resource Manager will search to match a dialed number with the rejected expressions first. Then, it will then search to match the dialed number with the accepted expressions. If not found, the dialed number is accepted.
		`<rule-type>;<regex>`
		where:
		• `<rule-type>` is either `a` or `r`, where `a = allow` and `r = reject`.
		• `<regex>` is a regular expression.
		Default value: Empty

Table 16: IVR Profile Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
	`gvp.policy.call-info` **Section**	

rule-<n>	For each `<n>`, this parameter specifies a set of DNIS, ANI, and User-Agent rules as obtained from the SIP message.
	`<n>` is a positive integer in the range of 1–100.
	The rules are applied in rule number order. When a rule is matched, subsequent rules are ignored.
	Notes:
	These rules apply only to inbound calls.
	DNIS and User-Agent rules are applicable for the Tenant.
	ANI based rules are applicable for the IVR Profile.
	Example:
	`rule-1 = r;ani;408666$;503`
	`rule-2 = s;dnis;^[0-9]234$;voicexml,http://www.gvp.com/play-error.vxml.`
		`<type>;<entity>;<regex>;<action>`
		where:
		• `<type>` is either `a`, `r`, or `s`, where:
		`a` = accept.
		`r` = reject.
		`s` = script play.
		• `<entity>` is either `ani`, `dnis`, or `ua`, where:
		`ani` = ANI.
		`dnis` = DNIS.
		`ua` = SIP User Agent.
		• `<regex>` is a regular expression.
		• `<action>` is set based on the `<type>` value, as follows. If:
		`<type>` = `a`, `<action>` is empty
		`<type>` = `r`, `<action>` is `<response-code>,<response-text>`
		`<type>` = `s`, `<action>` is `<service-type>,<url>`
		Default value: Empty

	`gvp.policy.speech-resources` **Section**	
ASR default engine	Specifies the default ASR engine to use. **Example:** `Nuance?MRCPv2`	`<vendor>?Protocol` where: • `<vendor>` is a supported ASR Vendor. • `Protocol` is either `MRCPv1` or `MRCPv2` **Default value:** Empty

Table 16: IVR Profile Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
TTS default engine	Specifies the default TTS engine to use. **Example:** `Realspeak?MRCPv2`	`<vendor>?Protocol` where: • `<vendor>` is a supported TTS Vendor. • `Protocol` is either `MRCPv1` or `MRCPv2` **Default value:** Empty
ASR/TTS default language	Specifies the default ASR or TTS language.	An alphanumeric string. **Default value:** Empty
`gvp.service-parameters` **Section**		
<service>.<param-name>	Each parameter that you create in this section takes the form of a pair of strings that determine whether a `Request-URI` parameter called `<param-name>`, with a value specified in `<value>`, will be included in forwarded SIP requests. Valid values for `<service>` are: • `announcement` • `ccxml` • `conference` • `cti` • `gateway` • `voicexml` The Resource Manager will apply this parameter to a SIP request only if the specified `<service>` is invoked by the SIP request. • Setting the `<value-type>` to `undefined` deletes the `<param-name>` parameter from the incoming SIP request. • Setting the `<value-type>` to `fixed` overrides the `<param-name>` parameter value in the incoming SIP request. • Setting the `<value-type>` to `default` provides a default value for the `<param-name>` parameter in the outgoing SIP request, if the `<param-name>` parameter does not already exist.	`<value-type>, <value>` where `<value>` is any string and `<value-type>` is: • `undefined`—The SIP `Request-URI` parameter with the specified `<param-name>` will not be in the forwarded request (even if the parameter was already in the incoming request). • `fixed`—The parameter `<param-name>=<value>` will be in the SIP `Request-URI`. • `default`—If the SIP `Request-URI` parameter with name `<param-name>` already exists, it will be left unmodified in the SIP `Request-URI`, but if the incoming request does not already include the parameter, the parameter `<param-name>=<value>` will be added to the SIP `Request-URI`.

Table 16: IVR Profile Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
	`gvp.service-prerequisite` **Section**	
Alternate VoiceXML URL	(For `voicexml` service only) The URL to an alternative initial page that the Media Control Platform will use if the request to the `Initial Page URL` fails. Before it forwards the service request, the Resource Manager inserts this information as the value of the `gvp.alternatevoicexml` SIP parameter.	Any valid URL. **Default value:** Empty
Announcement URL	(Mandatory for announcement service) Specifies the play parameter.	Any valid URL. **Default value:** Empty
Conference Id	(Mandatory for conference service) The conference identifier. Before it forwards the service request, the Resource Manager replaces the user part of the SIP `Request-URI` with `conf=<conference-id>`. The Resource Manager uses the `conference-id` to ensure that it routes all requests for the same conference to the same conference resource, even if the requests originate from different Resource Manager sessions.	Any alphanumeric string, without spaces. **Default value:** Empty
Default Properties Page	(For `voicexml` service only) The URL to a page that contains the default properties and handlers. Before it forwards the service request, the Resource Manager inserts this information as the value of the `gvp.defaultsvxml` SIP parameter.	Any valid URL. **Default value:** Empty
Initial Page URL	(Mandatory for `voicexml` and `ccxml` services) The URL of the initial page that is to be invoked. Before it forwards the service request, the Resource Manager inserts this information as the value of the `voicexml` or `ccxml` SIP parameter.	Any valid URL. **Default value:** Empty
	`gvp.context-services-authentication` **Section**	
Context Service Username	Specifies the username that will be used for context services authentication.	An alphanumeric string. **Default value:** Empty
Context Service Password	Specifies the password that will be used for context services authentication.	An alphanumeric string. **Default value:** Empty

Table 16: IVR Profile Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
OPM Section		
`transaction object ID`	Specifies the transaction or list object DBID that will be referenced during runtime of this profile.	Any string of integers (except `0` [zero]). **Default value:** Empty

Operational Parameter Management and Self-Service Applications

Genesys offers a means of controlling the behavior (logic) of routing and VoiceXML applications, using a common method called Operational Parameter Management (OPM). Through a web interface, a customer, business manager, or any third party who has access permission, can modify the value of a set of custom parameters. When the routing or VoiceXML application executes in servicing a call, its behavior can be changed based on OPM parameter values. The parameters, possible values and logic to handle them is designed by the application developer (routing strategy or GVP VoiceXML application), and the set of parameters used within an application(s) are configured for end-user access, and eventually accessed, using the Genesys Administrator Extension (GAX) interface.

An end-user—a managed services administrator, a business marketing person, or even a hosted services customer—can be given security access to specific OPM parameter set(s) via the web. Using dropdown menus, data entry, or radio buttons, the end-user can change the behavior of an application to match their needs without ever knowing how the application was created or where it is executed.

Once configured in GAX, an OPM set can be assigned to a routing strategy or configured within a GVP IVR Profile, or even both. GVP supports up to two parameter sets per IVR Profile (application) and is configured as a setting in the IVR Profile. When the VoiceXML application executes, the GVP interpreter fetches the named parameter set for use as operational variables within the application.

IVR Profile Configuration for GVPi

Configure the IVR Profile parameters in Genesys Administrator:

- Configure the parameters in the `gvp.service-parameters` section and, if applicable, in the `dbmp` section on the `Provisioning > Voice Platform > IVR Profile > <IVR Profile> > Options` tab.

- Configure all the other IVR Profile parameters on the `Provisioning > Voice Platform > IVR Profile > <IVR Profile> > Option` tab.

For more information about using Genesys Administrator to add or modify configuration sections and options, see Procedure: Viewing or modifying GVP configuration parameters, on page 34.

Table 17 section describes the specific IVR Profile configuration options for GVPi.

Note: All changes to IVR Profile configuration options take effect with the next session that uses the IVR Profile.

Table 17: IVR Profile Configuration Options for GVPi

Option Name	Description	Valid Values and Syntax
	`gvp.service-parameters` **Section**	
App Module Name	Specifies which interpreter that Media Control Platform is to use.	• `VXML-LGVP`—GVPi • `VXML-NG`—Next Generation **Default value:** `VXML-NG`
Enable Debugging	Specifies whether to enable debugging.	• `True`—Disable debugging • `False`—Enable debugging **Default value:** `False`
ASR Platform Used for Recognition	Specifies the ASR speech engine that is used for voice recognition.	Comma separated string. **Default value:** Empty
Enable Record Utterance	Specifies whether to record utterance files.	• `True`—Record utterance files • `False`—Do not record utterance files **Default value:** `True`
Bad XML Page Hook	Specifies the URL to which unsuccessfully parsed VoiceXML pages are written. **Note:** This option is available only if the `voicexml.gvpi.$adn-flag$` (Enable Debugging) option is set to `1`.	Any URL. **Default value:** Empty

Table 17: IVR Profile Configuration Options for GVPi (Continued)

Option Name	Description	Valid Values and Syntax
Call Trace Hook	Specifies the URL to which call trace information is written. **Note:** This option is available only if the `voicexml.gvpi.$adn-flag$` (Enable Debugging) option is set to 1.	Any URL. **Default value:** Empty
CTI - End Call when Agent Hangs up	Specifies whether to end a call if an agent hangs up when using a route request through IVR Server.	• `True` • `False` **Default value:** `False`
CTI - Reroute Timeout	Specifies how long the platform waits before ending the call if the Agent leg ends without initiating a ReRoute. This is only applicable when `CTI - End Call when Agent Hangs up` is set to `False`.	Any integer range from 3–60. **Default value:** `60`
Debug Hook	Specifies the URL to which debugging information is written. **Note:** This option is available only if the `voicexml.gvpi.$adn-flag$` (Enable Debugging) option is set to 1.	Any URL. **Default value:** Empty
Default Language	The default language for the VoiceXML application.	A string. **Default value:** `en_US`
IVR Timeout	The timeout interval, in seconds, for the initial page url to execute before trying the alternate page url.	An integer range from `0` – `300` **Default value:** `0`
Dial out Number	Specifies the phone number for calls to transfer to if there is an error in the IVR Profile.	Any integer. **Default value:** Empty
CPA Timeout	The timeout interval, in seconds, for call progress analysis.	Any unsigned integer. **Default value:** Empty
Dump Fetched Pages	Specifies whether to move fetched VoiceXML pages to the `<mcp installation>\tmp` folder.	• `True`—Move files to the temp folder • `False`—Do not move files to the temp folder **Default value:** `False`

Table 17: IVR Profile Configuration Options for GVPi (Continued)

Option Name	Description	Valid Values and Syntax
Enable setting application.lastresult $.utterance with DTMF nomatch result	Specifies whether to enable the last utterance with dtmf nomatch when using an on-board DTMF recognizes. If set to `False`, when ASR is disabled, and the application throws a `nomatch`, the `application.lastresult$.utterance` parameter is not populated with invalid digits.	• `True`—Enable • `False`—Disable. **Default value:** `True`
Transfer Option	Specifies the type of SIP transfer. **Note:** If transfer type is set to `1 Signal Channel`, this option must be set to `SipRefer`, `ATTCourtesy, ATTConsultative, or ATTConference`. If transfer type is set to `2 Signal Channel`, this option must be set to `ReferWithReplaces`, or empty.	• `SipRefer` • `ReferWithReplace` • `AttCourtesy` • `ATTConsultative` • `ATTConference` • `AttOOBCourtesy` • `ATTOOBConsultative` • `ATTOOBConference` **Default value:** Not set.
Transfer Type	Specifies whether the transfer that is requested is a blind transfer (one step), or a bridge and consultation transfer (two step).	• `1 Signal Channel`—Blind • `2 Signal Channel`—Bridge and consultation **Default value:** `2 Signal Channel`
Transfer Connect Url	Specifies the script to execute when establishing a transfer using the AT&T switch.	Any URL. **Default value:** Empty
Reclaim Code (ATT only)	Specifies the sequence of DTMF tones to use when removing the caller from hold during an ATTConference transfer, after the conference with the Agent is completed.	Any valid DTMF sequence. **Default value:**`*7`
Transfer Connect	Specifies whether to enable or disable Transfer Connect functionality. Works with Transfer Connect Script. This parameter is used for backward compatibility.	• `True`—Enable • `False`—Disable. **Default value:** `False`
Transfer Connect Script	Specifies the script to use if the Transfer Connect parameter is enabled. This parameter is used for backwards compatibility.	A string of characters. **Default value:** Empty

Table 17: IVR Profile Configuration Options for GVPi (Continued)

Option Name	Description	Valid Values and Syntax
Trap Hook	Specifies the URL for sending SNMP traps.	Any URL. **Default value:** Empty
TTS Vendor	Specifies the Text-to-Speech (TTS) vendor that is used.	Comma separated string. **Default value:** Empty
TTS Gender	Specifies the voice gender that is used for TTS.	• `Male` • `Female` **Default value:** `Male`
`gvp.service-prerequisite` **Section**		
Alternate VoiceXML URL	The URL to an alternative initial page that the Media Control Platform will use if the request to the `Initial Page URL` fails. Before it forwards the service request, the Resource Manager inserts this information as the value of the `gvp.alternatevoicexml` SIP parameter.	Any valid URL. **Default value:** Empty
Initial Page URL	The URL of the initial page that is to be invoked. Before it forwards the service request, the Resource Manager inserts this information as the value of the `voicexml` or `ccxml` SIP parameter.	Any valid URL. **Default value:** Empty

IVR Profile Configuration for Cisco ICM

Table 18 describes the options that must be configured in the IVR Profile to support the CTI Connector and Cisco Intelligent Contact Management (ICM) call flows through NGI.

Table 18: IVR Profile Configuration Options to Support CTIC/ICM

Option Name	Description	Valid Values and Syntax
`gvp.service-parameters` **Section**		
ICM Service ID	A static, unique service ID that indicates the ICM service that is associated with this IVR Profile.	A string of characters. **Default value:** `fixed`

Table 18: IVR Profile Configuration Options to Support CTIC/ICM (Continued)

Option Name	Description	Valid Values and Syntax
Script Mapping	Specifies whether the ICM routing script for the IVR Profile is chosen, based on the Toll-Free Number (TFN) or the DNIS. For example, the ICM script can be specified by using the DNIS. If so, the default value would be a TFN and the applicable values could be either a TFN or the DNIS.	Choose one of two option values: `fixed,TFN` or `fixed,DNIS` **Default value:** `fixed,TFN`
Use Bridge Transfer	When enabled, specifies that a `BRIDGE` transfer will be invoked by the CTI Connector to connect the caller to agent. When disabled, (by default) a `BLIND` transfer is triggered. This feature is applicable for Service Control Interface only, when the `CONNECT` message is received with `TransferHint` flag set to `false`.	Choose one of two option values: `fixed, TRUE` or `fixed, FALSE`. **Default value:** `fixed,TRUE`
CTI Default Agent Number	Specifies the default agent number to which the CTI Connector will send a transfer (to an agent) if an ICM `CONNECT` message is sent with the label type set to `DEFAULT`.	A string of characters. **Default value:** `fixed`
	`gvp.general` **Section**	
Toll Free Number	A unique identifier for the ICM script which is an attribute that is configured in the IVR profile. The Resource Manager sends the toll-free number attribute in the `Request-URI` that is sent to the CTI Connector in the `tollfreenum` format	String of integers. **Default value:** `0-9`
	`gvp.policy` **Section**	
CTI Allowed	Specifies that this IVR Profile is enabled for CTI functionality through the CTI Connector.	• `True`—Enable • `False`—Disable. **Default value:** `False`

Mapping IVR Profiles to DID Numbers

DID numbers are the DNs that are obtained from Dialed Number Identification Service (DNIS).

The Resource Manager can be configured so that it obtains DNIS information from SIP Server (see `rm.sip-header-for-dnis` on page 91).

- If your GVP configuration includes a mapping of IVR Profiles to DIDs, the Resource Manager will use the DNIS information to determine which IVR Profile to invoke for the session.
- If you do not map IVR Profiles to DIDs, the Resource Manager will use the default IVR Profile that you specify for the tenant.

All options described create the DID mappings as sections in the Tenant's `Annex` tab. Genesys recommends that you use Genesys Administrator's IVR Profile Wizard to map the DIDs.

The following procedure describes how to create mapping rules.

Procedure: Mapping IVR Profiles to DIDs

Purpose: To associate IVR Profiles with DIDs so that the Resource Manager can use DNIS information to invoke the required GVP services.

Prerequisites

- The IVR Profiles have been created if the DID Group is to be mapped to a specific IVR Profile.

  Note: DID Groups can be created without IVR Profiles, and assigned to IVR Profiles at a later time.

  For more information about creating an `IVR Profile`, see the chapter about post-installation activities in the *Genesys Voice Platform 8.1 Deployment Guide.* For more information about configuring the IVR Profile, see "IVR Profile Configuration Options" on page 113.
- You are logged in to Genesys Administrator. To access Genesys Administrator, go to the following URL:

  `http://<Genesys Administrator host>/wcm`

Start of procedure

1. Go to the `Provisioning > Voice Platform > DID Groups`.
2. In the menu bar of the tab, click `New`.

   The `Property` screen displays (see Figure 6).

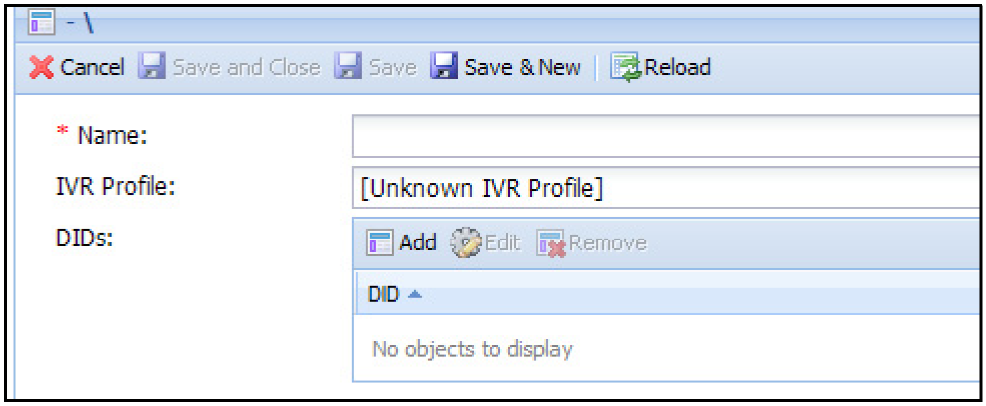


Figure 6: Specifying a DID Mapping

3. In the `Name` text box, enter the name of the DID Group.
4. From the `IVR Profile` drop-down list, select the required IVR Profile.
5. In the Click `DIDs` box, click `Add`.

   The `Add/Edit DID` dialog box displays (see Figure 7).

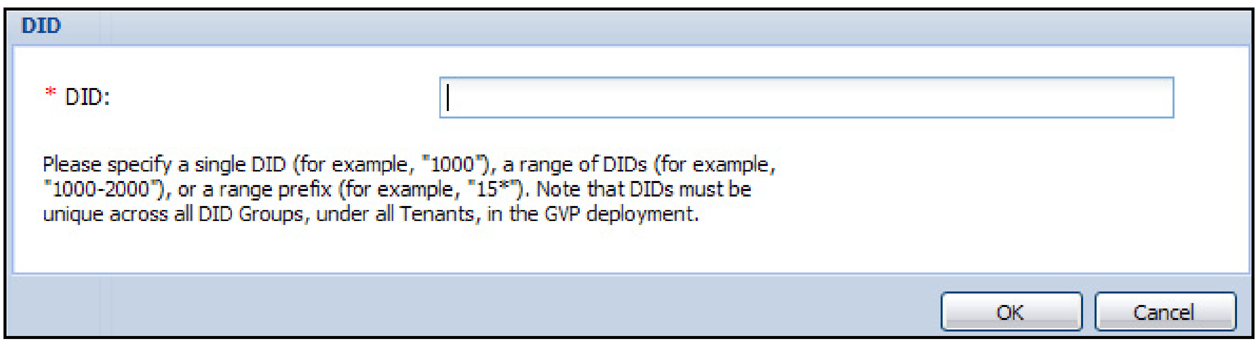


Figure 7: Add/Edit DID

6. Enter the DID.
7. Click `OK`.

   Continue entering the DIDs until finished.

Note: The DIDs can be entered as either a single DID (for example, 100), a range of DIDs (for example, 100-199), or a DID prefix (for example, 100*).

8. To edit or delete existing DIDs from the group, select the DID you want to change, and click either `Edit` or `Remove`.

9. Click Save.

Note: To add, edit or delete multiple DIDs, see “DID Group Bulk Operations Wizard” on page 135.

End of procedure

DID Group Bulk Operations Wizard

The DID Group Bulk Operations Wizard allows you to add, delete, and move multiple DIDs among DID groups. You can also create new DID groups with the wizard. The following procedure describes how to use the DID Group Bulk Operations Wizard.

Procedure:
Using the DID Group Bulk Operations Wizard

Purpose: To add, delete, and move DIDs using the DID Group Bulk Operations Wizard.

Prerequisites

- The IVR Profiles have been created.

  For more information about creating an IVR Profile, see the chapter about post-installation activities in the *Genesys Voice Platform 8.1 Deployment Guide.* For more information about configuring the IVR Profile, see “IVR Profile Configuration Options” on page 113.
- You are logged in to Genesys Administrator. To access Genesys Administrator, go to the following URL:

  `http://<Genesys Administrator host>/wcm`

Start of procedure

1. Go to the Provisioning > Voice Platform > DID Groups.
2. Optionally, select the DID Group you want to change.
3. In the Tasks panel, click Bulk Operations Wizard to invoke the wizard.
4. After reading the introduction, click Next to start the wizard.

5. Select the appropriate operation.
   a. Select `Add` to add multiple DIDs to the selected group, or to create new DID groups, then click `Next`.
      i. Click `Browse` to select the CSV file containing the list of new DIDs

Note: The uploaded file must be a CSV file with the following columns (in the given order):

- DID
- DID Group
- Tenant

A DID is either a single DID (in the form <did>), a range of DIDs (in the form <start>-<end>) or a DID prefix (in the form <prefix>*). Lines of text that don't match these patterns are considered invalid.

A DID Group is the name of either an existing group or a new group that is to be created. This column is optional and defaults to the selected DID Group (if one was selected during the launching of the wizard).

A Tenant is the name of the tenant that owns (or will own) the specified DID Group. This column is optional and defaults to the current tenant.

Because DID Group and Tenant columns are optional, a flat list of DIDs can be uploaded (instead of a CSV) for addition/moving into of DIDs into the selected (in the DID group list) DID group.

ii. Click `Next`.

iii. Review the `Confirmation` screen for a summarization of the operation.

Note: The summary includes the counts for invalid DIDs, valid DIDs, and DIDs that currently belong to other DID Groups.

iv. Click `Finish`.

b. Select `Move` to move the specified DIDs to a selected group, then click `Next`.

   i. Click `Browse` to select the file containing the list of new DIDs.

   Note: The uploaded file must be a CSV file with the following columns (in the given order):

   - DID
   - DID Group
   - Tenant

   A DID is either a single DID (in the form <did>), a range of DIDs (in the form <start>-<end>) or a DID prefix (in the form <prefix>*). Lines of text that don't match these patterns are considered invalid.

   A DID Group is the name of either an existing group or a new group that is to be created. This column is optional and defaults to the selected DID Group (if one was selected during the launching of the wizard).

   A Tenant is the name of the tenant that owns (or will own) the specified DID Group. This column is optional and defaults to the current tenant.

   ii. Click `Next`.

   iii. Review the `Confirmation` screen for a summarization of the operation.

   Note: The summary includes the counts for invalid DIDs, valid DIDs, and DIDs that currently belong to other DID Groups.

   iv. Click `Finish`.

c. Select `Delete` to remove the specified DIDs from the selected group, then click `Next`.

   i. Click `Browse` to select the file containing the list of new DIDs.

   Note: The file must be a text file containing a list of DID numbers. Each line is either a single DID (for example, 100), a range of DIDs (for example, 100-199), or a DID prefix (for example, 100*). Lines of text that do not match these patterns are ignored.

   ii. Click `Next`.

iii. Review the `Confirmation` screen for a summarization of the operation.

Note: The summary includes the counts for invalid DIDs, valid DIDs, and DIDs that currently belong to other DID Groups.

iv. Click `Finish`.

You can download these generated CSV files for diagnostic or auditing purposes.

End of procedure

Data Retention Policy Wizard

The Data Retention Policy Wizard enables data retention policies for data retained by the Reporting Server for Tenants and IVR Profiles easily. The wizard allows the step-by-step configuration for the data retention policies for call detail reporting, operational reporting, VAR, and service quality data. Retention policies can also be configured in the Reporting Server application in Genesys Administrator. For more information, see "Configuring Database Retention Policies" on page 278.

Procedure:
Using the Data Retention Policy Wizard

Purpose: To create data retention policies using the Data Retention Policy Wizard.

Prerequisites

- The IVR Profiles have been created.

  For more information about creating an `IVR Profile`, see the chapter about post-installation activities in the *Genesys Voice Platform 8.1 Deployment Guide*. For more information about configuring the IVR Profile, see "IVR Profile Configuration Options" on page 113.
- You are logged in to Genesys Administrator. To access Genesys Administrator, go to the following URL:

  `http://<Genesys Administrator host>/wcm`

Start of procedure

1. Go to the Provisioning > Voice Platform > IVR Profiles.
2. Under the Tasks panel, select Configure Data Retention Policies to invoke the wizard.

   The Introduction screen appears.
3. Click Next.
4. If you do not want to use the default values, on the CDR Data Retention screen, enter the required duration for Call Log Events, and CDR.

   Note: The value configured for Call Log Events must be less than the value configured for CDR for the Call Log Events value to be valid.

5. Click Next.
6. If you do not want to use the default values, on the OR Data Retention screen, enter the required durations for the following:
   - Five-minute Summaries
   - Thirty-minute Summaries
   - Hourly Summaries
   - Daily Summaries
   - Weekly Summaries
   - Monthly Summaries
7. Click Next.
8. If you do not want to use the default values, on the VAR Data Retention screen, enter durations for the following:
   - Five-minute Summaries
   - Thirty-minute Summaries
   - Hourly Summaries
   - Daily Summaries
   - Weekly Summaries
   - Monthly Summaries
9. Click Next.
10. If you do not want to use the default values, on the SQ Data Retention screen, enter the required durations for the following:
    - Five-minute Summaries
    - Thirty-minute Summaries
    - Hourly Summaries
    - Daily Summaries
    - Weekly Summaries
    - Monthly Summaries
11. Click Next.

12. If you do not want to use the default values, on the `Latency Data Retention` screen, enter durations for the following:
    - `Hourly Summaries`
    - `Daily Summaries`
    - `Weekly Summaries`
    - `Monthly Summaries`
13. Click `Next`, and then `Finish` to save the values.

For more information on the default values, see “Configuring Database Retention Policies” on page 278.

End of procedure

IVR Profile Configuration for Tenants

The `Tenant` configuration options determine how the Resource Manager will use IVR Profiles in the tenant’s environment.

Setting options at the tenant level sets values that are inherited as defaults by the IVR Profiles. You can override these settings for individual IVR Profiles by setting different values for the equivalent options in the IVR Profile.

Configure the tenant parameters in Genesys Administrator on the `Provisioning > Environment > Tenants > <tenant> > Options > Advanced View (Annex)` tab.

All changes to these parameters take effect with the next session that uses the IVR Profile.

gvp.general Section

Table 19 describes the parameters in the `gvp.general` section. These parameters specify general configuration information for the Resource Manager in the tenant’s environment.

Table 19: General Tenant Configuration Options

Parameter Name	Description	Valid Values and Syntax
	`gvp.general` **Section**	
Default Application	(Mandatory) The default IVR Profile for a request to the Resource Manager. The Resource Manager uses the default IVR Profile if the incoming request does not contain information to map the request to an application.	`<IVR Profile>` where `<IVR Profile>` is the name of the IVR Profile that you assigned when you created the `IVR Profile` object. **Default value:** Empty
SIP Session Timer Interval	The timeout value, in seconds, for the SIP session that executes for this IVR Profile. If the Resource Manager receives no SIP messages associated with this call leg within the timeout interval, the Resource Manager considers the call leg to have ended. For the call leg associated with this IVR Profile, the value of this `sip.sessiontimer` parameter overrides session expiry timeouts that are set at the level of the Resource Manager, but may be overridden by the `sip.sessiontimer` setting for the IVR Profile. For more information about how the Resource Manager uses expiry timeouts to manage sessions, see "Resource Manager Session Timers" on page 80.	Any positive integer. **Default value:** Empty

gvp.policy Section

The parameters in this section are identical to the configuration parameters in the `gvp.policy` section of the `IVR Profile` object. These parameters enable you to configure policies for the Resource Manager—for example, to specify which requests the Resource Manager will allow, or to attach certain `Request-URI` parameters to send to the endpoint to enable or disable particular features. You can also specify Reporting Server tenant parameters in the `gvp.policy` section.

For more information about the configuration options in the `gvp.policy` section, see Table 16 on page 113.

gvp.policy.sqa Section

The parameters in this section are determine the Reporting Server service quality behavior for the tenant. Table 20 describes the import options in the `gvp.policy.sqa` section.

Table 20: Service Quality Tenant Configuration Options

Parameter Name	Description	Valid Values and Syntax
	`gvp.policy.sqa` **Section**	
Threshold for Service Quality Notification	If the percentage of successful calls for an IVR Profile falls below this threshold during a service quality period, a notification is generated. If set to `-1`, the Reporting Server default value is used.	An integer in the range of -1–100. **Default value:** `-1`

For more information on how to change the default `sqa.error.notification.threshold` value for the Tenant in Genesys Administrator, see the *Framework 8.1 Genesys Administrator Help*.

gvp.service-parameters Section

The parameters in this section are identical to the configuration parameters in the `gvp.service-parameters` section of the `IVR Profile` object. The Resource Manager uses these values to add, modify, or delete `Request-URI` parameters in the SIP requests that it forwards.

For more information about the configuration options in the `gvp.service-parameters` section, see Table 16 on page 113.

gvp.policy.<child-tenant> Section

The parameters in this section are those policies that Resource Manager must enforce for a child tenant on behalf or a parent tenant. For more information about the configuration options in the `gvp.policy parameters` section, see Table 16 on page 113.

gvp.dn-groups Section

Table 21 describes the parameters in the `gvp.dn-groups` section. These parameters specify the mapping for DID groups in an Hierarchical Multi-Tenancy (HMT) environment. The Resource Manager obtains DID group information from this section.

Table 21: DID Groups Tenant Configuration Options

Parameter Name	Description	Valid Values and Syntax
`gvp.dn-groups` **Section**		
DN Group Name	The name of each parameter represents the name of the DID Group. The value contains the list of DIDs. The value is a comma-separated string in which each value represents one of the following: • An individual DID • A consecutive block of DIDs • A prefix DID with a * suffix	• An individual DID, for example, 1000 • A consecutive block of DIDs, for example, 1000-1999 • A prefix DIO with * suffix, for example, 1800555* • Any number of the above combinations separated by commas for a DID Group.

gvp.dn-group-assignments Section

Table 22 describes the parameters in the `gvp.dn-group-assignments` section. These parameters specify the IVR Profile mapping for DID Group assignments in an HMT environment.

Table 22: DID Group Assignments Tenant Configuration Options

Parameter Name	Description	Valid Values and Syntax
`gvp.dn-group-assignments` **Section**		
DN Group Name	The DBID of the IVR profile to which the DID group is mapped. The value must be a positive integer. Any tenant in the hierarchy can define DID Groups. Each group can contain: • An individual DN. • A consecutive block of DNs. • A prefix DN string (for example, 1234*). Corresponding to these DID Groups, each tenant can contain DID group assignments to specify the target IVR Profile for each group. Only those IVR Profiles that are at the current tenant level can be assigned.	A positive integer.

Chapter

7 Configuring the Media Control Platform

The Genesys Voice Platform (GVP) Media Control Platform (MCP) component provides media-centric services. This chapter provides information about the configuration of the Media Control Platform and, if required, the provisioning of the resources for Automatic Speech Recognition (ASR) and Text-to-Speech (TTS). This chapter contains the following sections:

Task Summary: Configuring the Media Control Platform

Task Summary: Configuring the Media Control Platform summarizes the configuration steps and options to implement Media Control Platform functionality in your GVP deployment.

Task Summary: Configuring the Media Control Platform

Objective	Related Procedures and Actions
Integrate the Media Control Platform with the Resource Manager.	1. Point the Media Control Platform to the Resource Manager as the SIP Proxy server, and define the properties for SIP communications. Key configuration options are: • `sip.transport.x` • `sip.routeset` or `sip.securerouteset` 2. To secure SIP communications, ensure that you specify a transport for the Transport Layer Security (TLS) protocol and a secure routeset for outbound calls. For additional, relevant configuration options, see “Configuring SIP Communication and Routing” on page 42. 3. If you if intend to use the Call Recording Solution through third-party recording servers, configure the following option so that it points to the Resource Manager’s IP address and port: • `vrmrecorder.sip.routeset`

(Optional) Secure the media channel between the Media Control Platform and the remote endpoint.	1. Enable Secure Real-time Transport Protocol (SRTP) by specifying the required mode (`accept-only`, `offer`, or `offer_strict`) in the `mpc.srtp.mode` parameter. By default, SRTP is not enabled.
	2. If necessary, modify the default values for the encryption and authentication algorithms (the cryptographic suites) and session parameters that the Media Control Platform will advertise in the SDP `crypto` attribute:
	`mpc.srtp.cryptomethods`
	`mpc.sessionparams`
	`mpc.sessionparamsoffer`

If required for your deployment, provision the third-party Media Resource Control Protocol (MRCP) servers for ASR and TTS.	See “Enabling ASR and TTS” on page 150.

Task Summary: Configuring the Media Control Platform (Continued)

Objective	Related Procedures and Actions

Configure the IP DiffServ (ToS) and RTP/RTCP.	Set the RTP/RTCP packet and the SIP packets ToS using:
	`[mpc] rtp.tos`
	`[mpc] rtcp.tos`
	`[sip] transport.[n].tos`
	See "Configuring SIP Communication and Routing" on page 42.

Configure conferencing.	See "Enabling Conference Services" on page 62.
Configure reporting.	See "Configuring Reporting" on page 63.
Configure logging.	See "Configuring Logging" on page 66.
Tune Media Control Platform performance.	• Configure appropriate maximums and timeouts for your deployment. Consider the following options, in particular: `vxmli.cache.document.max_count` (default is `50`) `vxmli.cache.document.max_size` (default is `1000000 bytes`) `vxmli.max_num_documents` (default is `2000`) `vxmli.initial_request_fetchtimeout` (default is `30000 ms`) `vxmli.max_num_sessions` (default is `10000`) • If your deployment includes ASR and TTS, consider the following options, which affect the MRCP Client behavior: `vrm.client.timeout` (default is `10000 ms`) `stack.connection.timeout` (default is `10000 ms`) • See also "Configuring Session Timers and Timeouts" on page 80. • It is usually not necessary to modify the default settings for the media processing behavior of the Media Server (`mpc` and `mtinternal` configuration sections). However, review buffer and packet size related `mediamgr.*` and `rtp.*` options in the `mpc` configuration section, to verify that they are optimal for your deployment (especially if using video).
Customize Media Control Platform behavior in relation to VoiceXML applications.	• Review and, if necessary, modify the configuration options in the `vxmli` configuration section (see the *Genesys Voice Platform 8.1 Configuration Options Reference*). Some of the important NGI `vxmli` options are described in Table 23 on page 160. Some of the important GVPi options are described in Table 23. • For Next Generation Interpreter (NGI), consider also the parameters in the `sip` configuration section that specify what parts of SIP messages are exposed to the VoiceXML application (for example, `in.invite.headers` and `in.invite.parameters`). For the list of SIP headers that are known to GVP, see Table 96 on page 449.

Task Summary: Configuring the Media Control Platform (Continued)

Objective	Related Procedures and Actions

Customize Media Control Platform for Inband DTMF detection.	• Verify that settings for the following configuration options, which are required for DTMF detection, are suitable for your deployment. Consider the following parameters:
	`mpc.rtp.dtmf.receive`
	`mpc.rpt.dtmf.send`
	`mpc.dtmf.detectedge`
	`mpc.dtmf.maxsilence`
	`mpc.dtmf.minduration`
	`mpc.fcr.defaultdtmfhandling`
	`mpc.record.defaultdtmfhandling`

Customize Media Control Platform for Call Progress Analysis	• Verify that the general CPA settings are suitable for your deployment. Consider the following parameters:
	`cpa.maxpreconntime`
	`cpa.maxpostconntime`
	`cpa.maxbeepdettime`
	`cpa.keptdur_before_statechange`
	`cpa.priority_normal_machinegreetingdur`
	`cpa.priority_normal_voicepausedur`
	`cpa.priority_normal_maxvoicesigdur`
	`cpa.priority_voice_machinegreetingdur`
	`cpa.priority_voice_voicepausedur`
	`cpa.priority_voice_maxvoicesigdur`
	`cpa.priority_machine_machinegreetingdur`
	`cpa.priority_machine_voicepausedur`
	`cpa.priority_machine_maxvoicesigdur`
	`cpa.faxdur`
	`cpa.voice_range_db`
	`cpa.maxrings`
	`cpa.voice_level_db`
	`cpa.preconn_tones_det_mode`

Task Summary: Configuring the Media Control Platform (Continued)

Objective	Related Procedures and Actions

Customize Media Control Platform for Call Progress Analysis
(Continued)
	• Re-define standard tones or add new custom tones if necessary for your deployment:
	`cpa.fax`
	`cpa.ringback`
	`cpa.busy`
	`cpa.fastbusy`
	`cpa.sit_nocircuit`
	`cpa.sit_vacantcircuit`
	`cpa.sit_operatorintercept`
	`cpa.sit_reorder`
	`cpa.custom1`
	`cpa.custom2`
	`cpa.custom3`
	`cpa.custom4`
	`cpa.tone[1-10].segment[1-3].f[1-2]min`
	`cpa.tone[1-10].segment[1-3].f[1-2]max`
	`cpa.tone[1-10].segment[1-3].ontimemin`
	`cpa.tone[1-10].segment[1-3].ontimemax`
	`cpa.tone[1-10].segment[1-3].offtimemin`
	`cpa.tone[1-10].segment[1-3].offtimemax`

Customize Media Control Platform for video recording.	• For NGI, if required, set the following configuration option to enable I-frame request during video recording: `mpc.rtp.request_iframe`

Customize Media Control Platform for PSTN Connector prompt play support.	• For PSTN Connector, if installed, the following configuration options can be adjusted for your requirements:
	`mpc.playremoteflushtimeout`
	`mpc.playremoteeodtimeout`
	`mpc.rtp.prefilltime`

Customize client-side communication ports.	See “Configuring Client-Side Connections” on page 72
Customize Media Control Platform for Outbound dialing.	See “Enabling Outbound Dialing” on page 153.
Customize session management behavior and performance.	See “Configuring Session Timers and Timeouts” on page 80.
Customize Media Control Platform messaging.	See “Configuring SNMP” on page 72 and Table 100 on page 470.

Task Summary: Configuring the Media Control Platform (Continued)

Objective	Related Procedures and Actions
Customize Media Control Platform for SIP Server and MSML.	Create a VoIP Service DN and set the contact option to the Resource Manager IP address in the T-Server section. For more information, see the *Framework 8.1 SIP Server Deployment Guide*.

Customize Media Control Platform for DTMF transmission method.	Consider the following configuration parameters:
	`mcp.rtp.dtmf.send`
	`mcp.sdp.map.orign.[n].dtmftype`

Customize Media Control Platform jitter buffer.	Consider the following configuration parameters:
	`mpc.rtp.recvaudiobuffersize`
	`mpc.rtp.recvvideobuffersiz,`
	`mpc.rtp.dejitter.delay`
	`mpc.rtp.dejitter.timeout`

Customize Media Control Platform for VoIP metrics reporting.	Consider the following configuration parameters:
	`mpc.voipmetrics.enable`
	`sip.voipmetrics.localhost`
	`sip.voipmetrics.registration`
	`sip.voipmetrics.remoteserver`
	`sip.voipmetrics.routeset`

Enabling ASR and TTS

The following procedure describes how to create and configure MRCP server `Applications`, to provision ASR and TTS speech resources for the GVP deployment.

Procedure:
Provisioning ASR and TTS resources

Purpose: To provide an overview of the steps to configure logical GVP MRCPv1 or MRCPv2 speech server `Applications`, to provide a presence for third-party speech engines in the Genesys Configuration Layer.

Repeat this procedure as required to create the necessary `Application` objects. You must create a separate `Application` for each third-party speech server in your deployment. The `Application` type is `Resource Access Point`.

Prerequisites

- You are logged in to Genesys Administrator. To access Genesys Administrator, go to the following URL:
  `http://<Genesys Administrator host>/wcm`
- The Media Control Platform Installation Package (IP) is available.

Start of procedure

1. Create the MRCPv1 or MRCPv2 `Application` object.
   a. Import the required `Application Template` from the Media Control Platform Installation Package (IP).
      The following `Application Templates` are available:.

• `MRCPv1_ASR_IBM`	• `MRCPv2_ASR_NUANCE`
• `MRCPv1_ASR_NUANCE`	• `MRCPv2_TTS_NUANCE`
• `MRCPv1_ASR_TELISMA`	• `MRCPv2_ASR`
• `MRCPv1_TTS_IBM`	• `MRCPv2_TTS`
• `MRCPv1_TTS_NUANCE`	•
• `MRCPv1_ASR`	•
• `MRCPv1_TTS`	•

Note: The generic templates can be used for those vendors that are not listed above.

   b. Import metadata into the `Application Template.`
   c. On the `Provisioning > Environment > Applications` tab, create and name the new `Resource Access Point,` *based on the required* `Application Template.`

   For detailed information about importing `Application Templates` and metadata, and creating `Applications` from the templates, see Appendix A in the *Genesys Voice Platform 8.1 Deployment Guide,* which describes the pre-installation activities.

Configure Resource Access Points:

2. If you are configuring the resource access points for the MRCP Server, omit Step b. If you are configuring them for the MRCP Proxy, omit Step a:

MRCP Server

   a. On the `Provisioning > Environment > Applications > <MRCP Server> > Options` tab, configure the options in the `provision` section.
      - `vrm.client.resource.type`—Enter the resource type (ASR or TTS).

- `vrm.client.resource.name`—Enter a name for the speech resource. (See template names in Step 1).

> **Note:** MRCP resources of the same type (ASR or TTS) can be assigned the same resource name. However, a resource name that is used by a ASR resource cannot be used by a TTS resource.

- `vrm.client.resource.uri`—Enter the MRCP Server's RTSP URI. For example, `rtsp://<MRCPServer Host IP>:<MRCPServer Port>/<suffix>`
- `vrmproxy.ping_interval`—Enter a value for the interval between pings (to the MRCP Proxy). (Not required if you have not deployed the MRCP Proxy.)

MRCP Proxy

b. On the `Provisioning > Environment > Applications > <MRCP Proxy> > Options` tab, configure the options in the `provision` section.
   - `vrm.client.resource.type`—Enter the resource type (ASR or TTS).
   - `vrm.client.resource.name`—Enter a name for the speech resource.
   - `vrm.client.resource.uri`—Enter the MRCP Proxy's RTSP URI. For example, `rtsp://<MRCPP Host IP>:<MRCPP Port>/<suffix>`

For other important configuration options that you may need to be modify, see "Important MRCP Server Configuration Options" on page 194.

Configure TTS Vendor-Specific Parameters:

3. If required, configure the TTS vendor-specific parameters that will be sent in `SET-PARAM` requests:
   a. In the `provision` section on the `Provisioning > Environment > Applications > <MRCP Server> > Options` tab, add a new parameter, `vrm.client.TTSVendorSpecific.xxxxxx`.

      This defines one arbitrary TTS vendor-specific parameter to be sent to the MRCP server.
   b. Define as many vendor-specific keys as you require for the desired vendor-specific key-value pairs, using the following format:
      `vrm.client.TTSVendorSpecific.param<n>=value<n>`
   c. Click `Save` or `Apply` to save the MRCP server configuration.

Creating the Connections:

MCP/ MRCP Server

4. If you are deploying the Media Control Platform without the MRCP Proxy:
   - On the `Provisioning > Environment > Applications > <Media Control Platform> > Configuration` tab, create the connection between the Media Control Platform and the MRCP Server.

MCP/ MRCPP/ MRCP Server

5. If you are deploying the Media Control Platform with the MRCP Proxy:
   - On the `Provisioning > Environment > Applications > <Media Control Platform> > Configuration` tab, create the connection between the Media Control Platform and the MRCP Proxy.
   - On the `Provisioning > Environment > Applications > <MRCP Proxy> > Configuration` tab, create the connection between the MRCP Proxy and the MRCP Server.

For more details, see the procedure to assign the MRCP Proxy to the Media Control Platform and the MRCP Server to the MRCP Proxy in the chapter about post-installation activities in the *Genesys Voice Platform 8.1 Deployment Guide.*

End of procedure

Enabling Outbound Dialing

One of the most useful features in GVP 8.1 is the ability to initiate outbound calls in an asynchronous manner through remote dialing and either configured bi-directional or outbound channels.

Making a Call

You can use the telnet interface to connect to the preconfigured remote dialing port (default 6999) to place outbound calls. The following example shows the outbound request with a VoiceXML page:

```
pw@galahad 379>
pw@galahad 379> telnet localhost 6999
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
PW RemoteDial>
call 4167360905 4167362012
http://www.genesyslab.com/helloworld.vxml 0001 Test
!CALL_SENT 1: telno:4167360905 dnis:4167362012
url:http://www.genesyslab.com/helloworld.vxml
uuidata:Test
PW RemoteDial>
!CALL_STATUS 1: CONNECTED: Line is connected.
PW RemoteDial>
!CALL_DROP 1 41: USER_END: User hung up call. (time spent
was 41 secs) (protocol reason: [DlgcChannel] User
hangup)
PW RemoteDial>
```

You can also use the command-line interface to make an outbound call. This interface provides a number of useful commands. These include:

```
call <telno> <ani> <url> <refno> [uuidata] [defaults]
```

`[parameter_list]`

The call command initiates an outbound call to the specified telephone number (`<telno>`). The `<telno>` parameter can accept up to 1023 characters. This can either be a `sip: uri`, or a `tel: uri`. If not specified, a `tel: uri` is assumed. If a `tel:uri` is used, the `defaultgw` configuration option in the `sip` section must be configured to point to a device that can handle the SIP call (for example, a media gateway), so that the call can be forwarded. When connected, the VoiceXML page referred to by the specified URL (`<url>`) is attached to the call.

The value of the `<ani>` parameter is displayed in the CDR as the `local.uri`. If the value of `<ani>` is a SIP URI, it is used in the `From` header of the SIP INVITE request. If the value of `<ani>` is not a SIP URI, it is converted into a SIP URI, and used in the `From` header of the SIP INVITE request.

The `<platform ANI>` parameter can accept up to 32 characters. The actual number of ANI digits that can be delivered on PSTN depends on the network—for example, the maximum number on ISDN T1 is 15.

The reference number (`<refno>`) parameter is a user-supplied identifier that is used to associate status replies with the call initiation, and is unique for each active call. This reference number must be an integer between `0` and `2147483647`.

There are three other optional parameters that can be specified:

- `[uuidata]`—The user-to-user information element.
- `[defaults]`—The default VoiceXML page.
- `[parameter_list]`—The name value pair separated by the pipe (`|`) character that is passed from the interface to the call manager.

The `gvp.appmodule` in the parameter list is used to specify the Next Generation Interpreter used to execute the vxml page.

Note: GVPi does not support remote dialing.

You can also specify all the parameters before the `gvp.appumodule` parameter, or specify the dash (-) character for the default value. For example, if

you wish to specify the parameter list, but not the uuidata and the defaults file, use the following command:

```
PW RemoteDial> call 4167366493 2323
http://205.150.90.12/developer/main/cgi-bin/index.cgi
1223 - - NWNAME=dtiB1T21|NUMBERINGPLAN=0
```

Note: The size of `<uuidata>` cannot exceed 254 characters.

Other valid commands within the interface are:

- `cc`—Clears all message counters.

- dump—Toggles debug information.
- e, q, or x —Exit the command-line interface.
- setto <unit_type> <num_units>—Sets the timeout for the call setup.
- timelimit <seconds>—Sets the maximum number of seconds for the call. The default is 604800 seconds.
- end <refno>—Ends the call at the Remote Dial interface.
- analysis y, n, or a—Enables dialogic call analysis.
- scall—Displays the calls for this session.
- scount—Displays the counters for this session.
- show—Displays the current settings. For example:

  ```
  !SETTINGS: max_calls:500 timeout_length:120
  timeout_unit:s call_analysis:enabled time_limit:604800
  ```
- ? or h—Displays the help information the interface.

Status Messages

Once the call has been placed with the CALL_SENT notification, there are two possible status messages returned.

Note: The <refno> returned in the status message must match the one the following:

- !CALL_SENT <refno>—telno:<telno> dnis:<dnis> url:<url> uuidata:<uuid> defaults_file:<defaults> parameter_list:<parameter_list>
- !SOCKET_ERROR <refno>— Socket not found
- !NO_REFNO—No reference number
- !INVALID_REFNO <refno>—Invalid reference number
- !TOO_MANY_CALLS <refno>—Too many calls in progress
- !INVALID_TELNO <refno>—Incorrect telephone number
- !INVALID_URL <refno>—Incorrect URL format
- !INVALID_UUIDATA <refno>—Incorrect UUIDATA format
- !INVALID_DEFAULTFILE—Incorrect DEFAULTFILE format
- !INVALID_PAIRLIST—Incorrect PAIRLIST format.
- !CALL_FAILED <refno>—telno:<telno> dnis:<dnis> url:<url> uuidata:<uuid> defaults_file:<defaults> parameter_list:<parameter_list>

In all these cases, except for CALL_SENT, there will be no further status returned for this call attempt.

1. The call connects successfully:

   !CALL_STATUS <refno> followed by one of the following:

- CONNECTED—Connected successfully.
- MACHINE—Answering machine detected.
- UNKNOWN_STATUS <status number>—Status is unknown.

The call is then dropped with the following message:

```
!CALL_DROP <refno> <timespent> <network disconnect reason>: <one of
the disconnect reason listed below:> <protocol reason: protocol
disconnect string>
```

- USER_END—The caller disconnected the call.
- APPL_END—The VoiceXML application disconnected the call.
- TIMELIMIT_END—The timelimit of call is reached.
- UNKNOWN_REASON <internal disconnect reason number>—The reason for the disconnect is unknown.

2. The call does not connect and is dropped immediately with a !CALL_DROP message instead of a !CALL_STATUS message.

```
!CALL_DROP <refno> <timespent> <network disconnect reason>:
<drop_status>. < protocol reason: protocol disconnect string>
```

where <drop_status> is one of the following:

- MACHINE—Answering machine detected.
- VXML_DECLINE—The VoiceXML Interpreter declined the call.
- BUSY—The connection is busy.
- NO_ANSWER—There is no answer in <num_units> <unit_type>.
- NO_RESOURCES—There are no free channels or media resources.
- CALL_FAILED—The call failed.
- URLTIMEOUT—The fetch URL timed out.
- BADURI—The URI type is invalid.
- NOAUTH—The network denied the call.
- SHUTTINGDOWN—The Interpreter is shutting down.
- NETWORKTIMEOUT—The network timed out.
- BADDEST—The destination number is invalid.
- UNSUPPORTED_URL—The URL is not supported.
- INVALID_TELNO—The telephone number is invalid.
- USER_END—The caller disconnected the call.
- UNKNOWN_REASON <internal disconnect reason number>—The reason for the disconnect is unknown.

The <network disconnect reason>, and the <protocol disconnect string> are returned by the callmanager to give more information about the reason the call was dropped.

Media Server Markup Language

GVP uses the Media Server Markup Language (MSML) application module to control many different types of services for the Media Control Platform.

In the current release of SIP Server (up to version 7.6), media interactions can be initiated with different components. For IVR interactions, SIP Server integrates with GVP using VoiceXML applications, and for simple media operations and conferencing, SIP Server integrates with the Stream Manager. Simple media operations (conferencing, announcements, and simple dialog management) are controlled using SIP, NETANN (RFC 4240), and extensions to the NETANN protocol.

With more complex media requirements, such as media switching, a full featured media server language is required to provide control over the media channel. MSML support has been introduced to allow for such complex media interactions. The use of MSML allows the Media Control Platform to provide advanced media services to other Genesys components. For example, Call Progress Detection services can be provided to Genesys offerings such as the Outbound Contact Solution, and for GVP itself.

The MSML functionality on the MCP is designed to operate with the Supplementary Services Gateway, and the Outbound Contact Server through the SIP Server.

The MSML interface offers the following functionality:

- Performing Call Progress Detection (CPD) on calls.
- Launching VoiceXML applications with the NGI after performing CPD operations.
- Launching VoiceXML applications with the NGI without performing CPD operations.
- Playing audio and video prompts.
- Recording, including dual-channel (audio) call recording.
- DTMF collection.
- Conferencing.
- Bridging calls without using a VoiceXML application.

For the complete Genesys Media Server 8.1 supported MSML specification, see the *Genesys Voice Platform 8.1 Deployment Guide,* Appendix B.

Important Media Control Platform Configuration Options

This section describes the key configuration options that you either must or might want to customize.

Configure the options on Genesys Administrator on the `Provisioning > Environment > Applications > <Media Control Platform> > Options` tab. For the detailed steps to configure option settings, see Procedure: Viewing or modifying GVP configuration parameters, on page 34.

The configurable Media Control Platform parameters are in the following configuration sections:

- `asr`—Session Manager parameters determine specific configuration for ASR behavior.
- `calllog`—Parameters determine call recording file management.
- `conference`—Parameters determine the default behavior of the Conference application module, for NETANN-initiated conference calls.
- `cpa`—Parameters determine call progress analysis (cpa) type detection.
- `email`—Parameters enable you to configure e-mail address information for maintained e-mail messages.
- `ems` (see Table 6 on page 64)—Parameters determine Reporting behavior for call detail records (CDRs) and metrics.
- `fm`—Parameters determine file fetching and caching behavior for NGI.
- `log` (see “Service Quality Analysis (SQA)” on page 65)—Parameters determine behavior for Management Framework logging.
- `mpc` and `mtmpc`—Parameters determine the default media processing and transport behavior of the Media Processing Component (MPC), or Media Server.
- `mtinternal`—Parameters determine the behavior of the Internal Media Transport application module, which is responsible for managing internal media transmission between the Media Server and the ASR and TTS speech engines. This internal data transmission uses RTP.
- `msml`—Parameters determine Media Server Markup Language (MSML) functionality.
- `Netann`—Parameters determine default behavior for the NETANN Prompt Announcement application module.
- `remdial`—Parameters determine remote dialer behavior.
- `sessmgr`—Parameters determine call control and platform-level behavior of the Call Manager API (CMAPI) application modules that are loaded at startup.

Note: Genesys recommends that you do not modify the default values, unless you are an advanced user who needs to use special CMAPI applications for your deployment.

- `sip`—Parameters integrate the Media Control Platform with the SIP Proxy (the Resource Manager). These parameters determine the behavior of the SIP Line Manager application module, and configure the supported transport interfaces.
- `snmp` (see “Configuring SNMP” on page 72)—Parameters that determine SNMP behavior.
- `stack`—Parameters relate to the MRCP stack and determine the way the Media Control Platform manages connections to the external MRCP server.
- `tts`—Parameters determine specific configuration for TTS behavior.
- `vrm`—Parameters determine the behavior of the MRCP Client. These parameters relate to the Voice Resource Management (VRM), or Speech Resource Management (SRM), module.
- `vxmli`—Parameters determine the behavior of the Next Generation Interpreter (NGI).

Table 23 provides information about important Media Control Platform parameters that are not described in Chapter 3 on page 41. Table 23 provides parameter descriptions as well as the default parameter values that are preconfigured in the Media Control Platform `Application` object.

Unless indicated otherwise, all changes take effect on restart.

For information about all the available configuration options for the Media Control Platform, see the *Genesys Voice Platform 8.1 Configuration Options Reference.*

For information about configuring multiple Media Control Platforms, see “Deploying Multiple Media Control Platforms” in the *Genesys Voice Platform 8.1 Deployment Guide*.

Table 23: Selected Media Control Platform Configuration Options

<table>
<tr><th>Option Name</th><th>Description</th><th>Valid Values and Syntax</th></tr>
<tr><td colspan="3">asr Section</td></tr>
<tr><td>ASR Load once per call</td><td>Specifies whether there will be one VRM session for the entire call, or whether a separate VRM session will be opened for each recognition request.
A single session for the entire call (load_once_per_call = 1) means that each call may have multiple recognition sessions.
If this parameter is set not to enable a single session for the entire call (load_once_per_call = 0), each VRM session is closed when the recognition request completes, either successfully or unsuccessfully (such as no match). Therefore, each call may have multiple VRM sessions.
Having multiple VRM sessions in a call may improve the efficiency of ASR server license usage. However, be aware of the following possible consequences:
• There will be longer delays on speech barge-in.
• Some recognizer servers delete saved utterance data after each VRM session. In these cases, the VoiceXML application cannot refer to the saved utterance file after the recognition session.
Changes take effect: Immediately.</td><td>• True—Single session not enabled.
• False—Single session enabled.
Default value: False (only one VRM session for the entire call)</td></tr>
<tr><td>ASR Engine Default</td><td>Specifies the default ASR Engine resource when using Request URI.
Changes take effect: Immediately</td><td><resourcename>
[?<protocol>]
For example, SPEECHWORKS?MRCPv1</td></tr>
<tr><td colspan="3">**callmgr Section**</td></tr>
<tr><td>FIPS Mode Enabled</td><td>Enables FIPS mode in MCP.</td><td>True
False
Default value: False
Changes take effect: start/restart</td></tr>
</table>

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
conference Section		
Conference Participant Limit	Specifies the maximum number of participants allowed for a conference that is initiated by a conferencing application. If this option is set to 0, the number of participants allowed is unlimited and depends on the machine resource limits.	Any integer greater than or equal to 0. **Default value:** 0
Conference Highest Input	Specifies the number of highest inputs that will be used for mixing output. If this option is set to 0, all inputs are used.	A range of integers. **Default value:** 3
Conference Video Output Type	Specifies the type of video output for conferences. • If set to single, a single stream output is enabled, where the video stream from one conference participant is sent to each conference participant. • If set to mixed, a video mixed output is enabled, where the video streams from multiple conference participants are combined into one frame and sent to each participant.	• single • mixed **Default value:** single
cpa Section		
The CPA Method Used for Outbound Calls	Specifies the supported Outbound CPA method. **Changes take effect:** Immediately.	• NONE—Disable CPA for outbound calls. • AUDIOCODES—CPA using AudioCodes gateway. • PSTNC—CPA using PSTN Connector. • NATIVE—CPA using Native CPA. **Default value:** NONE

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Supported Gateway CPA Events	A space-separated list of supported Gateway CPA events. If `NONE` is selected, CPA events will not be sent to the application. **Changes take effect:** Immediately.	• `AMD` —Answering Machine Detection • `CPT` —Call Progress Tones Detection • `FAX` —Fax Machine Detection • `PVD`—Positive Voice Detection • `PTT` —Push to Talk Events **Default value:** `AMD CPT FAX PVD`
Outbound Calls with Native CPA - Initial State	Specifies the initial CPA state when using Native CPA. **Changes take effect:** Immediately.	• `preconnect` —Detection starts as soon as the call is initiated. • `postconnect` —Detection starts when the call is connected. **Default value:** `preconnect`
Outbound Call with Native CPA - Ignore Call Connect Events	Specifies whether the CPA algorithm ignores the call connect event. **Note:** This parameter is valid only if Outbound Calls with Native CPA - Initial State is set to `preconnect`. **Changes take effect:** Immediately.	• `True` —Ignore call connect event. • `False` —Use the call connect event. **Default value:** `False`
`fm` Section		
HTTP Proxy	Specifies the HTTP proxy to use for HTTP requests.	`<host:port>` **Default value:** `localhost:3128`
HTTPS Proxy	Specifies the HTTPS proxy to use for HTTPS requests.	`<host:port>` **Default value:** Empty

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Outgoing Interface	Specifies the network interface IP address that is used for outgoing HTTP requests. If this configuration option has an empty value, the Media Control Platform automatically selects the network interface it will use. If the Squid HTTP proxy is used, it must be configured to accept HTTP requests from the interface that is specified. Otherwise, by default, it accepts HTTP requests from the local host only.	Any string of characters. **Default value:** Empty
No Cache URL Substring	Specifies that documents fetched from a URL containing one of the substrings in this list should not be cached. Any substring listed in this comma delimited list, *will not* be cached.	Any comma delimited list of characters. **Default value:** `cgi-bin,jsp,?`
Maxage for Local File	Specifies, in seconds, how long to cache local file for. If set to `0`, local files will not be cached.	Any integer. **Default value:** `60`
Maximum Cache Size	The total maximum size, in bytes, of all cached files.	Any integer. **Default value:** `50,000,000`
Maximum Cache Entry Size	Specifies the maximum size, in bytes, of each cache entry.	Any integer. **Default value:** `500,000`
Maximum Cache Entry Count	Specifies the maximum number of entries that can be stored in cache.	Any integer. **Default value:** `1000`
Maximum Redirections	Specifies the maximum number of times to follow the `Location` header in the HTTP response. If set to `0`, HTTP redirection is disabled.	Any string of characters. **Default value:** `5`
Enable 100-Continue header	Specifies whether to enable the Expect: 100-continue header in HTTP 1.1 requests.	• `0`—Disable • 1—Enable **Default value:** `0`
SSL Certificate	Specifies the certificate file name.	Any string of characters. **Default value:** Empty

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
SSL Certificate Type	Specifies the format of the certificate file name.	• `PEM`—Privacy Enhanced Mail • `DER`—Distinguished Encoding Rules **Default value:** `PEM`
SSL Key	Specifies the private key file name.	Any string of characters. **Default value:** Empty
SSL Key Type	Specifies the format of the key file name.	• `PEM`—Privacy Enhanced Mail • `DER`—Distinguished Encoding Rules **Default value:** `PEM`
SSL Key Password	Specifies the password required in order to use the SSL Key.	Any string of characters. **Default value:** Empty
SSL Version	Specifies the Secure Socket Layer version to use.	• `0`—Automatically detect version • `1`—Force TLSv1 • `2`—Force TLSv2 • `3`—Force TLSv3 **Default value:** `0`
Verify Peer Certificate	Specifies whether to verify the peer's certificate. **Note:** SSL CA Info or SSL CA Path must also be set in order for this parameter to take affect.	• `0`—Do not verify • `1`—Verify **Default value:** `0`
SSL CA Info	Specifies the file name to use for verifying peer certificate.	Any string of characters. **Default value:** Empty
SSL CA Path	Specifies the path to the directory holding the peer certificates. **Note:** This directory must be created using the `openssl c_rehash` utility.	Any string of characters. **Default value:** Empty
SSL Random File Seed	Specifies the random initial value used to generate the first number of the SSL key.	Any string of characters. **Default value:** Empty

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
SSL Verify Host	Specifies how the common name from the peer certificate is to be verified during the SSL handshake.	• 0—Do not verify • 1—Check existence only • 2—Make sure that it matches provided host name **Default value:** 0
SSL Cipher List	Specifies the list of ciphers to use for the SSL Connection.	Any string of characters. **Default value:** Empty
	mpc Section	
Append Rejected Codecs	Specifies whether GVP will advertise all supported codecs when it generates a Session Description Protocol (SDP) answer or SDP offer. Even if codecs are rejected or not presented in the caller's SDP message, the platform will still support receiving these codecs. The platform will not send for the SDPs unless a payload is presented by the caller. **Changes take effect:** Immediately.	• 0—GVP will not advertise all supported codecs. • 1—GVP will advertise all supported codecs. **Default value:** 0
Codecs	A space-separated list of the codecs that correspond to the platform capabilities advertised with SDP. The list controls which codecs the Media Control Platform offers to the remote party, for media sent from the remote party to GVP. **Changes take effect:** Immediately.	• amr • amr-wb • g722 • g726 • g729 • gsm • h263 • h263-1998 • h264 • pcmu • pcma • telephone-event • tfci **Default value:** pcmu pcma g726 gsm h263 h263-1998 h264 telephone-event

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Codec Preference	Specifies whether remote or local preferences will be used to interpret the list of accepted codecs. • Local preferences means that the effective accept list is the locally configured accept list, filtered to include only those capabilities also offered by the remote entity. • Remote preferences means that the effective accept list is the list of formats offered by the remote entity, filtered to include only those entries also on the locally configured list. **Changes take effect:** Immediately.	• `l`—Local preferences will be used. • `r`—Remote preferences will be used. **Default value:** `r`
<codec> maxptime	If the MCP is offering the SDP, or answering the SDP where the offer does not have the maxptime, the maxptime attribute will be set according to this configuration. If this configuration does not exist, or is disabled, the maxptime attribute will not be sent unless the SDP offer had the maxptime attribute. In the case where other codecs in the SDP also specify maxptime, the configuration of the codec listed before this codec will take precedence.	• `0` • `10` • `20` • `30` • `40` • `60` • `80` • `100` **Note:** 0 = Disabled
<codec > ptime	Specifies the duration, in milliseconds, and arrival interval of one RTP packet. For example, if the AMR codec is sent at 20 ms ptime, then one AMR RTP packet contains 20 ms duration of audio data. Also, this 20 ms ptime packet must be sent every 20 ms in order to supply audio data continuously. If the remote SDP does not specify the `ptime` attribute, this option is used as the transmission rate of this codec.	• `0` • `10` • `20` • `30` • `40` • `60` • `80` • `100` **Note:** 0 = Disabled

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
<codec > ptime (continued)	If this option is disabled, the SDP `ptime` attribute is not sent to the remote SDP unless the SDP offer had the SDP `ptime` attribute. Display name values for <codec> ptime: • AMR • AMR-WB • G.722 • G.726-32 • G.729 • GSM 6.10 • G.711 A-law (PCMA) • G.711 U-law (PCMU) • RFC2833 DTMF (Telephone-Event) **Notes:** AMR, AMR-WB and GSM 06.10 do not support the `10` and `30` millisecond durations. G.726-32 does not support the 80 and 100 millisecond durations. The packet interval does not always conform to the `ptime` for the bridging, conferencing, or ASR operations. However, packet size always conforms to the `ptime`. **Changes take effect:** Immediately.	

Maximum and Minimum Frequency of Segments	Specifies the accuracy of the minimum and maximum tone frequencies (in Hz) for CPA. The options names are configured as follows:
	`mpc.cpa.tone<m>.segment<n>.f1min`
	`mpc.cpa.tone<m>.segment<n>.f2min`
	`mpc.cpa.tone<m>.segment<n>.f1max`
	`mpc.cpa.tone<m>.segment<n>.f2max`
	Where `<m>` ranges from `1` to `10` for each tone, and `<n>` ranges `1` to `3` for each segment.
		A positive integer.

Default Audio Formats	The default audio format for the Call Manager.	• `ALAW` • `ULAW` **Default value:** `ULAW`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
SRTP Mode	The mode of operation with regard to Secure Real-Time Transport Protocol (SRTP). For `offer` mode: • If the other side ignores SRTP, the platform will fall back to non-SRTP mode. • If a previously negotiated m-line is used in a reoffer or if the far end requests an offer, and that m-line did not have SRTP negotiated, SRTP will not be added. • If the far end reoffers and adds SRTP to a previously negotiated m-line, SRTP will be negotiated.	• `none`—No SRTP support. The Media Control Platform will ignore the `crypto` attribute in SDP offers. • `accept_only`—SRTP is supported for SDP offers sent to the Media Control Platform, but the platform will not add SRTP to m-lines in outgoing offers that did not previously contain it. • `offer`—SRTP is supported for SDP offers sent to the Media Control Platform, and will be included in all outgoing SDP offers. • `offer_strict`—The Media Control Platform accepts SRTP received in the offer, and sends a crypto line in its own offer, but will fail if the answer does not contain a valid crypto line. • `offer_selectable`—Two media lines are offered for each media type, one with crypto, one without. If both media lines are accepted, all RTP is sent and received through the crypto line. **Default value:** `none`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax

RTP Send Mode	Specifies the output mode for outgoing RTP streams.
	Notes:
	• Continuous mode applies only for G.711 mulaw, G.711 alaw, AMR, and AMR-WB audio codecs.
	• Continuous mode does not apply in the following scenarios:
	  ◆ When a bridge transfer is in progress
	  ◆ When RTP data is sent to ASR speech engines.
	  ◆ When the RTP data contains video.
		• `continous`—Audio silence is sent when there is no data to send.
		• `vad`—RTP transmissions stop when there is no data to send.
		Default value: `vad`

IP Type of Service for RTP/RTCP	Specifies the IP differentiated services field (ToS) to set in all outgoing RTP/RTCP packets. **Notes:** • For Windows Server 2003, the ToS must be enabled in the registry. See `http://support.microsoft.com/kb/248611` • For Windows Server 2008, the ToS configuration is not supported. It must be configured at the OS level. You can define per executable and per port, and what type of DiffServ bits to set on the outgoing packets using the QoS policy defined in the following article. `http://technet.microsoft.com/en-us/library/cc771283.aspx` • For all Operating Systems, when the SIP/RTP packets are sent across different subnets, the router may reset the DiffServ bits in the IP header even though it was set by MCP.	Range: `0-255` Examples: • `0`—Disabled • `16`—IPTOS LOWDELAY (0x10) • `32`—IPTOS PREC PRIORITY (0x20) • `64`—IPTOS PREC CRITICAL (0x40) • `184`—DiffServ EF (Expedited Forward 0xB8) **Default value:** `0`
Maximum Record File Size	Specifies the maximum file size, in bytes, reached before the recording is stopped. If this option is set to `0`, disables this limit. **Note:** The recorded file may exceed this limit by a few hundred bytes depending on the codec and container chosen.	An integer range of 0–4,000,000,000. **Default value:** `0`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
SDP Origin Name Map [n]	Specifies the origin to match in the SDP. If the origin specified by this parameter matches the SDP, the DTMF type and `confgain` specified by DTMF Send Type [n] and Conference gain [n]. n= 0 to 9	`<FQDN or IP Address>/[session name content]` **Default value:** Empty
DTMF Send Type [n] for SDP Origin Name Map[n]	Specifies the DTMF type to use when SDP Origin Name Map [n] matches the SDP of the call. n=0 to 9	• `SIPINFO` • `INBAND` **Default value:** `INBAND`
Conference Gain [n] for SDP Origin Name Map [n]	Specifies the input gain percentage to apply for the SDP matching connection when joining a conference. n= 0 to 9	An integer range of 0–1000 **Default value:** `100`
RTP De-Jitter Delay	Specifies the duration, in milliseconds, of buffer time to allow for RTP packet inter-arrival dejittering. This translates to an initial delay before the packets are dispatched. If set to `0`, inter-arrival detector is disabled.	An integer range of 0–10000 **Default value:** `0`
RTP De-Jitter Timeout	Specifies the length of time, in milliseconds, that the RTP packets are to wait for the missing RTP packet. Once the timeout expires, the packets are dispatched without the missing packet.	An integer range of 0–1000 **Default value:** `200`
RTP/RSTP/RSTP RTP Port Range	Specifies the ports for MPC to use. **Note:** The Media Control Platform allocates local RTP port in a round-robin manner starting from the lowest port specified, and starting from the lowest port again when the highest port is reached.	A character string with possible values of 1030 to 65535. **Default value:** `10000-65535`
Local RTSP/RTP Address	Specifies the where the RTSP interface is located.	`<IP Address>` **Default value:** `$LocalIP$`
RTP Audio Buffer Size	Specifies the size of the buffer to be used for sending RTP audio data.	Any integer. **Default value:** Empty

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
RTP Video Buffer Size	Specifies the size of the buffer to be used for sending RTP video data. **Notes:** The higher frame rates and resolutions require larger values this parameter, but the default value should be big enough for MCP to play any frame rates and resolution. For H263 or H264 video file play, SQCIF, QCIF, and CIF resolution with 10, 15, and 30 frame rates have been tested with the default configuration.	Any integer. **Default value:** Empty
Media Manager Audio Buffer Size	Specifies the size of the buffer to be used for sending non-TTS audio data.	Any integer. **Default value:** `102400`
Media Manager Video Buffer Size	Specifies the size of the buffer to be used for sending non-TTS video data. **Notes:** The higher frame rates and resolutions require larger values this parameter, but the default value should be big enough for MCP to play any frame rates and resolution. For H263 or H264 video file play, SQCIF, QCIF, and CIF resolution with 10, 15, and 30 frame rates have been tested with the default configuration.	Any integer. **Default value:** Empty
Transcoders	Specifies the list of transcoders that will be used to provide transcoding services. The G.726 transcoder is loaded by default. If this option value is set to `none`, all transcoders are disabled.	Valid values: • `G.722` • `G.726` • `G.729` • `AMR` • `GSM` • `AMR-WB` • `MP3` • `H.263` • `H.264` • `None` **Default value:** `None`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
VoIP Metrics	Specifies whether to collect the metrics defined in RFC 3611 for each audio session. These metrics are divided into local and remote. For local metrics, MCP collects some by exchanging RTCP messages between itself and the remote party, and some are calculated locally from ongoing activities. For remote metrics, the remote party, if it supports RFC 3611, sends the metrics to MCP periodically. MCP records these metrics when it receives an update.	• `0`—Disable • `1`—Enable **Default value:** `0`
`msml` Section		

Beep Filename	Specifies the filename of the beep that is sent before the <join> operation.
	Example:
	`file://$InstallationRoot$/audio/ulaw/default_audio/endofprompt.vox`
		`$InstallationRoot$/`
		Default value:
		`file://$InstallationRoot$/audio/ulaw/default_audio/endofprompt.vox`

Beep File Time Limit in Join	Specifies the time limit, in milliseconds, for the audible beep when played during a <join> element.	An integer range of 1–10000. **Default value:** `5000`
CPD default Beep Timeout	Specifies the CPD beep timeout, in seconds, if the `<cpd>` element is not used in the VoiceXML application. **Note:** Setting this parameter to `0` disables the functionality.	An integer range of 0–60. **Default value:** `30`
CPD default Post-connect Timeout	Specifies the CPD post-connect timeout, in seconds, if the `<cpd>` element is not used in the VoiceXML application. **Note:** Setting this parameter to `0` disables the functionality.	An integer range of 0–60. **Default value:** `30`
CPD default Pre-connect Timeout	Specifies the CPD pre-connect timeout, in seconds, if the `<cpd>` element is not used in the VoiceXML application. **Note:** Setting this parameter to `0` disables the functionality.	An integer range of 0–60. **Default value:** `30`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Root Directory for Play Media	Specifies the path to the prompt media root directory.	A character string. **Default value:** `file://$InstallationRoot$`
Root Directory for Record Media	Specifies the path to the recording media root directory.	A character string. **Default value:** `file://$InstallationRoot$`
Root Directory for CPD Recording	Specifies the path to the CPD recording root directory.	A character string. **Default value:** `file://$InstallationRoot$/Record`
File Extension for CPD Recording	Specifies the CPD recording file extension that determines the MIME-type and extension to use.	A character string. **Default value:** `.wav`
Default Final Silence Timeout	Specifies the final silence duration, in seconds, in order to terminate the recording. **Changes take effect:** Immediately	An integer range of 0–10000. **Default value:** `4`
MSML INFO Allowed Content-Types	Specifies the content -types allowed in a SIP INFO messages for the MSML AppModule. Only the defined content types are processed, others are ignored.	An alphanumeric string of space delimited characters. **Default value:** `application/vnd.radisys.msml+xml`
Default Audio File Extension for Play Prompt and Recording	Specifies the default file extension of the audio files used in play prompt or recording.	A character string. **Default value:** `.wav`
	`Netann` **Section**	
Root Directory for Prompt Media	Specifies the path to the prompt media root directory.	A character string. **Default value:** `$InstallationRoot$/`
Root Directory for Recorded Media	Specifies the path to the record media root directory.	A character string. **Default value:** `$InstallationRoot$/record`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Send DTMF-Relay SIP Info Messages	Specifies whether to include prompt announcement services in the SIP header when receiving DTMF.	• `Auto` • `True` • `False` **Default value:** `Auto`
Root Directory for Record Media	Specifies the path to the recording media root directory.	A character string. **Default value:** `$InstallationRoot$/record/`
Maximum Allowed Silence Time During Recording	Specifies the maximum amount of silence, in seconds, allowed during a recording. If set to `0`, silence detection is not used.	Any integer. **Default value:** `0`
Maximum Recording Time	Specifies the maximum time, in seconds, allowed to record. If set to 0, the recording time is unlimited.	Any integer. **Default value:** `0`
Default Repeat Times for Play Netann Announcement Prompts	Specifies the default repeat times to be used for Netann announcement playback. **Note:** This parameter is not applicable to DTMF prompts.	A character string. **Default value:** `forever`
Conference Recording Mode	Specifies the recording mode when recording is enabled in a conference.	• `mixed`—The recorded file format will be specified by request with audio from all participants mixed into a single file. • `pcap`—One pcap format file will be created for each participant. **Default value:** `mixed`
List of H.263 Video Formats	Specifies, in a comma separated list, H.263 video formats that are used for selecting H.263 video files to play. H.263 video formats are: SQCIF=1 to 6 QCIF=1 to 6 CIF=1 to 6 CIF4=1 to 6 CIF16=1 to 6	A character string. **Default value:** `QFIC=2`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
`remdial` Section		
Remdial Port	Specifies the port used for remote dialing.	An integer in the range of 1025–65535. **Default value:** `6999`
Remdial Max Calls	Specifies the maximum number of concurrent remdial calls.	Any integer greater than zero. **Default value:** `500`
Remdial Max Client Sockets	Specifies the maximum number of remdial clients allowed to connect to the interface.	Any integer greater than zero. **Default value:** `64`
Remdial Telnet Mode	Specifies the telnet Operating System mode.	• `Auto`—Mode automatically selected based on the OS. • `RAW`—Windows OS mode. • `Normal`—Linux OS mode. **Default value:** `Auto`
`sip` Section		
(**Note:** For additional important options in this configuration section, see also “Configuring SIP Communication and Routing” on page 42.)		
Default Blind Transfer	The default transfer method for SIP, for blind transfers.	• `HKF`—Hookflash • `REFER`—REFER-based transfer • `BRIDGE`—Bridge-based transfer • `REFERJOIN`—Consultative `REFER` transfer
		• `MEDIAREDIRECT`—Media redirect transfer • `ATTCOURTESY`—AT&T In-band Courtesy transfer • `ATTCONSULT`—AT&T In-band Consult transfer • `ATTCONFERENCE`—AT&T In-band Conference transfer

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Default Blind Transfer (continued)		• `ATTOOBCOURTESY`—AT&T Out-of-Band Courtesy transfer • `ATTOOBCONSULT`—AT&T Out-of-Band Consult transfer • `ATTOOBCONFERENCE`—AT&T Out-of-Band Conference transfer **Default value:** `REFER`
Default Bridge Transfer	The default transfer method for SIP, for bridge-type transfers.	• `BRIDGE`—Bridge-based transfer • `MEDIAREDIRECT`—Media redirect transfer **Default value:** `BRIDGE`
Default Consultation Transfer	The default transfer method for SIP, for consult-type transfers.	• `HKF`—Hookflash • `BRIDGE`—Bridge-based transfer • `REFERJOIN`—Consultative `REFER` transfer • `MEDIAREDIRECT`—Media redirect transfer • `ATTCONSULT`—AT&T In-band Consult transfer • `ATTCONFERENCE`—AT&T In-band Conference transfer • `ATTOOBCONSULT`—AT&T Out-of-Band Consult transfer • `ATTOOBCONFERENCE`—AT&T Out-of-Band Conference transfer **Default value:** `REFERJOIN`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Default Gateway	The default gateway host and port that will be used for SIP calls (transfer, call, or remote dial) to a telephone, if the destination address does not specify a gateway. If this parameter is not specified, telephony calls that do not specify a gateway in the destination address will fail. **Example:** If `sip.defaultgw=pstn-gw.voiceplatform.com:5060` and a SIP call is placed to telephone number `123456789`, the SIP Line Manager translates the destination address to `sip:123456789@default-gw`, and the call is routed to port `5060` on host `pstn-gw.voiceplatform.com`.	`<Host name or IP address>:<SIP port>` **Default value:** Empty
Default Host	The default host and port that the Media Control Platform will use for SIP calls (transfer, call, or remote dial), if the destination address does not contain a host name or IP address. If this parameter is not specified, calls that do not specify a host in the destination address will fail. **Example:** If `sip.defaulthost=voiceplatform.com:5060` and a SIP call is placed to address `sip:1234@`, the destination address is translated to: `sip:1234@voiceplatform.com:5060`	`<Host name or IP address>:<SIP port>` **Default value:** Empty
Defer Out Alerting	Enables early media for an outbound call, by specifying whether the CallOutAlerting response to the session manager will be deferred until the media session is initialized and registered. If enabled, the session manager can start performing media operations on the channel as soon as the session manager receives the CallOutAlerting notification.	• `0`—CallOutAlerting will not be deferred. • `1`—CallOutAlerting will be deferred. **Default value:** `0`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
DNIS Correlation ID Length	The length of the correlation ID, within the `user-id` portion of the DNIS. The correlation ID is the portion of the `user-id` that will be stripped, in order to isolate the DNIS. **Note:** In the special case where the correlation ID is all of the `user-id`, the ampersand character (@) will also be stripped away from the DNIS, because `@<hostname>` does not make sense.	A non-negative integer. **Default value:** `0` (no correlation ID)
DNIS Correlation ID Offset	The offset that specifies where the correlation ID starts, within the `user-id` portion of the DNIS. The correlation ID is the portion of the `user-id` that will be stripped, in order to isolate the DNIS.	Any integer. A negative value indicates that the offset is from the right. **Default value:** `0` (no offset)
Enable Send/Receive Events	Enables the sending and receiving of SIP `INFO` messages for VoiceXML application usage. This parameter does not affect SIP `INFO` messages used for other purposes (for example, DTMF). **Changes take effect:** Immediately.	• `True`—VoiceXML applications are enabled to send and receive SIP `INFO` messages. • `False`—VoiceXML applications cannot send and receive SIP `INFO` messages. **Default value:** `True`
Enable SDP answer in provisional response	Specifies whether to send an SDP answer in the reliable provisional response if the INVITE contains an SDP offer. **Note:** Applies only if `Enable Reliable Provisional Responses` is set to `Supported` or `Required`, or if `Send Alert` is set to `2`.	• `True`—The MCP includes the SDP answer. • `False`—The MCP does not include the SDP answer. **Default value:** `True`
Enable Reliable Provisional Responses	Specifies whether to allow the SIP stack to send reliable 101-199 provisional responses. If set to `1`, the `100rel` extension is included in the header of the outbound INVITE request giving the remote end the option to send the reliable provisional response. If set to `2`, the MCP includes the `100rel` extension in the `Require` header of the outbound INVITE forcing the remote end, that supports PRACK, to send the reliable provisional response.	• `0`—Disabled • `1`—Supported • `2`—Required **Default value:** `0`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
HF Disconnect Type	The timeout value, in milliseconds, to terminate a SIP hookflash transfer. • If `sip.hftype=0` (wait for disconnection), the transfer is treated as failed if a `BYE` is not received from the remote end before this timeout expires. • If `sip.hftype=1` (force disconnection), the transfer is always treated as successful. If a `BYE` is not received from the remote end before this timeout expires, then a `BYE` will be sent from the local end.	Any non-negative integer. **Default value:** `5000`
HF Prefix	The SIP hookflash transfer dialing prefix. **Examples:** • `sip.hfprefix=none` means the dial string is exactly as specified in the transfer. • `sip.hfprefix=!` means dial a hookflash. • `sip.hfprefix=*8,` means dial `*8` followed by two pause durations.	A string that contains one or more of the following characters: `0–9, ! * none` **Default value:** `!`
HF Stop Dial	The digits to dial to stop a hookflash transfer. Dialing the digits specified in this parameter will abort a multi-phase hookflash. The connection is switched back to the original caller.	A string that contains one or more of the following characters: `0–9 !` **Default value:** `!`
Hook Flash Transfer Type	Specifies the type of hookflash transfer for SIP.	• `0`—Wait for disconnection. • `1`—Force disconnection. **Default value:** `0`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Customer Inbound <SIP request> Headers	The list of header names from incoming `<SIP request>` messages that will be exposed to the VoiceXML application, where `<SIP request>` is one of: • `BYE` • `INFO` • `INVITE` The names of the exposed headers appear in the application in the following format: `sip.invite.<headername>=<value>`	`<Header1> [<Header2>...]` where `<HeaderX>` is: • A header name—Each specified header name will be exposed. • `*`—All header names will be exposed. • `none`—No header names will be exposed. If any other value is specified alongside `none`, `none` is ignored. **Example:** `From To Via` **Default values:** • For `BYE` requests: `Reason` • For `INFO` and `INVITE` requests: `*`
Custom Inbound Invite Parameters	The list of header names from incoming `INVITE` requests whose parameters will be exposed to the VoiceXML application. The exposed parameter values appear in the application in the following format: `sip.invite.<headernam>.<paramname>=<value>`	`<Header1> [<Header2>...]` where `<HeaderX>` is: • A header name—Each specified header name will be exposed. • `none`—No header names will be exposed. If any other value is specified alongside `none`, then `none` is ignored. **Default value:** `RequestURI`
INFO Request Content-Type	The content type of outgoing SIP `INFO` messages that correspond to VoiceXML application `<log>` events. A VoiceXML application can trigger the sending of a SIP `INFO` message by using the `<log>` tag with `dest="callmgr"`. Call Manager will then send a SIP `INFO` message to the remote end. The content of the SIP `INFO` message is the content of the `<log>` tag.	A string indicating the content type. **Default value:** `application/text`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Local RTP Address	The Media Control Platform IP address to advertise for Real-time Transport Protocol (RTP). With multicast or proxied systems, you may need to specify what IP address to advertise in the SDP description for a session. By default, the IP address of the local system is retrieved by performing a standard `gethostname()`. However, if your system is multi homed or behind a firewall, use this parameter to control the IP address that is advertised.	`<IP address>` **Default value:** Empty (which causes the local IP address to be determined automatically)
P-Asserted-Identity Header	Specifies whether the `P-Asserted-Identity` header is used as the ANI if it is included in the incoming SIP INVITE, and its value is exposed with the `session.connection.remote.uri` session variable. If not, the `From` header is used.	• `0`—Do not use P-Asserted-Identity for ANI • `1`—Use P-Asserted-Identity for ANI **Default value:** `1`
P-Called-Party-ID Header	Specifies whether the P-Called-Party-ID header is used as the DNIS if it is included in the incoming SIP INVITE, and its value is exposed with the `session.connection.local.uri` session variable. If not, the `To` header is used.	• `0`—Do no use P-Called-Party-ID for DNIS • `1`—Use P-Called-Party-ID for DNIS **Default value:** `1`

P-Alcatel-CSBU Header Value	Specifies the P-Alcatel-CSBU header value of the `200OK` response to the initial incoming `INVITE` if the request contains this header. If this parameter is an empty string, no header is set.	A character string.
		Default value:
		`fb=notransfer;`
		`dtmf_auto=on`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Custom Outbound <SIP request> Headers	The list of header names from outgoing `<SIP request>` messages that will be exposed to the VoiceXML application, for customization. `<SIP request>` is one of: • `INFO` • `INVITE` • `REFER` The customized names of the exposed headers appear in the application in the following format: `sip.invite.<headername>=<value>`	`<Header1> [<Header2>...]` where `<HeaderX>` is: • A header name—Each specified header name will be exposed. • `*`—All header names will be exposed. • `none`—No header names will be exposed. If any other value is specified alongside `none`, `none` is ignored. **Example:** `From To Via` **Default value:** `*`
Custom Outbound <SIP request> Params	The list of header names from outgoing `<SIP request>` messages whose parameters will be exposed to the VoiceXML application, for customization. `<SIP request>` is one of: • `INVITE` • `REFER` The exposed parameter values appear in the application in the following format: `sip.invite.<headername>.<paramname>=<value>`	`<Header1> [<Header2>...]` where `<HeaderX>` is: • A header name—Each specified header name will be exposed. • `none`—No header names will be exposed. If any other value is specified alongside `none`, `none` is ignored. **Default value:** `RequestURI`
Route Set	Specifies the route set for non-secure SIP outbound calls. If defined, this route set is inserted as the ROUTE header for all outgoing calls and forces the MCP to send the SIP messages through this defined route set. Each element in the `routeset` must be separated by commas. For example, `sip.routeset=<sip:p1.example.com;lr>,<sip:p2.domain.com;lr>` **Note:** This parameter does not apply to SIP REGISTER messages.	Any string of characters. **Default value:** Empty

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Secure Route Set	Specifies the route set for secure SIP outbound calls. Secure SIP calls must specify the `sips` scheme or `tls` transport parameters. If defined, this route set is inserted as the ROUTE header for all outgoing calls and forces the MCP to send the SIP messages through this defined route set. Each element in the `routeset` configuration option must be separated by commas. For example, `sip.securerouteset=<sips:p1.example.com;lr>,<sips:p2.domain.com;lr>` **Note:** This parameter does not apply to SIP REGISTER messages.	Any string of characters. **Default value:** Empty
SIP Static Route List	Specifies, in a pipe delimited list, the static route groups. Each route group contains a list, separated by commas, of IP addresses. Within the route group, each IP address may substitute each other as an alternate route destination if sending a SIP request to one of the IP address that fails. For example, `10.0.0.1,10.0.0.2|10.0.10.1,10.0.10.2` specifies two static route groups, and each group specified two routes that are alternate to each other.	Any string of characters. **Default value:** Empty
Use Original Gateway in Outbound Call	Specifies how the Media Control Platform will determine which gateway to use for an outbound call or transfer, if the destination address does not contain a host name or IP address. **Example:** If `sip.outcalluseoriggw=1` and the inbound call came from a gateway with host name `3000,` the call will be placed to one of the following: • `tel://3000` • `sip:3000@`—The ampersand character (`@`) is required to delimit the user part from the host part of the address.	• `0`—The gateway specified in `sip.defaultgw` or `sip.defaulthost` will be used. • `1`—The gateway of the inbound call will be used. **Default value:** 1
Refer Transfer Hold	Specifies whether to put the originating caller on hold (Invite hold) before the Media Control Platform sends the `REFER` message for a `REFER` or `REFERJOIN` transfer.	• `0`—Original caller will not be put on hold. • `1`—Original caller will be put on hold. **Default value:** 1

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Refer Transfer Retry REFER on the Outbound Leg	Specifies the action to take if the caller (or its Media Gateway) cannot handle the REFER request. If the Caller (or its Media Gateway) cannot handle the REFER request, the transfer will fail. When failed, MCP will send REFER with Replaces to the Agent instead (hoping the Agent can establish direct connection to the Caller, when the Caller cannot do so).	• `0`—Disabled • 1—Enabled **Default value:** `0`
Registration	The settings for registering the Media Control Platform with the SIP Registrar. You can configure the system to register with one or more SIP registration servers on the network. To specify more than one registration entry, separate the entries with a pipe (|). The Media Control Platform will attempt to register with all defined registration entries, and will periodically reregister as required (in accordance with the `<requested-expiry>` parameter). The Media Control Platform will de-register when it shuts down.	`<registration-server> <register-as> <requested-expiry> <username> <password> [<routeset>]` where: • `<registration-server>` is the host and port of the Resource Manager or other SIP registration server.
	Example: `ResourceManager.yourdomain.com:5064 mcp@10.0.0.101 60 - -|proxy2.yourdomain.com:5064 mcp@10.0.0.102 60 user password` means that the Media Control Platform will register with the Resource Manager as SIP user `mcp@10.0.0.101`, and with another SIP proxy as SIP user `mcp@10.0.0.102`, with authentication user name (`user`), and password (`password`).	• `<register-as>` is the SIP identity of the Media Control Platform. • `<requested-expiry>` is the duration of registration, in seconds. `<username>` is the user name when authentication is required by the server. This may or may not be the same as `<register-as>`. A dash (–) indicates that no user name is needed. If `username`=— (dash) and the server requests authentication, `Anonymous` is used.

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Registration (continued)		• `<Password>` is the password associated with the authentication user name. To specify an empty string, use a dash (–). • `<Routeset>` is a comma-separated list of the servers that the `REGISTER` messages will go through. If a route set is not defined, the `REGISTER` messages will be sent directly to the `<registration-server>`. **Default value:** Empty
Send Alert	The SIP response for alerting and intermediate provisional responses. **Changes take effect:** Immediately	• `0`—No SIP response • `1`—Send `180 RINGING` response • `2`—Send `183 Session Progress` response with SDP information **Default value:** `1`
INFO Allowed Content-Type	A space-delimited list of the content types that are allowed to be passed up to the VoiceXML application level in a SIP `INFO` message. Any content types that have not been defined will be ignored.	`<Content type1>[<Content type2>...]` where `<Content typeN>` is: • An alphanumeric string—Defines the content type. • An empty string—Allows all content to be passed upstream. **Default value:** `application/text`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Transfer Methods	A space-separated list of the supported transfer methods for SIP.	• `HKF`—Hookflash • `REFER`—REFER-based transfer • `REFERJOIN`—Consultative `REFER` transfer • `MEDIAREDIRECT`—Media redirect transfer • `none`—No transfer methods for SIP • `ATTCOURTESY`—AT&T In-band Courtesy transfer • `ATTCONSULT`—AT&T In-band Consult transfer • `ATTCONFERENCE`—AT&T In-band Conference transfer • `ATTOOBCOURTESY`—AT&T out-of-band courtesy transfer • `ATTOOBCONSULT`—AT&T out-of-band consult transfer
Transfer Methods		• `ATTOOBCONFERENCE`—AT&T out-of-band consult transfer **Default value:** `REFER REFERJOIN MEDIAREDIRECT ATTCOURTESY ATTCONSULT ATTCONFERENCE ATTOOBCOURTESY ATTOOBCONSULT ATTOOBCONFERENCE`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
VoiceXML URL INVITE	Specifies whether VoiceXML URLs in SIP `INVITE` messages will be accepted, thereby bypassing the normal method of selecting a VoiceXML application on the basis of DNIS mapping. If `vxmlinvite` is enabled, the originator of a SIP call can specify the initial VoiceXML URL that will be fetched for the session. To implement this functionality, the originator of the SIP call must encode the `Request-URI` in the following special form: `"sip:dialog.vxml.<URL>@host.com"` where the `<URL>` portion is encoded (for example, `%3A`).	• `0`—VoiceXML URLs will not be accepted. • `1`—VoiceXML URLs will be accepted. **Default value:** `1`
Warning Headers	Specifies whether the Media Control Platform will send warning headers. **Changes take effect:** Immediately.	• `0`—The Media Control Platform will send warning headers only when it receives an error response. • `1`—The Media Control Platform will always send warning headers, if there are any. • `2`—The Media Control Platform will never send warning headers. **Default value:** `0`
Transfer Copy Headers	A space-delimited list of the headers to be copied from inbound call `INVITE` requests to outbound call `INVITE` requests for the same VoiceXML session (in other words, for bridged and Release Link Transfer [RLT] calls). The headers are re-scanned for the re-`INVITE` (the outbound call `INVITE` request), so changes that have been made to the values of the headers during the inbound call leg are applied on any outbound calls made within the call session. **Changes take effect:** Immediately.	`<Header1> [<Header2>...]` where `<HeaderX>` is: • A header name—Each specified header will be copied. • `*`—All headers will be copied, including unknown headers. • `none`—No headers will be copied. **Default value:** `*`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
IP Type of Service for Transport	Specifies the IP differentiated services field (ToS) to set in all outgoing SIP packets over the SIP transport. **Note:** This configuration parameter is not valid on a Windows 2008 operating system.	Range: `0-255` Examples: • `0`—Disabled • `16`—IPTOS LOWDELAY (0x10) • `32`—IPTOS PREC PRIORITY (0x20) • `64`—IPTOS PREC CRITICAL (0x40) • `184`—DiffServ EF (Expedited Forward 0xB8) **Default value:** `0m`
Transport Instance 0	Specifies the transport layer for the SIP stack and the network interfaces that are used to process SIP requests. This option uses the following format: `sip.transport.x = transport_name type: ip:port [parameters]` Where: • `transport_name`—Is any string. • type—Is UDP, TCP, or TLS. • ip—Is the IP address of the network interface that accepts incoming SIP messages. • `port`—Is the port number where the SIP stack accepts incoming SIP messages. • `[parameters]`—Defines any extra SIP transport parameters. This is used for LMSIP2.	A string. **Default value:** Empty

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Transport Instance 0 (continued)	If `ip` is an IPv6 address, `[]` must be used. To define a transport to listen to all IPv4 interfaces, use `any` or `any4` for `ip`. To define a transport to listen to all IPv6 interfaces, use `any6` for `ip`. For example: `cert=[cert path and filename]`—The path and the filename of the TLS certificate to be used. Applicable to SIPS only and mandatory if SIPS is used. `key=[key path and filename]`—The path and the filename of the TLS key to be used. Applicable to SIPS only and mandatory if SIPS is used. `type=[Type of secure transport]`—The type of secure transport to be used. Value can be TLSv1, SSLv2, SSLv3, SSLv23. Default value is `SSLv23`. Applicable to SIPS only and optional. `password=[password]`—The password associated with the certificate and key pair. Required only if key file is password protected. Applicable to SIPS only and optional. `cafile=[CA cert path and filename]`—The path and the filename of the certificate to be used to verify the peer. The same certificate that is specified in the `cert=[cert path and filename]` parameter can be used as the value here if only one certificate is preferred. `verifypeer=true`—Turns on TLS mutual authentication. Mandatory for TLS mutual authentication. `verifydepth=[max depth for the certificate chain verification]`—Sets the maximum depth for the certificate chain verification. The recommended value is `1` for the default Genesys certificate that is provided. Applicable to TLS mutual authentication only.	

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Transport Instance 0 (continued)	The default transport is the smallest non-empty ID. If all `transport.x` values are empty, the UDP, TCP, and TLS transports are all enabled and ports 5060, 5060 and 5061 respectively, listen on any network interface. TLS transport uses the `certificate`, `x509_certificate.pem`, and `key`, `x509_private_key.pem`, in the `config` directory and UDP is the default transport.	
Transport Instance 1	See the description for Transport Instance 0.	A string. **Default value:** Empty
Transport Instance 2	See the description for Transport Instance 0.	A string. **Default value:** Empty
Preferred IP version to be used in SIP	Specifies the connection timer bucket depth. If this parameter is set to a higher value, the initial memory usage increases, but the allocation of run-time memory for high loads is prevented, thereby enhancing performance and stabilizing memory at a lower mark. The default for this section is set to the maximum value of `3000` for performance reasons.	A numeric string. **Default value:** `3000`
Local Transport IPv6 Address	Specifies whether or not the `sip.transport.localaddress` configuration option contains an `SRV` domain name. • If this option is set to true, the port part is not automatically generated by the SIP stack. • If this option is set to false, the port of the outgoing transport is used together with the host name that is specified by the `sip.transport.localaddress` option.	• true • `false` **Default value:** `false`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
sessmgr Section		
Send SDP in INVITE for Media Redirect	Specifies whether to send the caller's last SDP to the called party for Media Redirect calls. For NGI applications: • If this option is set to `true`, if a call has `connectwhen` specified as `answered`, MCP will send the caller's last SDP in the re-`INVITE` and the `ACK` message. • If this option is set to `false`, MCP will not send the caller's last SDP in the re-`INVITE`. For GVPi applications: • If this option is set to `true`, and the transfer is a 2 leg transfer, if a call has `connectwhen` specified as `answered`, MCP will send the caller's last SDP in the re-`INVITE` and the `ACK` message. **Note:** For backward compatibility with legacy devices, which always require SDP in INVITE, Media Control Platform in the default configuration does not fully conform to RFC3264 when performing a Media Redirect Transfer. In particular, MCP will send a SDP offer in INVITE and will send an updated SDP in the ACK. If full offer/answer behavior is desired, and legacy devices are not involved, Genesys recommends setting this parameter to `False`.	• `True` • `False` **Default value:** `True`
Accept Call Timeout	Specifies the time, in milliseconds, to wait after an alert is issued when the application module does not accept the inbound call before disconnecting it.	Any integer. **Default value:** `30000`
tts Section		

TTS Engine Default	Specifies the default TTS Engine resource when using Request URI.
	Changes take effect: Immediately
	For example, `REASPEAK?MRCPv2.`
		`<resourcename>`
		`[?<protocol>]`
		Default value: Empty

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
`vrm` **Section**		
FIPS Enabled	Specifies whether to enable FIPS mode in MRCP Proxy. When FIPS mode is enabled, only FIPS 140-2 approved ciphers and algorithms can be used in SSL connections.	• `true` • `false` **Default value:** `false`
Native DTMF Grammar Maxage	Specifies the Maxage, in milliseconds, that a native DTMF recognizer uses to fetch an external grammar.	An integer value of -1 indicates the server's maxage value is used. **Default value:** `-1`
Native DTMF Grammar Maxstale	Specifies the Maxstale, in milliseconds, that a native DTMF recognizer uses to fetch an external grammar. A value of -1 indicates to use the server's maxstale value.</	An integer value of -1 indicates the server's maxstale value is used. **Default value:** `-1`
SRM Default Response Timeout	The timeout interval, in milliseconds, for the MRCP client to wait for a response from the MRCP server. If no response is received within this timeout period, the request is deemed to have failed.	An integer in the range of 1–60000. **Default value:** `10000`
SRM Ping Frequency	The interval, in milliseconds, at which the MRCP Client pings each MRCP server that has been provisioned. The MRCP `DESCRIBE` method is used as a ping message.	An integer in the range of 1–3000000. **Default value:** `30000`
SRM Ping Timeout	The timeout interval, in milliseconds, for the MRCP client to wait for a ping response from the MRCP server. If no response is received within this timeout period, the MRCP server is considered to be unavailable. The MRCP Client disconnects from the server, and then periodically tries to re-establish a connection, at a retry interval specified in the `client.ping.frequency` parameter. Genesys recommends setting the `client.ping.timeout` value to twice the value of the `client.ping.frequency` parameter.	An integer in the range of 1–6000000. **Default value:** `60000`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Universals Grammar URI	The URI convention that the NGI uses to specify the universals grammars.	`builtin:grammar/universals` **Default value:** `builtin:grammar/universals`
	`vxmli` Section	
Release ASR Engines on Transfer	Specifies that for successful transfers, without speech grammars loaded, the interpreter will release all open ASR engines.	• `True` • `False` **Default value:** `True`
Strict Grammar Mode	Specifies whether the NGI will follow the VoiceXML specification strictly when handling the grammar element. The default value (`false`) means that the NGI will ignore the `mode` attribute for an external grammar.	• `True` • `False` **Default value:** `False`
Enable Real Time Debugging	Enables real-time debugging for the platform.	• `True` • `False` **Default value:** `False`
Initial Request Method	The HTTP method to use for the initial request.	• `GET` • `POST` **Default value:** `GET`
Maximum Subdialog Depth	Specifies the maximum number of dialogs that are allowed in a VoiceXML session. The depth increments when a subdialog is entered, and the depth decrements when a subdialog is returned.	An integer range of 1–1000. **Default value:** `50`
Maximum bytes of total saved temp files per session	Specifies the maximum number of bytes that are allowed for the total saved temp files per session. If the limit is exceeded, saving the temp files is disabled for the applicable session.	An integer range of 0–2 GB. **Default value:** `100 MB`
Maximum Size of VXML Document	Specifies the maximum size (in bytes) that is allowed for a VoiceXML document. If the limit is exceeded, the interpreter will generate a `error.badfetch` event. **Note:** Setting the parameter to `0` disables the functionality.	An integer range of 0–1 GB. **Default value:** `0`

Table 23: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Maximum Size of Script File	Specifies the maximum size (in bytes) that is allowed for a script file. If the limit is exceeded, the interpreter will generate a `error.badfetch` event. **Note:** Setting the parameter to `0` disables the functionality.	An integer range of 0–1 GB. **Default value:** `0`
Maximum Size of XML/JSON data	Specifies the maximum size (in bytes) that is allowed for XML or JSON data. If the limit is exceeded, the interpreter will generate a `error.badfetch` event. **Note:** Setting the parameter to `0` disables the functionality.	An integer range of 0–1 GB **Default value:** `0`
Enable External Messaging Within VoiceXML	Specifies whether the application can access messaging. **Note:** If set to `False`, executing a `<send>` or `<receive>` will result in an `error.unsupported.send` event or `error.unsupported.receive` event.	• `True` • `False` **Default value:** `True`
Transfer Allowed	Specifies whether dialog-initiated transfers are allowed.	• `True` • `False` **Default value:** `True`
Userdata Prefix	The string that, when used as a prefix in a SIP header, identifies userdata variables.	Any string. **Default value:** `X-Genesys-`

Important MRCP Server Configuration Options

This section describes important configuration options that you either must or may want to customize.

Configure the options in Genesys Administrator on the `Provisioning > Environment > Applications > <MRCP Server> > Options` tab. For the detailed steps to configure option settings, see Procedure: Viewing or modifying GVP configuration parameters, on page 34.

The configurable MRCP server options are in the `provision` configuration section. Table 24 provides information about these options. Table 23 provides parameter descriptions as well as the default parameter values that are preconfigured in the MRCPv1_ASR, MRCPv1_TTS, MRCPv2_ASR, and MRCPv2_TTS `Application` objects.

All changes take effect on restart.

For information about all the available configuration options for the MRCP servers, see the *Genesys Voice Platform 8.1 Configuration Options Reference.*

Table 24: Selected MRCP Server Configuration Options

Option Name	Description	Valid Values and Syntax
	`ASR and TTS`	
New MRCP Connection Per Session	(For MRCPv2 only) Specifies whether the MRCP Client will create a new connection to the ASR or TTS server for each MRCP session setup.	• `True` • `False` **Default value:** `True`
Vendor Name	The name of the speech resource vendor.	`<vendor_name>` **Default value:** Empty
Speech Resource URI	The URI to the speech resource.	• For MRCPv1 ASR: `rtsp://<MRCP server IP>:<port>/media/speechrecognizer` • For MRCPv1 TTS: `rtsp://<MRCP server IP>:<port>/media/speechsynthesizer` • For MRCPv2: `sip:mresources@<MRCP server IP>:<port>` **Default value:** Empty
	`ASR Only`	
ASR Resource Reservation	Specifies whether or not the MCP reserves an ASR resource prior to accepting the call. This resource is available until the resource is explicitly released, or until the end of the call. The call is rejected if the resource is not successfully reserved.	• `true` (enabled) • `false` (disabled) **Default value:** `false`

Table 24: Selected MRCP Server Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Disable Hotword Recognition	Specifies whether or not the platform treats recognition-based barge-in as speech-based barge-in. Set this parameter to `true` for all ASR servers that do not support recognition-based barge-in.	• `true`—Recognition-based barge-in will be treated as speech-based. • `false`—Recognition-based barge-in will not be treated as speech-based. **Default value:** `false`
HotKey Base Path	The HTTP fetchable location for the hotkey grammars. The value of this parameter is concatenated with the IP address of the Media Control Platform to form a fetchable location for hotkey grammars. The `<vendor name>` in the path must be the same as the vendor name that is specified in `vrm.client.resource.name` on page 194.	`/mcp/$AppName$/grammar/common/hotkey` **Default value:** Empty
HotKey Local Path	The local path for the hotkey grammars on the Media Control Platform. The MRCP Client uses the HotKeyBasePath to translates this address to the appropriate URI, which is sent to the ASR servers.	`$InstallationRoot$/grammar/<vendor name>/hotkey` **Default value:** Empty
Enable Silence Filling	Specifies whether to send silence audio during an ASR recognition session pause period.	• `True`—Send silence audio to the MRCP server. • `False`—Does not send silence audio to the MRCP server. **Default value:** `True`
	`TTS Only`	
TTS Resource Reservation	Specifies whether or not the MCP reserves an TTS resource prior to accepting the call. This resource is available until the resource is explicitly released, or until the end of the call. The call is rejected if the resource is not successfully reserved.	• `true` (enabled) • `false` (disabled) **Default value:** `false`

Note: Media Control Platform supports various MRCP vendors, and provides select vendor specific templates that have pre-populated parameters for your convenience. Some of those parameters may not have defaults listed, as the so-called default values are provided by the templates and not by software (in the event a parameter is manually deleted)

Chapter

8 Configuring the MRCP Proxy

The Genesys Voice Platform (GVP) Media Resource Control Protocol (MRCP) Proxy component acts as proxy for all MRCPv1 traffic, residing between the Media Control Platforms and the MRCPv1 resources.

This chapter provides information about configuring the MRCP Proxy in the following sections:

Task Summary: Configuring the MRCP Proxy

Task Summary: Configuring the MRCP Proxy summarizes the configuration tasks that are required to implement MRCP Proxy functionality in your GVP deployment.

Task Summary: Configuring the MRCP Proxy

Objective	Related Procedures and Actions
Complete the prerequisites.	1. Create the ASR and TTS speech resource `Application` objects If you have not already done so, see the Procedure: Provisioning ASR and TTS resources, on page 150.
	2. In the Media Control Platform `Application`, create a server connection to the MRCP Proxy. To create server connections, see the procedure in Chapter 7 of the *Genesys Voice Platform 8.1 Deployment Guide*.
Configure the MRCP Proxy `Application`	1. Configure the client-side connections. See "Configuring Client-Side Connections" on page 72.
	2. Create the server connections to: • The ASR and TTS speech resource access points that will be used by this proxy. • The Reporting Server • The SNMP Master Agent (optional) To create server connections, see the procedure in Chapter 7 of the *Genesys Voice Platform 8.1 Deployment Guide*.
	3. Configure the `[vrmproxy] uri` option with URI that the Media Control Platform uses to contact the MRCP Proxy. If the MRCP Proxy and Media Control Platform are installed on the same host, retain the default value for this option. Otherwise, configure the host part with the actual IP address of the MRCP Proxy.

Task Summary: Configuring the MRCP Proxy for HA

Task Summary: Configuring the MRCP Proxy for HA summarizes the configuration tasks that are required to implement MRCP Proxy for High Availability (HA).

Task Summary: Configuring the MRCP Proxy for HA

<table>
<tr><th>Objective</th><th>Related Procedures and Actions</th></tr>
<tr><td rowspan="3">Complete the prerequisites.</td><td>1. Ensure that the Solution Control Server (SCS) <code>Application</code> is configured to support HA licenses:
For a description of how to create and configure the license files, see the Framework 8.1 Deployment Guide and the Framework 8.1 Management Layer User's Guide.
Note: To support HA mode, you must ensure that the latest versions of Management Framework and LCA are installed. In addition, the Solution Control Server (SCS) must have an HA license. If the SCS is not licensed, it cannot provide HA functionality.</td></tr>
<tr><td>2. Create the ASR and TTS speech resource <code>Application</code> objects
If you have not already done so, see the Procedure: Provisioning ASR and TTS resources, on page 150.</td></tr>
<tr><td>3. In the Media Control Platform <code>Application</code>, add a connection to the primary MRCP Proxy.
To create server connections, see the procedure in Chapter 7 of the Genesys Voice Platform 8.1 Deployment Guide.</td></tr>
<tr><td rowspan="3">Configure the primary MRCP Proxy <code>Application</code></td><td>1. Create the server connections to:
• The ASR and TTS speech resource access points that will be used by this proxy.
• The Reporting Server
• The SNMP Master Agent (optional)
To create server connections, see the procedure in Chapter 7 of the Genesys Voice Platform 8.1 Deployment Guide.</td></tr>
<tr><td>2. Configure the <code>[vrmproxy] uri</code> option with URI that points to the MRCP Proxy.
If the MRCP Proxy and Media Control Platform are installed on the same host, retain the default value for this option. Otherwise, configure the host part with the actual IP address of the MRCP Proxy.</td></tr>
<tr><td>3. In the <code>Server Info</code> section, add the backup MRCP Proxy in the <code>Backup Server</code> field.</td></tr>
</table>

Task Summary: Configuring the MRCP Proxy for HA (Continued)

Objective	Related Procedures and Actions
Configure the backup MRCP Proxy `Application`	Complete the same steps as you did for the primary MRCP Proxy. See "Configure the primary MRCP Proxy `Application`", (Steps 1 and 2 only) in this table. **Note:** The connections must be the same for both the primary and backup proxy.

Important MRCP Proxy Configuration Options

This section describes the key configuration options that you either must or may want to customize.

Configure the options in Genesys Administrator on the `Provisioning > Environment > Applications > <MRCP Proxy> > Options` tab. For the detailed steps to configure option settings, see Procedure: Viewing or modifying GVP configuration parameters, on page 34.

The configurable MRCP Proxy parameters are in the following configuration sections:

- `vrmproxy`—Parameters that contain connection information for MRCP Proxy such as, IP address and port number, and application session timers.
- `stack`—Parameters that determine the connection timeouts, trace behavior, and RTSP port ranges.
- `ems`—Parameters that determine Reporting behavior.
- `log`—Parameters that determine the behavior for Management Framework logging.
- `snmp`—Parameters that determine the behavior of SNMP.

Table 25 provides parameter descriptions as well as the default parameter values that are preconfigured in the MRCP Proxy `Application` object.

Unless indicated otherwise, all changes take effect immediately.

For a complete list of MRCP Proxy configuration options and their descriptions, see the *Genesys Voice Platform 8.1 Configuration Options Reference.*

Table 25: Selected MRCP Proxy Configuration Options

Option Name	Description	Valid Values and Syntax
ems Section		
MF Sink Metrics Filter	Specifies the metrics that are delivered to the MF Sink. An asterisk in the value (*) indicates that all metrics will be sent to the sink. Alternatively, 5-8,50-55,70,71 indicates that metrics with IDs 5,6,7,8,50,51,52,53,54,55,70 and 71 will be sent to the MF sink.	A string of characters in the format of a comma-separated list of values or ranges. A metric value must be between 0 and 141 inclusive, but values '*' and blank are also allowed. **Default value:** *
MF Sink Log Filter	Specifies how the log messages that are sent to the MF sink are controlled. The values between pipes can be in the format: m-n,o,p (for example, 0-4, 5,6). The wildcard character * (asterisk) can also be used to indicate all valid numbers. For example, *|*|* indicates that all log messages should be sent to the sink. Alternatively, 0,1|0-10|*|4|*|* indicates that CRITICAL(0) and ERROR(1) level messages with module IDs in the range 0-10 will be sent to the sink; as well as all INFO(4) level messages.	A pipe-delimited range or string of characters for log levels, module IDs and specifier IDs in the format: levels|moduleIDs|specifierIDs (repeated if necessary). **Default value:** *|*|*
Persistent DB File for CDR Data	Specifies the full path of the local database file that is used to locally persist data for CDRs.	A character string. **Default value:** cdrQueue_rm.db Changes take effect: start/restart
Persistent DB File for OR Data	Specifies the full path of the local database file that is used to locally persist data for Operational Reporting.	A character string. **Default value:** orsQueue_rm.db Changes take effect: start/restart
CDR Batch Size	Specifies the number of CDR messages that can be queued up by the Reporting Client before they are sent to the Reporting Server. Larger batch sizes (for example, 50 records) lessen bandwidth constraints, at the cost of making and sending CDR data at larger intervals.	An integer between 1-5000 inclusive. **Default value:** 500 Changes take effect: start/restart

Table 25: Selected MRCP Proxy Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
OR Batch Size	Specifies the number of OR messages that can be queued up by the Reporting Client before they are sent to the Reporting Server.	An integer between `1-5000` inclusive. **Default value:** `500` Changes take effect: start/restart
OR Reporting Interval	Specifies the interval, in seconds, between the accumulation of operational reports that are submitted to the Reporting Server.	An integer between `1-299` inclusive. **Default value:** `60` Changes take effect: start/restart
Maximum Records in the Persisted Local DB File for CDR Data	Specifies the maximum number of data items to the local database for CDR reporting. Queuing occurs either when the Reporting Server is unavailable, or when data is provided to the client faster than the Reporting Server can consume it. The default value `-1` indicates an unlimited number of records are allowed. A value of `0` indicates that no records are persisted locally and data is discarded if the Reporting Server is unavailable.	An integer greater or equal to `-1`. **Default value:** `-1` Changes take effect: start/restart
Maximum Records in the Persisted Local DB File for OR Data	Specifies the maximum number of data items to the local database for CDR reporting. Queuing occurs either when the Reporting Server is unavailable, or when data is provided to the client faster than the Reporting Server can consume it. The default value `-1` indicates an unlimited number of records are allowed. A value of `0` indicates that no records are persisted locally and data is discarded if the Reporting Server is unavailable.	An integer greater or equal to `-1`. **Default value:** `-1` Changes take effect: start/restart

Table 25: Selected MRCP Proxy Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
TLS Certificate for Reporting Client	Specifies the file name of the TLS certificate in PEM format. The certificate is required to make the connection to the Reporting Server (ActiveMQ) over TLS.	A string of characters. **Default value:** `$InstallationRoot$/config/MRCPPROXY_EMStoMFLogID.txt` Changes take effect: start/restart
`log` Section		
Verbose Level	Determines whether or not a log output is created. If it is, this option specifies the minimum level of log events that are generated. Any one of the following log event levels can be selected as the value for this option (starting with the highest priority level): `standard`, `interaction`, `trace`, `debug`, `all`, or `none`. For a description of the log events levels, see Table 8 on page 68.	Select one of several log event levels. **Default value:** `standard`
Output for Level All	Specifies the outputs to which an application sends all log events. The log output types must be separated by a comma when more than one output is configured. Log events are sent to the Standard output (`stdout`).	A string of characters. **Default value:** `../logs/MRCPProxy`
Output for Level Standard	Specifies the outputs to which an application sends the log events of the Standard level. Log events are sent to the Standard output (stdout).	A string of characters. **Default value:** `../logs/MRCPProxy`
Output for Level Interaction	Specifies the outputs to which an application sends the log events of the Interaction level and higher, which means, more than one output is configured—`standard` and `interaction` levels. Log events are sent to the Standard output (stdout).	A string of characters. **Default value:** `../logs/MRCPProxy`

Table 25: Selected MRCP Proxy Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Output for Level Trace	Specifies the outputs to which an application sends the log events of the Trace level and higher, which means, more than one output is configured—`standard, interaction,` and `trace` levels. Log events are sent to the Standard output (stdout).	A string of characters. **Default value:** `../logs/MRCPProxy`
Output for Level Debug	Specifies the outputs to which an application sends the log events of the Debug level and higher, which means, more than one output is configured—`standard, interaction, trace,` and `debug` levels. Log events are sent to the Standard output (stdout).	A string of characters. **Default value:** `../logs/MRCPProxy`
Log Segmentation	Specifies whether or not there is a segmentation limit for a log file. If there is, this option sets the mode of measurement, along with the maximum size. If the current log segment exceeds the size set by this option, the file is closed and a new one is created. For a complete description of the option values for log segmentation, see Table 8 on page 68.	A string of characters. **Default value:** `10000`
Log Expiration	Determines whether or not log files expire. If they do, this option value sets the measurement for determining when they expire, along with the maximum number of files (segments) or days before the files are removed.	A string of characters. **Default value:** `20`
Keep Startup Log File	Specifies whether or not a startup segment of the log, containing the initial T-Server configuration, is kept. If it is, this option value can be set to true or to a specific size. If this option value is set to `true,` the size of the initial segment will be equal to the size of the regular log segment that is defined by the segment option. If this option value is set to `false` (segmentation is turned off), the value of this option will be ignored.	A string of characters. **Default value:** `false` Changes take effect: start/restart

Table 25: Selected MRCP Proxy Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Message File	Specifies the file name of application-specific log events. The name must be valid for the operating system on which the application is running. The option value can also contain the absolute path to the application-specific `*.lms` file. Otherwise, an application looks for the file in its working directory.	A string of characters. **Default value:** Empty
Log Messages Format	Specifies the format of log record headers that an application uses when writing logs in the log file. Using compressed log record headers improves application performance and reduces the log file's size. For a complete description of each option value, see Table 8 on page 68.	Select one of two option values—`short` or `full`. **Default value:** `short`
Time Generation for Log Messages	Specifies the system in which an application calculates the log record time when a log file is generated. The time is converted from the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970). For a complete description of each option value, see Table 8 on page 68.	Select one of two option values—`local` or `utc`. **Default value:** `local`
Time Format for Log Messages	Specifies how to represent the time when an application generates log records in a log file. For a complete description of each option value, see Table 8 on page 68.	Select one of three option values—`time`, `locale`, or `ISO8601`. **Default value:** `ISO8601`

Table 25: Selected MRCP Proxy Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Enable Printing Extended Attributes	Specifies whether the application attaches extended attributes, if any exist, to a log event that it sends to log output. Typically, log events of the Interaction log level and audit-related log events contain extended attributes. **Note:** When this option is set to `true`, audit capabilities are enabled, but performance is negatively affected. Genesys recommends that you enable this option for Solution Control Server (SCS) and Configuration Server when audit tracking is used. For other applications, see *Genesys Combined Log Events Help* to find out whether an application generates Interaction-level and audit-related log events; If it does, enable the option when testing new interaction scenarios only.	Select one of two option values—`true`, or `false`. **Default value:** `false`
Check Point Interval	Specifies how often the application generates a check point log event to divide the log into sections of equal time. By default, the application generates this log event every hour. Setting the option to 0 prevents the generation of check-point events.	An integer. **Default value:** `1`
Memory Snapshot File Name	Specifies the name of the file to which the application regularly prints a snapshot of the memory output, if configured to do so. The new snapshot overwrites the previously written data. If the application terminates abnormally, this file contains the latest log messages. **Note:** Memory output is not recommended for processors with a CPU frequency lower than 600 MHz.	A string of characters. **Default value:** Empty
Memory Output Buffer Size	Specifies the buffer size for log output to the memory, if configured. This option value can be configured in kilobytes (KB), minimum 128 KB, or megabytes (MB), maximum 64 MB.	A string of characters. **Default value:** Empty

Table 25: Selected MRCP Proxy Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Folder for Temporary Network Log Output File	Specifies the folder, including the full path, in which an application creates temporary files that are related to network log output. If the option value is changed while the application is running, the change does not affect the network output that is currently open.	A string of characters. **Default value:** Empty
Enable 6.x Compatibility Log Output Priority	Specifies whether the application uses 6.x output logic. For a complete description of each of the option values, see Table 8 on page 68.	Select one of two option values—`true` or `false`. **Default value:** `false`
`snmp` Section		
SNMP Task Timeout	Specifies the maximum amount of time, in milliseconds, that SNMP waits for a new task.	Any integer value greater than zero (`0`). **Default value:** `100`
`stack` Section		
MRCP Connection Timeout	Specifies the connection timeout, in milliseconds, for SRM MRCPv1 and MRCPv2 stack to establish a TCP connection to the server.	Any integer value. **Default value:** `10000` Changes take effect: start/restart
Enable MRCP Stack Debug Trace	Specifies whether or not to enable the `STACK DEBUG` message.	Boolean: `True/False` **Default value:** `True` Changes take effect: start/restart
RTSP Port Range for MRCPv1 Client	Specifies the port range of the RTSP stack that is used by the MRCPv1 client.	A string of characters. **Default value:** `10000-11999` Changes take effect: start/restart
`vrmproxy Section`		
FIPS Mode Enabled	Enables FIPS mode in MRCP Proxy.	• `True` • `False` **Default value:** `True` Changes take effect: start/restart

Table 25: Selected MRCP Proxy Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
MRCP Proxy Contact RTSP URI	Specifies the full Real-Time Streaming Protocol URI that is used by the MRCPv1 clients to contact this proxy. The MRCP Proxy listens for TCP connections at the port that is specified by the URI. If port is not specified in the URI, default port 11000 is assumed. If the MRCP Proxy is deployed on a host separate from the Media Control Platform, the default value must be changed to the IP of the MRCP Proxy.	String of characters in RTSP URI format. **Default value:** `rtsp://$LocalIP$:11000/mrcpproxy` Changes take effect: immediately
Error Recovery Time for Speech Resource	Specifies the timeout, in milliseconds, for a server to be put back in service after it encounters errors, such as timeouts or TCP connection errors.	Any integer value. **Default value:** `10000` Changes take effect: immediately
Barge-In Timeout	Specifies the timeout, in milliseconds, for the `BARGE-IN` to occur.	Any integer value. **Default value:** `10000` Changes take effect: immediately
Session Clean Interval	Specifies the interval, determined by `timeout.max_idle` parameter, to clean idle sessions.	Any integer value. **Default value:** `60000` Changes take effect: immediately
Close Session Timeout	Specifies the timeout, in milliseconds, of `Close-Session` requests.	Any integer value. **Default value:** `10000` Changes take effect: immediately
Control Timeout	Specifies the timeout, in milliseconds, of `CONTROL` messages.	Any integer value. **Default value:** `10000` Changes take effect: immediately
Define-Grammar Timeout	Specifies the timeout, in milliseconds, of `DEFINE-GRAMMAR` messages.	Any integer value. **Default value:** `10000` Changes take effect: immediately

Table 25: Selected MRCP Proxy Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Get-Params Timeout	Specifies the timeout, in milliseconds, of `GET-PARAMS` messages.	Any integer value. **Default value:** `10000` Changes take effect: immediately
Get-Result Timeout	Specifies the timeout, in milliseconds, of `GET-RESULT` messages.	Any integer value. **Default value:** `10000` Changes take effect: immediately
Get-Server-Info Timeout	Specifies the timeout, in milliseconds, to get a response to `Get-Server-Info` requests (ping).	Any integer value. **Default value:** `10000` Changes take effect: immediately
Session Max Idle Timeout	Specifies the maximum session idle time. Sessions that exceed this idle time are terminated.	Any integer value. **Default value:** `180000` Changes take effect: immediately
Open-Session Timeout	Specifies the timeout, in milliseconds, of open sessions.	Any integer value. **Default value:** `10000` Changes take effect: immediately
Pause Timeout	Specifies the timeout, in milliseconds, of `PAUSE` messages.	Any integer value. **Default value:** `10000` Changes take effect: immediately
Recognition-Start-Timers Timeout	Specifies the timeout, in milliseconds, of `RECOGNITION-START-TIMERS` messages.	Any integer value. **Default value:** `10000` Changes take effect: immediately
Recognize Timeout	Specifies the timeout, in milliseconds, of `RECOGNIzE` messages.	Any integer value. **Default value:** `10000` Changes take effect: immediately

Table 25: Selected MRCP Proxy Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
TCP Re-connect Interval	Specifies the interval, in milliseconds, between connection attempts, when the TCP connection with the MRCP server is not yet established.	Any integer value. **Default value:** `10000` Changes take effect: immediately
Resume Timeout	Specifies the timeout, in milliseconds, of `RESUME` messages.	Any integer value. **Default value:** `10000` Changes take effect: immediately
Set-Params Timeout	Specifies the timeout, in milliseconds, of `SET-PARAMS` messages.	Any integer value. **Default value:** `10000` Changes take effect: immediately
Speak Timeout	Specifies the timeout, in milliseconds, of `SPEAK` messages.	Any integer value. **Default value:** `10000` Changes take effect: immediately
Stop Timeout	Specifies the timeout, in milliseconds, of `STOP` messages.	Any integer value. **Default value:** `10000` Changes take immediately

Chapter

9 Configuring the Call Control Platform

This chapter provides information about how to configure the Call Control Platform and how to provision the device profiles in your Genesys Voice Platform (GVP) deployment. It contains the following sections:

Task Summary: Configuring the Call Control Platform

Task Summary: Configuring the Call Control Platform summarizes the configuration steps and options to implement Call Control Platform functionality in your GVP deployment.

Task Summary: Configuring the Call Control Platform

Objective	Related Procedures and Actions
Specify use of the `sips:` schema	Set the platform-level configuration option `mediacontroller.sipsecure` to `true`, to specify that all calls initiated by CCP via the tags `<createcall>`, `<dialogprepare>`, `<dialogstart>`, `<createconference>` and `<redirect>` are initiated in the `sips:` schema.

Task Summary: Configuring the Call Control Platform (Continued)

Objective	Related Procedures and Actions

Control SIP Secure Mode using the `hints` Attribute	Use the CCXML `hints` attribute in the tags listed above to override the platform-level configuration. Examples:
	To enable SIP Secure:
	`<var name="hints" expr="new Object()"/>`
	`<assign name="hints.sipsecure" expr="'1'"/>`
	To disable SIP Secure:
	`<var name="hints" expr="new Object()"/>`
	`<assign name="hints.sipsecure" expr="'0'"/>`
	.
	`<dialogstart src="'helloworld.vxml'"`
	`connectionid="in_connectionid" dialogid="dialogid"`
	`hints="hints"/>`

Control SIP Secure Mode Using the `dest` Attribute	Use the `dest` attribute in the `<createcall>` and `<redirect>` tags to override the corresponding `hints`. When using `dest` for this purpose requires that you also use the `sips:` schema. Examples:
	`<createcall dest="'sips:dialog@135.17.176.10:5071'"`
	`hints="hints"/>`
	`<redirect dest="'sips:dialog@135.17.176.10:5071'"`
	`hints="hints"/>`

Integrate the Call Control Platform with the Resource Manager and Media Control Platform.	Point the Call Control Platform to the Resource Manager as the SIP Proxy server and interim target of media service requests, and define the properties for SIP communications. Key configuration options are: • `mediacontroller.sipproxy` • `mediacontroller.bridge_server` (see page 222) • `sip.transport.x` (see page 45) • `sip.routeset` or `sip.secureroutest` (see page 45) For additional, relevant configuration options and actions, see "Configuring SIP Communication and Routing" on page 42 and "Enabling Secure Communication" on page 46.

Task Summary: Configuring the Call Control Platform (Continued)

Objective	Related Procedures and Actions
(Required only if you made TCP or TLS the preferred default transport protocol [see page 42]) Ensure that the Request-URI header in SIP requests specifies the required transport protocol.	Modify the CCXML applications so that the Request-URI for any endpoints includes the transport=TCP or transport=TLS parameter. • Use CCXML hints in the <createcall>, <dialogprepare>, <dialogstart>, and <createconference> tags. For example: <var name="hints" expr="new Object()"/> <assign name="hints.requesturi" expr="new Object()"/> <assign name="hints.requesturi.transport" expr="'tcp'"/> <dialogstart src="'file:///C:\Program Files\GCTI\gvp\VP Media Control Platform 8.1\MCP_80\helloaudio.vxml'" hints="hints"/> **Note:** The W3C CCXML Specification, Draft 29, specifies that the hints attribute in <createcall>, and <createconference> be an ECMAScript object. There is no such specification for <dialogprepare>, however, Genesys recommends that hints are not passed as any other primitive data type for any of these three hints attributes.
Ensure that the Call Control Platform can interact with all other SIP devices in your deployment.	Verify and, if necessary, modify the device profiles that have been provisioned. For more information, see "Configuring Device Profiles" on page 488.
Enable PRACK support.	Configure CCP for Reliable Provisional Responses, specifically: • sip.prack.support
Configure the IP DiffServ (ToS).	Set the SIP packet's ToS using [sip]transport.[n].tos See "Configuring SIP Communication and Routing" on page 42".
Configure conferencing.	See "Enabling Conference Services" on page 62.
Configure reporting.	See "Configuring Reporting" on page 63.
Configure logging.	See "Configuring Logging" on page 66.

Task Summary: Configuring the Call Control Platform (Continued)

Objective	Related Procedures and Actions
Tune Call Control Platform performance.	Configure appropriate maximums and timeouts for your deployment. Consider the following options, in particular: • `ccxmli.max_num_documents` (default is `6000`) • `ccxmli.num_session_processing_threads` (default is `5`) • `ccxmli.max_num_sessions` (default is `6000`) • `ccxmli.max_conn_per_session` (default is `100`) • `ccxmli.max_dialog_per_session` (default is `100`) • `ccxmli.max_conf_per_session` (default is `100`) See also “Configuring Session Timers and Timeouts” on page 80.
Customize client-side communication ports.	See “Configuring Client-Side Connections” on page 72.
Customize session management behavior and performance.	See “Configuring Session Timers and Timeouts” on page 80.
Customize Call Control Platform messaging.	See “Configuring SNMP” on page 72 and Table 100 on page 470.

Important Call Control Platform Configuration Options

This section describes the key configuration options that you either must or may want to customize.

Configure the options in Genesys Administrator on the `Provisioning > Environment > Applications > <Call Control Platform> > Options` tab. For the detailed steps to configure option settings, see Procedure: Viewing or modifying GVP configuration parameters, on page 34.

Except for some `ems` options, all changes to Call Control Platform options take effect immediately.

The Call Control Platform configuration options are in the following configuration sections:

- `ccpccxml`—Parameters determine the behavior of the Call Control Platform in relation to the CCXML applications (for example, whether transfers through dialogs are allowed).
- `ccxmli`—Parameters determine the behavior of the CCXML Interpreter (for example, the HTTP port and URL for the `IOProc` function; maximums for the number of sessions, documents, and per-session conferences, connections, dialogs, processing threads, and so on).

- `ems` (see Table 6 on page 64)—Parameters determine Reporting behavior for call detail records (CDRs) and metrics.
- `fm`—Parameters determine file fetching behavior.
- `log` (see "Service Quality Analysis (SQA)" on page 65)—Parameters determine behavior for Management Framework logging.
- `mediacontroller`—Parameters integrate the Call Control Platform, through the Resource Manager, with the Media Control Platform, which acts as a bridge server for call transfers and conferences.
- `session`—Parameters determine the behavior of the Call Control Platform during sessions (for example, whether unknown headers will be copied into forwarded SIP messages).
- `sip`—Parameters integrate the Call Control Platform with the SIP Proxy (the Resource Manager).
- `snmp` (see "Configuring SNMP" on page 72)—Parameters determine SNMP behavior.

Table 26 provides information about important Call Control Platform parameters that are not described in Chapter 3 on page 41. Table 26 provides parameter descriptions as well as the default parameter values that are preconfigured in the Call Control Platform `Application` object.

For information about all the available configuration options for the Call Control Platform, see the *Genesys Voice Platform 8.1 Configuration Options Reference.*

Table 26: Selected Call Control Platform Configuration Options

Option Name	Description	Valid Values and Syntax
	`ccpccxml` Section	

FIPS Mode Enabled	Enables FIPS mode in CCP.	`True`
		`False`
		Default value: `False`
		Changes take effect: start/restart

Default CCXML	The URI for the default CCXML application.	`<URI path to file>`
		Default value:
		`file://$InstallationRoot$\config\default.ccxml`

Table 26: Selected Call Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Send SIP Progressing	Specifies whether CCP is to send the 180 SIP response with the `<accept>` tag for all incoming calls.	• `0`—The 180 response is sent when the `<send>` tag is called. • `1`—The 180 response is sent immediately after sending the `100 Trying` message. **Default value:** `0`
	`ccxmli` Section	
BasicHTTP Receive - Show Error Body	Specifies whether or not a descriptive text will be returned in the response body when an HTTP failure response is given for a request to the Basic HTTP Event I/O Processor. If this option is set to `true`, it is enabled and a descriptive text will be returned.	• `true` • `false` **Default value:** `false`
BasicHTTP Receive - Host for IPv4 network	Specifies the IPv4 address or host name on which the basic HTTP event I/O processor will listen for HTTP requests on IPv4 network interface. If the value of this option is an empty string, the system listens on all available IPv4 network interfaces. If the host name is specified, the first IPv4 address in the resolved list is used.	String **Default value:** Empty
BasicHTTP Receive - Host for IPv6 network	Specifies the IPv6 address or host name on which the basic HTTP event I/O processor will listen for HTTP requests on IPv6 network interface. If the value of this option is an empty string, the system listens on all available IPv6 network interfaces. If the host name is specified, the first IPv6 address in the resolved list is used.	String **Default value:** Empty
CreateSession Receive Host for IPv4 network	The IPv4 address or host name on which the session creation event I/O processor will listen for HTTP requests on IPv4 network interface. If the value of this option is an empty string, the system listens on all available IPv4 network interface. If the host name is specified, the first IPv4 address in the resolved list is used.	String **Default value:** Empty

Table 26: Selected Call Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
CreateSession Receive Host for IPv6 network	Specifies the IPv6 address or host name on which the session creation event I/O processor will listen for HTTP requests on IPv6 network interface. If the value of this option is an empty string, the system listens on all available IPv6 network interface. If the host name is specified, the first IPv6 address in the resolved list is used.	String **Default value:** Empty
Preferred IP Version Used in BasicHTTP Access URI	Specifies the preferred IP version that will be used in basic HTTP access URI `session.ioprocessors["basichttp"]`.	• `ipv4` • `ipv6` **Default value:** `ipv4`
Preferred IP version Used in CreatesSession Access URI	Specifies the preferred IP version that will be used in the create session access URI `session.ioprocessors["createsession"]`.	• `ipv4` • `ipv6` **Default value:** `ipv4`
Save CCXML Files Save Script Files	Specifies whether fetch request, response, and data for each CCXML or ECMAScript file that is fetched and processed in a session will be saved to disk. This feature is convenient for debugging CCXML applications, particularly when CCXML pages are dynamically generated during a session.	• `TRUE` • `FALSE` **Default value:** `FALSE`
	`fm` Section	
HTTP Port Range	Specifies the local port range that will be used for HTTP requests. If this parameter is not specified, the CCP allows the operating system choose the local port.	String **Default value:** Empty
HTTP Proxy	Specifies the HTTP proxy to use for HTTP requests.	`<host:port>` **Default value:** `localhost:3128`
HTTPS Proxy	Specifies the HTTPS proxy to use for HTTPS requests.	`<host:port>` **Default value:** Empty

Table 26: Selected Call Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Outgoing Interface	Specifies the network interface IP address that is used for outgoing HTTP requests. If this configuration option has an empty value, the Media Control Platform automatically selects the network interface it will use. If the Squid HTTP proxy is used, it must be configured to accept HTTP requests from the interface that is specified. Otherwise, by default, it accepts HTTP requests from the local host only.	Any string of characters. **Default value:** Empty
No Cache URL Substring	Specifies that documents fetched from a URL containing one of the substrings in this list should not be cached. Any substring listed in this comma delimited list, *will not* be cached.	Any comma delimited list of characters. **Default value:** `cgi-bin,jsp,?`
Maxage for Local File	Specifies, in seconds, how long to cache local file for. If set to `0`, local files will not be cached.	Any integer. **Default value:** `60`
Maximum Cache Size	The total maximum size, in bytes, of all cached files.	Any integer. **Default value:** `50,000,000`
Maximum Cache Entry Size	Specifies the maximum size, in bytes, of each cache entry.	Any integer. **Default value:** `500,000`
Maximum Cache Entry Count	Specifies the maximum number of entries that can be stored in cache.	Any integer. **Default value:** `1000`
Maximum Redirections	Specifies the maximum number of times to follow the `Location` header in the HTTP response. If set to `0`, HTTP redirection is disabled.	Any integer. **Default value:** `5`
Enable Cookie	Specifies whether to enable HTTP cookie.	• `0`—Disable • `1`—Enable **Default value:** `1`
Enable 100-Continue header	Specifies whether to enable the Expect: 100-continue header in HTTP 1.1 POST requests.	• `0`—Disable • `1`—Enable **Default value:** `0`
SSL Certificate	Specifies the certificate file name.	Any string of characters. **Default value:** Empty

Table 26: Selected Call Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
SSL Certificate Type	Specifies the format of the certificate file name.	• `PEM`—Privacy Enhanced Mail • `DER`—Distinguished Encoding Rules **Default value:** `PEM`
SSL Key	Specifies the private key file name.	Any string of characters. **Default value:** Empty
SSL Key Type	Specifies the format of the key file name.	• `PEM`—Privacy Enhanced Mail • `DER`—Distinguished Encoding Rules **Default value:** `PEM`
SSL Key Password	Specifies the password required in order to use the SSL Key.	Any string of characters. **Default value:** Empty
SSL Version	Specifies the Secure Socket Layer version to use.	• `0`—Automatically detect version • `1`—Force TLSv1 • `2`—Force TLSv2 • `3`—Force TLSv3 **Default value:** `0`
Verify Peer Certificate	Specifies whether to verify the peer's certificate. **Note:** SSL CA Info or SSL CA Path must also be set in order for this parameter to take affect.	• `0`—Do not verify • `1`—Verify **Default value:** `0`
SSL CA Path	Specifies the path to the directory holding the peer certificates. **Note:** This directory must be created using the `openssl c_rehash` utility.	Any string of characters. **Default value:** Empty
SSL Random File Seed	Specifies the random initial value used to generate the first number of the SSL key.	Any string of characters. **Default value:** Empty

Table 26: Selected Call Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
SSL Verify Host	Specifies how the common name from the peer certificate is to be verified during the SSL handshake.	• 0—Do not verify • 1—Check existence only • 2—Make sure that it matches provided host name **Default value:** 0
SSL Cipher List	Specifies the list of ciphers to use for the SSL Connection.	Any string of characters. **Default value:** Empty
mediacontroller Section		
Address of bridge server	The Resource Manager IP address. The Call Control Platform sends requests to the Resource Manager to find a bridging server to use when two endpoints cannot be joined because of media bridging limitations (implicit conference and transcoding). The bridge server must be capable of: • Sending media to multiple endpoints. • Sending and receiving from distinct endpoints. • Performing transcoding.	<IP address> **Default value:** Empty
Device Profile of Bridge Server	The name of the device profile to use with the configured bridge server. For information about configuring device profiles, see “Configuring Device Profiles” on page 488.	<Device profile name> **Default value:** Default Conference

Table 26: Selected Call Control Platform Configuration Options (Continued)

<table>
<tr><th>Option Name</th><th>Description</th><th>Valid Values and Syntax</th></tr>
<tr><td>Full Audio Codec
Full Video Codec</td><td>A space-separated list of the <code><media type></code> codecs that get set in the SDP in an initial offer when there is no media bridge. In other words, the media line that will be used to create a connection less SDP.</td><td><code><payload>|<codec>|<MIME-type>|<rate>|<number of channels></code>
Default values:
• Audio—
<code>0|pcmu|audio/basic|8000|1</code>
<code>8|pcma|audio/x-alaw-basic|8000|1</code>
<code>9|g722|audio/g722|8000|1</code>
<code>23|g726-16|audio/g726-16|8000|1</code>
<code>22|g726-16|audio/g726-16|8000|1</code>
<code>2|g726-32|audio/g726|8000|1</code>
<code>125|g726-40|audio/g726-40|8000|1</code>
<code>18|g729|audio/g729|8000|1</code>
<code>3|gsm|audio/x-gsm|8000|1</code>
• <code>|1105|AMR|audio/AMR|8000|1</code></td></tr>
<tr><td>Full Audio Codec
Full Video Codec (continued)</td><td>A space-separated list of the <code><media type></code> codecs that get set in the SDP in an initial offer when there is no media bridge. In other words, the media line that will be used to create a connection less SDP. (continued)</td><td><code>112|AMR-WB|audio/AMR-WB|16000|1</code>
<code>101|telephone-event|none|8000|1</code>
• Video—
<code>34|h263|video/H263|90000|1</code>
<code>99|h263-1998|video/H263-1998|90000|1</code>
<code>113|H264|video/H264|90000|1</code></td></tr>
<tr><td>Inbound allowed Media</td><td>The default allowed media types for an inbound call. All inbound calls will be limited to this set of media types in terms of SDP exchange.
Note: If set to <code>dynamic</code>, the media type is determined from the capability SDP of the inbound call. If capability SDP is not available, it defaults to <code>audio</code> and <code>video</code>.</td><td>• <code>dynamic</code>
• <code>audio video</code>
• <code>audio</code>
• <code>video</code>
Default value: <code>dynamic</code></td></tr>
</table>

Table 26: Selected Call Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
SDP localhost	Specifies the `host` part of the local host IPv4 address that is used in SDP.	String **Default value:** `$LocalIP$`
SDP localhost IPv6	Specifies the `host` part of the local host IPv6 address that is used in SDP.	String **Default value:** `$LocalIPv6$`
SIP Proxy	The Resource Manager address-of-record (AOR). The Resource Manager is the SIP Proxy that the Call Control Platform uses for outbound SIP requests.	`<Resource Manager IP address>:<SIP port>` **Default value:** Empty
Unknown headers allowed for a SIP Message	A space-separated list of the unknown headers that can be sent in an outgoing SIP message. Genesys recommends that you allow the following headers: • `Reason` • `Warning` **Note:** Specifying a wildcard (*) means that all unknown headers are allowed; therefore, the wildcard should be the only value in the field—for example, `sip.allowknownheaders` = `*`.	A string specifying a list header names. **Default value:** `Warning Reason`
`sip` Section		
Enable Reliable Provisional Responses	Specifies whether to allow the SIP stack to send reliable provisional responses (100-199). If set to `1`, PRACK is supported, and the `100rel` extension is included in the `Supported` header of the outbound `INVITE` request. If set to `2`, PRACK is required, and the `100rel` extension is included in the `Require` header of the outbound `INVITE` request.	• `0`—Disable • `1`—Supported • `2`—Required **Default value:** `0`
Default IPv4 route for UDP	Specifies the default IPv4 route for UDP. The number denotes the transport that is defined in the `sip.transport.x` configuration option, where `x` is the value of this parameter and is used when no IPv4 UDP routes are found.	Numeric **Default value:** Empty

Table 26: Selected Call Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Default IPv6 route for UDP	Specifies the default IPv6 route for UDP. The number denotes the transport that is defined in the `sip.transport.x` configuration option, where `x` is the value of this parameter and is used when no IPv6 UDP routes are found. If this parameter is not set, the first IPv6 UDP transport found in the `sip.transport.x` becomes the default.	Numeric **Default value:** Empty
Default IPv4 route for TCP	Specifies the default IPv4 route for TCP. The number denotes the transport that is defined in the `sip.transport.x` configuration option, where `x` is the value of this parameter and is used when no IPv4 TCP routes are found.	Numeric **Default value:** Empty
Default IPv6 route for TCP	Specifies the default IPv6 route for TCP. The number denotes the transport that is defined in the `sip.transport.x` configuration option, where `x` is the value of this parameter and is used when no IPv6 TCP routes are found. If this parameter is not set, the first IPv6 TCP transport found in the `sip.transport.x` becomes the default.	Numeric **Default value:** Empty
Default IPv4 route for TLS	Specifies the default IPv4 route for TLS. The number denotes the transport that is defined in the `sip.transport.x` configuration option, where `x` is the value of this parameter and is used when no IPv4 TLS routes are found.	Numeric **Default value:** Empty
Default IPv6 route for TLS	Specifies the default IPv6 route for TLS. The number denotes the transport that is defined in the `sip.transport.x` configuration option, where `x` is the value of this parameter and is used when no IPv6 TLS routes are found. If this parameter is not set, the first IPv6 TLS transport found in the `sip.transport.x` becomes the default.	Numeric **Default value:** Empty

Table 26: Selected Call Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Local Transport IPv4 Address	Specified the `sent-by` field of the `Via` header and the `hostport` part of the `Contact` header in the outgoing SIP message will be set to this value if a IPv4 transport is used. The value must be a host name or domain name. If this option value is left empty the outgoing transport's actual IP and port is used for the `Via` and `Contact` headers. **Note:** If the domain name that is used in the `SRV` record query is specified, the `sip.transport.localaddress.srv` configuration option must be set to `true` to prevent the `port` part from being automatically generated by the SIP stack.	String **Default value:** Empty
Local Transport IPv6 Address	Specifies that the `sent-by` field of the `Via` header and the `hostport` part of the `Contact` header in the outgoing SIP message will be set to this value if a IPv6 transport is used. The value must be a host name or domain name. If this option value is left empty the outgoing transport's actual IP and port is used for the `Via` and `Contact` headers. **Note:** If the domain name that is used in the `SRV` record query is specified, the `sip.transport.localaddress.srv` configuration option must be set to `true` to prevent the `port` part from being automatically generated by the SIP stack.	String **Default value:** Empty
Local Transport Address Contains SRV Domain Name	Specifies whether or not the `sip.transport.localaddress` configuration option contains an SRV domain name. • If this option value is set to `true,` the port part is not automatically generated by the SIP stack. • If this option value is set to `false,` the outgoing transport's port is used, together with the host name that is specified by the `sip.transport.localaddress` configuration option.	• `TRUE` • `FALSE` **Default value:** `FALSE`

Table 26: Selected Call Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Preferred IP version Used in SIP	Specifies the local TCP port range to be used for SIP transport. If this parameter is not specified, the CCP allows the operating system to choose the local port.	String **Default value:** Empty
Route Set	Specifies the route set for non-secure SIP outbound calls. If defined, this route set is inserted as the ROUTE header for all outgoing calls and forces the MCP to send the SIP messages through this defined route set. Each element in the routeset must be separated by commas. For example, sip.routeset=<sip:p1.example.com;lr>,<sip:p2.domain.com;lr> **Note:** This parameter does not apply to SIP REGISTER messages.	Any string of characters. **Default value:** Empty
SIP Static Route List	Specifies, in a pipe delimited list, the static route groups. Each route group contains a list, separated by commas, of IP addresses. Within the route group, each IP address may substitute each other as an alternate route destination if sending a SIP request to one of the IP address that fails. For example, 10.0.0.1,10.0.0.2|10.0.10.1,10.0.10.2 specifies two static route groups, and each group specified two routes that are alternate to each other.	Any string of characters. **Default value:** Empty

Table 26: Selected Call Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
IP Type of Service for RTP/RTCP	Specifies the IP differentiated services field (ToS) to set in all outgoing RTP/RTCP packets. **Notes:** • For Windows Server 2003, the ToS must be enabled in the registry. See `http://support.microsoft.com/kb/248611` • For Windows Server 2008, the ToS configuration is not supported. It must be configured at the OS level. You can define per executable and per port, and what type of `DiffServ` bits to set on the outgoing packets using the QoS policy defined in the following article. `http://technet.microsoft.com/en-us/library/cc771283.aspx` • For all Operating Systems, when the SIP/RTP packets are sent across different subnets, the router may reset the DiffServ bits in the IP header even though it was set by MCP.	Range: `0-255` Examples: • `0`—Disabled • `16`—IPTOS LOWDELAY (0x10) • `32`—IPTOS PREC PRIORITY (0x20) • `64`—IPTOS PREC CRITICAL (0x40) • `184`—DiffServ EF (Expedited Forward 0xB8) **Default value:** `0`
For other SIP section parameters, see Table 4 on page 44.		

Chapter

10 Configuring the CTI Connector

The Genesys Voice Platform (GVP) provides two modes of CTI deployment that provide access to Genesys Management Framework functionality—Genesys CTI through IVR Server and Cisco CTI through Intelligent Contact Management (ICM).

This chapter provides information about how to configure the CTI Connector to function in each of the two deployment modes. It contains the following sections:

Configuring the CTI Connector

When you install the CTI Connector, you can select the CTI Framework (Genesys or Cisco) that is appropriate for your environment. This section includes task summaries for each type of CTI deployment

Task Summary: Configuring the CTI Connector for Genesys CTI summarizes the configuration steps and options to implement CTI Connector functionality in your Genesys CTI deployment.

Task Summary: Configuring the CTI Connector for Genesys CTI

Objective	Related Procedures and Actions
Integrate the CTI Connector with the IVR Server.	Point the CTI Connector to the IVR Server. The key configuration section is `IVRServer_Sample` (see Page 233). For more information on integrating GVP with IVR Server, see the *Voice Platform Solution 8.1 Integration Guide*.
Enable CTI Transfers.	See "Provisioning IVR Profiles for GVP" on page 107.
Ensure that the CTI Connector can interact with all other SIP devices in your deployment.	Verify and, if necessary, modify the device profiles that have been provisioned. For more information, see "Configuring Device Profiles" on page 488. **Note:** The error response that is forwarded by SIP Server 8.1 to GVP from the agent by must always be the same (`603 Decline`). To ensure this happens, set the `sip-busy-type` configuration option value to `2` on the Trunk to which the error response will be sent.
Configure reporting.	See "Configuring Reporting" on page 63.
Configure logging.	See "Configuring Logging" on page 66.
Customize client side communication ports.	See "Configuring Client-Side Connections" on page 72.

Task Summary: Configuring the CTI Connector for Cisco CTI summarizes the configuration steps and options to implement CTI Connector functionality in your Cisco CTI deployment.

Task Summary: Configuring the CTI Connector for Cisco CTI

Objective	Related Procedures and Actions
Integrate the CTI Connector with Cisco ICM.	When you are installing the CTI Connector, and you select the Cisco ICM, the Service Control Interface is initialized by default. 1. If you want to use the Call Routing Interface (CRI), in the CTI Connector `Application`, configure the `[icmc].ICMInterface` option with the `CRI` value.
	2. Configure the ICM Trunk Group ID: In the CTI Connector `Application`, configure the `[icmc].TrunkGroupID` option with the applicable value.

Task Summary: Configuring the CTI Connector for Cisco CTI (Continued)

Objective	Related Procedures and Actions
Configure the listener ports for the VRU-PGs	**Single Tenant Environments** You can configure the `TenantName` configuration option for a specific tenant, to enable the CTI Connector to handle calls for that tenant only, and reject inbound calls from all other tenants. (See Specify the Tenant name. in this table.) The CTI Connector supports multiple VRU-PG connections for a single tenant, and you can specify a comma-separated list of listener ports, one for each VRU-PG. • In the CTI Connector Application's Tenant1 configuration section, change the value of the Ports configuration option, as required. A TrunkGroupID (TG ID) is a list of listener port numbers, separated by commas, on which CTIConnector waits for a TCP connection from the Cisco VRU-PG. Optionally, the TG IDs supported by the PIMs can also be configured here. The TG IDs can be listed for a particular PIM separated by an ampersand (&). For example: `6000:1&2,7000,8000:3&4` In this example, `6000` supports Trunk Group IDs `1` and `2`, `7000` does not specify the TG IDs it supports and 8000 supports TG IDs 3 and 4. **Notes:** • Valid range for Trunk Group IDs is `0-65535`. • Same TG IDs should not be mentioned by more than one PIM; Trunk Group IDs must be unique across all the PIMs. • The value mentioned as the default TG ID under the ICMC section should not be specified by any of the PIMs as a supported Trunk Group.

Task Summary: Configuring the CTI Connector for Cisco CTI (Continued)

<table>
<tr><th>Objective</th><th>Related Procedures and Actions</th></tr>
<tr><td>Configure the listener ports for the VRU-PGs (continued)</td><td>Multi-tenant Environments

The CTI Connector supports multi-tenant configurations and multiple VRU-PG connections for each tenant. For multi-tenant environments:
Copy the <code>Tenant1</code> section and rename it for each additional tenant. For example, <code>Tenant2, Tenant3, Tenant4</code>.
Ensure each section has a <code>TenantName</code> option configured with a valid Tenant's name in Configuration Server. If there is no value in the <code>TenantName</code> field, the CTI Connector processes requests from all tenants that do not have a <code>TenantName</code> configured.

For example, if there are three tenants and Tenant 1 is configured as follows:
<code>[T1]</code>
<code>TenantName=T1</code>
but Tenant 2 does not have a <code>TenantName</code> configured, it will cater to requests received for Tenant 2 and Tenant 3.
For each newly created tenant, in the CTI Connector <code>Application's Tenantx</code> configuration section, change the value of the <code>Ports</code> configuration option, as required. For example, <code>8000, 9000, 10000</code>.
TrunkGroupID is set to the value of the parameter <code>gvp.rm.resource-req.TrunkGroupID</code> received as a RURI param in the incoming sip INVITE message. If the parameter is not received, then you can set the value here. Default is <code>1</code>.
Note: Ensure that there are no duplicate ports configured across all tenants.</td></tr>
<tr><td>Specify the Tenant name.</td><td>In the CTI Connector <code>Application's Tenant1</code> section, enter the tenant name for value of the <code>TenantName</code> configuration option.
For each newly created tenant, in the CTI Connector Application’s Tenantx configuration section, change the value of the Ports configuration option, as required. For example, <code>8000, 9000, 10000</code>.</td></tr>
</table>

Task Summary: Configuring the CTI Connector for Cisco CTI (Continued)

Objective	Related Procedures and Actions
Ensure that the CTI Connector can interact with all other SIP devices in your deployment.	Verify and, if necessary, modify the device profiles that have been provisioned. For more information, see "Configuring Device Profiles" on page 488. **Note:** The error response that is forwarded by SIP Server 8.1 to GVP from the agent by must always be the same (`603 Decline`). To ensure this happens, set the `sip-busy-type` configuration option value to `2` on the Trunk to which the error response will be sent.
Configure reporting.	See "Configuring Reporting" on page 63.
Configure logging.	See "Configuring Logging" on page 66.
Customize client side communication ports.	See "Configuring Client-Side Connections" on page 72.

Important CTI Connector Configuration Options

This section describes the key configuration options that you either must or may want to customize.

Configure the options on Genesys Administrator on the `Provisioning > Environment > Applications > <CTI Connector> > Options` tab. For the detailed steps to configure option settings, see Procedure: Viewing or modifying GVP configuration parameters, on page 34.

The configurable CTI Connector parameters are in the following configuration sections:

- `ctic`—Parameters determine CTI Connector behavior.
- icmc—Parameters that enable CTI Connector/Intelligent Contact Management (ICM) functionality.
- `IServer_Sample`—Parameters determine IVR Server information and properties in the deployment.
- `ivrsc`—Parameters required for controlling CTI Connector integration with IVR Server.
- `ems` (see Table 6 on page 64)—Parameters determine Reporting behavior for call detail records (CDRs) and metrics.
- `log` (see "Service Quality Analysis (SQA)" on page 65)—Parameters determine behavior for Management Framework logging.

- mediacontroller—Parameters required for controlling the B2BUA framework that CTI Connector uses.
- sip—Parameters required for defining the SIP protocol level attributes for the Media Controller embedded SIP Stack.

Table 27 provides information about important CTI Connector parameters that are not described in Chapter 3 on page 41. Table 27 provides parameter descriptions as well as the default parameter values that are preconfigured in the CTI Connector Application object.

Unless indicated otherwise, all changes take effect on restart.

For information about all the available configuration options for the Media Control Platform, see the *Genesys Voice Platform 8.1 Configuration Options Reference.*

Table 27: Selected CTI Connector Configuration Options

Option Name	Description	Valid Values and Syntax
	ctic Section	
Default DNIS	Specifies the default DNIS if the IVR Server does not provide the DNIS. **Note:** This option is applicable only when IVR Server is configured in behind-the-switch-mode	Any string of characters. **Default value:** Empty
Disable CTI Connector's CDR Update	Specifies whether to collect CDRs for CTI Connector.	• True • False **Default value:** False
Fetch DNIS From IVR Server	Specifies whether the CTI Connector it to receive the DNIS from IVR Server. **Note:** This option is applicable only when IVR Server is configured in behind-the-switch-mode.	• True • False **Default value:** False
FIPS Mode Enabled	Enables FIPS mode in CTIC.	True False Default value: False Changes take effect: start/restart

Table 27: Selected CTI Connector Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
IVRPort Base Index	Specifies the starting IVR port number. Each port number increments by one after this is set. If this parameter is set to `-1`, CTI Connector will not generate an IVR Port, instead it will take the port base on `DNISIndicator`. **Notes:** Use this option when the port information is unavailable. IVR Server uses this port number to pass the DNIS information. For deployments using multiple CTI Connectors, the port range must be distinct. This parameter is applicable when IVR Server is deployed in-front-of-switch mode.	Any integer value. **Default value:** `-1`
Max IVRPorts	Specifies the maximum number of IVR ports that CTI Connector uses. **Notes:** Use this option when the port information is unavailable. This parameter is applicable when IVR Server is deployed in-front-of-switch mode.	Any integer value. **Default value:** `2000`
CTI Framework	Specifies which CTI framework to use for CTI functionality.	• `IVRServerClient`—The Genesys IVR Server • `CiscoICMClient`—The Cisco ICM **Default value:** `IVRServerClient`
`icmc` Section		
Trunk Group ID	The Trunk Group ID information that is sent to the ICM for every call through the Voice Resource Unit-Peripheral Gateway (VRU-PG) to report ICM metrics.	Any integer value. **Default value:** `0`
ECC Variables	CTI Connector registers the configured list of ECC variable names with ICM through the initial `REGISTER_VARIABLES` message. The ECC variable names must be separated by commas.	A string of characters. **Default value:** Empty

Table 27: Selected CTI Connector Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
ECC SessionID Variable Name	CTI Connector sends the SessionID to ICM through this variable. Example: `userECCVar1` The variable name must be specified in the ECC Variables list. If not, the SessionID will not be sent in the `NEW_CALL` message. By default, the variable is set to `userSessionId` and the SessionID is sent through `userSessionId`. If it is empty, the SessionID is not sent in the `NEW_CALL` message.	`eccSessionIdVarName` **Default value:** `userSessionId` Takes effect at start/restart.
Use Translation Label	This value indicates to CTIC whether the incoming call is translation-routed or normal. Set to `true` for Type 8 Network VRU deployment.	`translation-routed-call` **Default value:** `false` Takes effect at start/restart.
DNIS mapping attribute from `RUN_SCRIPT_REQ` message	This parameter value indicates which field from RUN_SCRIPT_REQ message should be used for fetching the DNIS value.	`DNISIndicator` **Valid values:** *ScriptID*, `1-10` **Default value:** None (blank) Takes effect after restart
ICM Interface to Use	Specifies the interface that is used by the CTI Connector to communicate with ICM. By default, it communicates with ICM by using the Service Control Interface (SCI).	• Service Control Interface—`0` • Call Routing Interface (CRI)/Event Data Feed (EDF)—`1` **Default value:** `0`
	`ivrsc` Section	
Customer IVR Servers List	Specifies the list of IVR Servers that CTI Connector uses.	A string of characters. **Default value:** `IServer_Sample;`
Fetch Script ID from URS	Specifies the user defined key value from Genesys Framework.	Any integer value. **Default value:** `0`
Script ID Key Name	Specifies the key name configured in URS that is used in the UdataGet message for IVR Server Client. **Note:** This parameter is applicable when IVR Server is set in behind-the-switch mode.	A string of characters. **Default value:** Empty

Table 27: Selected CTI Connector Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
	`IServer_Sample` **Section**	
IVR Client Name	Specifies the IVR Group Name that is configured in Genesys Administrator.	A string of characters. **Default value:** Empty
IVR Server Host IP Address	Specifies the host name of the IVR Server.	`<Host name or IP address>` **Default value:** Empty
IVR Server Communication Port	Specifies the `gli_server_address` port of the IVR Server application as configured in Genesys Administrator.	Any integer. **Default value:** Empty
	`mediacontroller` **Section**	
Default IP version in SDP	Specifies the default IP version that will be used in the SDP message, and applied to the initiated SDP offer to the endpoint.	• `ipv4` • `ipv6` **Default value:** `ipv4`
Local IPv6 Address for SDP	Specifies whether or not the `sent-by` field of the `Via` header and the `hostport` part of the `Contact` header in the outgoing SIP message is set to this value if a IPv6 transport is used. The value must be a host or domain name. If this option value is left empty the outgoing transport's actual IP and port is used for the `Via` and `Contact` headers. If the domain name that is used in the `SRV` record query is specified, the `sip.transport.localaddress.srv` option must be set to `true` to prevent the `port` part from being automatically generated by the SIP stack.	String. **Default value:** Empty
SIP Proxy	Specifies The address of SIP Proxy for outbound SIP requests, in the following format: `10.10.30.205:5070`	String. **Default value:** `$LocalIP$:5080`
	`sip` **Section**	
Contact Header User Name	Specifies the Contact Header name generated by the platform.	A string of characters. **Default value:** `CTIConnector`

Table 27: Selected CTI Connector Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
SIP Static Route List	Specifies, in a pipe delimited list, the static route groups. Each route group contains a list, separated by commas, of IP addresses. Within the route group, each IP address may substitute each other as an alternate route destination if sending a SIP request to one of the IP address that fails. For example, `10.0.0.1,10.0.0.2|10.0.10.1,10.0.10.2` specifies two static route groups, and each group specified two routes that are alternate to each other.	Any string of characters. **Default value:** Empty
`Tenant1` Section		
Ports		
Peripheral Gateway Communication Port Numbers	• Specifies a list of listener port numbers, separated by commas on which the CTI Connector waits for TCP connections from the Cisco VRU-PG. For example: `6000,7000,8000`. • `TrunkGroupID` is set to the value of the parameter `gvp.rm.resource-req.TrunkGroupID` received as a RURI parameter in the incoming sip INVITE message. If the parameter is not received, then you can set the value here. Default is `1`.	Any string of characters. **Default value:** `9000`
Tenant Name	Specifies the name of the tenant.	Any string of characters. **Default value:** Empty

Load Balancing Between IVR Servers

This feature enables the configuration between two or more IVR Servers to support the same set of IVR Server Clients. The IVR Server Client uses a round-robin methodology for distributing calls between IVR Servers. Load balancing is supported for both In-Front-of-the-Switch and Behind-the-Switch-modes.

Cisco ICM Messages and Data Formats

ICM's Voice Resource Unit (VRU) uses the GED-125 interface, which is divided into two sections—the *communications* interface and *applications* interface. The communications interface defines conventions and protocols necessary to establish, maintain, and terminate data communications between the ICM and the GVP. The applications interface defines messages that allow GVP to exchange call processing information with the ICM.

Table 28 lists the ICM messages that are supported by the CTI Connector for each interface and how they are used.

Table 28: CTI Connector-Supported ICM Messages

ICM messages			
Connection Management			
OPEN_REQ	GVP	<-	ICM
OPEN_CONF	GVP	->	ICM
CLOSE_REQ	GVP	<-	ICM
CLOSE_CONF	GVP	->	ICM
HEARTBEAT_REQ	GVP	<-	ICM
HEARTBEAT_CONF	GVP	->	ICM
FAILURE_CONF	GVP	->	ICM
FAILURE_EVENT	GVP	->	ICM
SCI Initialization sequence			
INIT_SERVICE_CTRL_REQ	GVP	<-	ICM
INIT_SERVICE_CTRL_CONF	GVP	->	ICM
INIT_SERVICE_CTRL_DATA	GVP	->	ICM
INIT_SERVICE_CTRL_END	GVP	->	ICM
REGISTER_VARIABLES	GVP	->	ICM
EDF Initialization sequence			
INIT_DATA_REQ	GVP	<-	ICM
INIT_DATA_CONF	GVP	->	ICM
FAILURE_CONF	GVP	->	ICM

Table 28: CTI Connector-Supported ICM Messages (Continued)

ICM messages			
INIT_VRU_DATA_EVENT	GVP	->	ICM
INIT_DATA_END_EVENT	GVP	->	ICM
SCI messages			
NEW_CALL	GVP	->	ICM
RUN_SCRIPT_REQ	GVP	<-	ICM
RUN_SCRIPT_RESULT	GVP	->	ICM
CONNECT	GVP	<-	ICM
RELEASE	GVP	<-	ICM
CANCEL	GVP	<-	ICM
DIALOGUE_FAILURE_CONF	GVP	->	ICM
DIALOGUE_FAILURE_EVENT	GVP	-> or <-	ICM
EVENT_REPORT	GVP	->	ICM
CONNECT_TO_RESOURCE	GVP	<-	ICM
RESOURCE_CONNECTED	GVP	->	ICM
CRI messages			
ROUTE_REQUEST_EVENT	GVP	->	ICM
ROUTE_SELECT	GVP	<-	ICM
ROUTE_END_EVENT	GVP	->	ICM
ROUTE_END	GVP	<-	ICM
EDF events			
DELIVERED_EVENT	GVP	->	ICM
CALL_CLEARED_EVENT	GVP	->	ICM
Mid-call events using EDF			
SET_CALL_VARIABLES	GVP	->	ICM

Interaction Data Formats

ICM can send or receive call-related data to or from the CTI Connector in ICM messages, that is used to make routing decisions. For example, ICM can determine which call control instruction to send to CTI Connector, based on Caller Entered Digits (CED) information that GVP passes to ICM. Similarly, ICM can send call data that VoiceXML applications use to play specific instructions to the caller.

Three kinds of constructs are used to exchange this call data with the IVR: CED, Call Variables and ECC variables. The data and variables information is passed within GVP in the SIP messages that are exchanged between the CTI Connector and the Media Control Platform. Depending on the call flow, the following SIP messages are used to exchange this information:

- `INVITE`, `REFER`—Interaction data is passed in the custom SIP header in the following format:

  `X-Genesys-ICM_CED: 109`

  `X-Genesys-ICM_CallVar1=<xx>`

  `X-Genesys-ICM_CallVar6=<yy>`

  `X-Genesys-ICM_ECC_user<variablename>=<value>`

- `INFO`, `BYE`—Interaction data in passed in the SIP message body. The content type `application/x-www-form-urlencoded;charset=utf-8` is passed in the following format:

  `ICM_CED=1&ICM_CallVar1=<xx>&ICM_CallVar7=<yy>&ICM_ECC_user<variable name>=<value>`

Table 29 contains a description of the format and the method that is used to exchange interaction data between the Media Control Platform and the CTI Connector.

Table 29: ICM Constructs Format in SIP Messages

ICM Construct	Naming format in SIP messages	Description
Caller Entered Digits	`ICM_CED`	ICM identifies the caller entered digit field by a tag identifier and not by name. The name is used between the VoiceXML application, Media Control Platform, and CTI Connector only to make it user friendly and readable.

Table 29: ICM Constructs Format in SIP Messages (Continued)

ICM Construct	Naming format in SIP messages	Description
Call variables	ICM_CallVar<N>	ICM refers to call variables by their numbered tags and not by the call variable name. The name is used between VoiceXML application, the Media Control Platform, and CTI Connector only to make call variables more user friendly and readable.
ECC variables	ICM_ECC_user<variable name>	ECC variable names must begin with user. To avoid name clash with other ECC variables that are currently in use, as a best practice, Genesys recommends that you add the company name to the ECC variable name. For example, user<variable name> The CTI Connector submits the user<variable name> part only to ICM as the ECC variable name.

CTIC (Genesys) and Treatments

This section describes how CTI Connector (Genesys) handles various treatments and treatment types. For brevity, CTI Connector (Genesys) is abbreviated to CTIC(G).

Invalid Treatment Types

CTIC(G) sends `TreatStatus` as `NotStarted` to IServer if:

- CTIC(G) receives an invalid treatment type.
- CTIC(G) receives a `PlayAnnounce` treatment request containing `PROMPT`, for the following cases:
    - if the `PROMPT` option `DigitsNumber` is present in the `TreatCall` request.
    - if the `PROMPT` option `iUser_Ann_ID` is present in the `TreatCall` request.
    - if the `TreatCall` request contains multiple prompts.

Note: CTIC(G) does not support `PROMPT` service for the Genesys Legacy Interpreter, or for the `PlayAnnounce&Collect` treatment type.

Music Treatment

CTIC(G) sends a `NETANN`-style `INVITE` to MCP as follows:

```
INVITE sip:annc@<RM-IP-Addr>:5080;DURATION=10;play=http://172.24.129.55:8080
/Test/Resources/Prompts/m.vox;gvp.netann.reportvxml=true SIP/2.0
```

PlayAnnounce & PlayAnnounceAndDigits Treatments

Upon receiving `PlayAnnounce` and `PlayAnnounceAndDigits`, CTIC(G) creates an `INVITE` message in `NETANN` format with the `voicexml` parameter set to the value of the pre-configured CTIConnector parameter `PlayAnnouncePath` or `PlayAnnounceAndDigitsPath`—the path to the pre-canned VXML applications supplied by MCP for handling these treatments.

- If the CTIC(G) parameters are not set, then `voicexml=SCRIPTURL` uses the SCRIPTURL provisioned in the IVR profile.
- The `INVITE` message for `PlayAnnounce`, when the `PlayAnnouncePath` parameter is set, is created in this format:

```
INVITE sip:dialog@172.24.129.86:5080;LANGUAGE=English(US);MSGID=2;
MSGTXT=In%20the%20process%20of%20playing%app;ScriptID=PlayAnnounce;
voicexml=http://localhost/treatments/PlayAnn.vxml SIP/2.0
```

- The `INVITE` message for PlayAnnounce, when PlayAnnouncePath parameter is *not* set, is created in this format:

```
INVITE sip:dialog@172.24.129.86:5080;LANGUAGE=English(US);MSGID=2;
MSGTXT=In%20the%20process%20of%20playing%app;ScriptID=PlayAnnounce;
voicexml=SCRIPTURL SIP/2.0
```

- The `INVITE` message for `PlayAnnounceAndDigits`, when the `PlayAnnounceAndDigitsPath` parameter is set, is created in this format:

```
INVITE sip:dialog@172.24.129.86:5080;LANGUAGE=English(US);MAX_DIGITS=1;
MSGID=2;MSGTXT=In%20the%20process%20of%20playing%app%20and%20Digits;
ScriptID=PlayAnnounceAndDigits;
voicexml=http://localhost/treatments/PlayAnnDigits.vxml SIP/2.0
```

- The `INVITE` message for `PlayAnnounceAndDigits`, when the `PlayAnnounceAndDigitsPath` parameter is *not* set, is created in this format:

```
INVITE sip:dialog@172.24.129.86:5080;LANGUAGE=English(US);MAX_DIGITS=1;
MSGID=2;MSGTXT=In%20the%20process%20of%20playing%app%20and%20Digits;
ScriptID=PlayAnnounceAndDigits;voicexml=SCRIPTURL SIP/2.0
```

CTIC(G) supports the `PROMPT` service for the `PlayAnnounce` treatment type. CTIC(G) expects one of the following options to be present in the TreatCall request:

- `TEXT`—The ASCII text to pronounce using text-to-speech technology (if supported by the IP equipment).

  The INVITE message for PlayAnnounce, when TEXT parameter is set, is created in this format:

```
INVITE sip:dialog@172.21.184.83:5060;TEXT="<ASCII text to
pronounce>";ScriptId=PlayAnnounce;voicexml=file://../treatments/PlayAnn.vxml
SIP/2.0
```

- `IntegerID`—ID of a message to play. CTIC uses NETANN "annc" to play the provided prompt that is identified by "IDintegerID".

  The `INVITE` message for `PlayAnnounce`, when the `ID` parameter is set, is created in this format:

```
INVITE sip:annc@10.10.11.7:5070;play=announcement/<IntegerID> SIP/2.0
```

If CTIC receives a `PlayAnnounce` treatment request from IServer containing `PROMPT` as well as `MSGTXT`, then `MSGTXT` is used for the announcement and `PROMPT`-related parameters are discarded.

- The `INVITE` message for PlayAnnounce in this case is created in this format:

```
INVITE sip:dialog@172.24.129.86:5080;LANGUAGE=English(US);MSGID=2;
MSGTXT=In%20the%20process%20of%20playing%app;ScriptID=PlayAnnounce;
voicexml=SCRIPTURL SIP/2.0
```

VoiceXML Call Reporting

You can configure CTIC(G) to send an indication to Media Control Platform, to report music or an announcement, as a `VoiceXMLcall` to the Reporting Server, in the CDR.

To enable this behavior, add `gvp.netann.reportvxml=true` as an `RURI` parameter in the outgoing SIP `INVITE` message to Resource Manager.

Multiple Trunk Group ID support for CTI Connector (ICM)

In a type 2 or type 8 ICM deployment, the incoming call to ICM is assigned a particular PIM before it is sent to the Voice Response Unit (GVP), and must be routed to the appropriate CTI Connector instance that is connected to the already-selected PIM.

Each Trunk Group ID (TGID) is allocated a specific set of trunks and only the calls targeted for that particular trunk group will share these trunks. If a call comes in without specifying any trunk group then such calls will continue to use trunks from the common pool.

There must be an LRG configured for each CTI Connector with the `capability` parameter defined as `TrunkGroupID=`*`TG1, TG2`*

...where `TG1, Tg2 is a` comma-separated list of TG IDs that are supported by the CTIC application under this LRG.

The CTIC (ICM) instance must be configured with the correct corresponding PIM port along with all the trunk groups supported by that PIM.

CTIC supports multiple TGIDs configuration through the `Ports` parameter in the respective `Tenant` section. List the TGIDs can be listed for the PIM along with TCP listener ports in this format:

port#:[*TGID&TGID*];*port#*:[*TGID&TGID*];

In this example: `6000:1&2;7000;8000:3&4`, the PIM listening on port 6000 supports TGIDs `1` and `2`, the PIM listening on port 7000 has no TGID specified, and the PIM on port 8000 supports TGIDs `3` and `4`.

Set the maximum value of TGID with the configuration parameter `MaxTrunkGroupID` (section `ICMC`). Default = `65535`.

In ICM, each TGID is associated with one set of DNISes having a unique prefix. Whenever a call is received by ICM for a particular TGID, then it picks the DNIS from the corresponding DNIS set and places the call to GVP with the selected DNIS.

So, when a call is received at SIP-Server, it will have only DNIS information but not the actual TGID of the call. GVP must identify a TGID for an incoming call for the provided DNIS. To extract the actual TGID, create a DN of the type `trunk` on the SIP switch for each TGID.

The Trunk DN should be configured with the following parameters.

```
[TServer]prefix="Unique DNIS Prefix"
[TServer]contact="sip:<RM-IPAddr>:<RM Port>
[TServer]request-uri="<uniqueDNISPrefix>@<RM-IPAddr>:<RM Port>;gvp.rm.resource-req.TrunkGroupID=<TGID>
```

CTI Connector (ICM) and ECC Variables

CTI Connector (ICM) reads ECC variables via the parameter `[ICMC]eccvariablelist` and registers them with ICM through the initial `REGISTER_VARIABLES` message.

Configure ECC variable names with their tag values as a comma separated string, in this format: `variable_name:tag_value,variable_name:tag_value`

Specifying Tag Values

Consider this example:

`userECCVar1:6010,userECCVar2:6011,userECCVar3,userECCVar4`

The first and second variables include a tag value and the third and fourth variable do not.

- For each ECC variable, CTIC (ICM) generates a tag value if you don't include one.
- Generated tag values begin with the start tag value configured for the parameter `[ICMC]eccStartTagValue`. Default: `5010`. This value is incremented by 1 for each subsequently generated tag value.
- If the variable `eccStartTagValue` is empty, CTIC (ICM) generates these tag values:

  for `userECCVar3`: `5010`

for `userECCVar4`: `5011`

- If the variable `eccStartTagValue` is 6051, CTIC (ICM) generates these tag values:

  for `userECCVar3`: `6051`

  for `userECCVar4`: `6052`

Specifying the ECC Variable for the Session ID

Beginning with GVP release 8.1.6, you can specify the ECC variable that CTIC (ICM) uses to pass the Session ID to ICM. Configure the parameter `[ICMC]eccSessionIdVarName`.

Specifying this particular ECC Variable is optional. Certain conditions apply to specifying and to *not* specifying it:

If You Specify

The variable name that you specify in the parameter `eccSessionIdVarName` must also be specified in the ECC Variables list. If it is not, or if `eccSessionIdVarName` is specified but empty or null, then the Session ID is not sent in the `NEW_CALL` message.

If You Do Not Specify

If you omit the parameter `eccSessionIdVarName`, then the Session ID is sent through `userSessionId`. This method is backward compatible with pre-8.1.6 releases. The default tag value for `userSessionId` is `5000`.

CTIC (ICM) Parameter Notes

TrunkGroupID

`TrunkGroupID` is set to the value of the `gvp.rm.resource-req.TrunkGroupID`, received as a RURI param in the incoming sip `INVITE` message.

If `gvp.rm.resource-req.TrunkGroupID` is not received, then the value of TrunkGroupID parameter configured in CTIC application is set in the field.

`eccvariablelist` and `eccSessionIdVarname`

These parameters are optional. The GVP 8.1.6 application template defines their default values, for compatibility with previous versions:

```
eccvariablelist=userSessionId:5000
eccSessionIdVarName=userSessionId
```

You can add or modify the ECC variable names as needed. See the "CTI Connector (ICM) and ECC Variables" on page 245.

Chapter

11 Configuring the Supplementary Services Gateway

The Genesys Voice Platform (GVP) Supplementary Services Gateway (SSG) component provides managed initiation of outbound sessions and queuing functionality to accept a batch of outbound session creation requests. It also provides result notifications for requests, including batch requests, that are received from the Trigger Application to determine whether a particular outbound call has succeeded or failed.

This chapter contains the following sections:

Trigger Applications interact with the Supplementary Services Gateway through HTTP. For more information on the HTTP Interface and the HTTP XML schema, see Appendix I, "SSG HTTP Interface," on page 519.

Task Summary: Configuring the Supplementary Services Gateway

Task Summary: Configuring the Supplementary Services Gateway (SSG) summarizes the configuration steps and options to implement SSG functionality in your GVP deployment.

Task Summary: Configuring the Supplementary Services Gateway

Objective	Related Procedures and Actions
Configure the Supplementary Services Gateway to initiate outbound calls.	The key configuration sections are `HTTP`, `Tenant1` and `SSG` (see page 250).
Enable Reporting Server to poll the Supplementary Services Gateway data.	Install LCA and SNMP Master Agent on the Supplementary Services Gateway server.
Configure reporting.	See "Configuring Reporting" on page 63.
Configure logging.	See "Configuring Logging" on page 66.
Customize client-side communication ports.	See "Configuring Client-Side Connections" on page 72.
Install and configure security certificates.	Create a security certificate on Windows and Linux to enable the Transport Layer Security (TLS) connection between the Supplementary Services Gateway and SIP Server. For installation and configuration procedures, see Chapters 16-18 in the *Genesys 8.1 Security Deployment Guide*.

Important Supplementary Services Gateway Configuration Options

This section describes the key configuration options that you either must or might want to customize.

Configure the options on Genesys Administrator on the `Provisioning > Environment > Applications > <Supplementary Services Gateway> > Options` tab. For the detailed steps to configure option settings, see Procedure: Viewing or modifying GVP configuration parameters, on page 34.

The configurable SSG parameters are in the following configuration sections:

- `Common`—Parameters that determine whether or not IPv6 communication is used between the SSG and SIP Server.
- `fm`—Parameters that determine the behavior of the HTTP Client.
- `HTTP`—Parameters that determine the embedded HTTP server behavior.
- `SSG`—Parameters that determine SSG behavior.
- `Tenant1`—Parameters that determine Tenant association to Trunk Group and Routing Point for multi tenancy support.
- `ems`—Parameters that determine Reporting behavior for call detail records (CDRs) and metric. (See Table 6 on page 64.)

- log—Parameters that determine behavior for Management Framework logging. (See “Service Quality Analysis (SQA)” on page 65.)

Table 30 provides information about important Supplementary Services Gateway parameters that are not described in Chapter 3 on page 41. Table 30 provides parameter descriptions as well as the default parameter values that are preconfigured in the Supplementary Services Gateway Application object.

Unless indicated otherwise, all changes take effect on restart.

For information about all the available configuration options for the Supplementary Services Gateway, see the *Genesys Voice Platform 8.1 Configuration Options Reference.*

Table 30: Selected SSG Configuration Options

Option Name	Description	Valid Values and Syntax
	Common Section	
Enable IPv6 for SIP Server connection	Specifies whether the IPv6 communication between SSG and SIP Server is enabled or disabled.	• 1 (true) • 0 (false) **Default value:** 0
	fm Section	
HTTP Proxy	Specifies the HTTP Proxy that will be used by the Fetching Module.	A string. **Default value:** Empty
	HTTP Section	
HTTPS Certificate File Name	Specifies the name of the HTTPS Server Certificate file.	Any string of characters. **Default value:** $InstallationRoot$/config/x509_certificate.pem
HTTPS Cert Key File	Specifies the name of the HTTPS Server Certification Key file.	Any string of characters. **Default value:** $InstallationRoot$/config/x509_private_key.pem
HTTPS Cert Password (optional)	Specifies the password to access the Certificate Key file.	Any string of characters. **Default value:** Empty
Secure Protocol Version	Specifies the name of the secure protocol and version.	• SSLv23 • SSLv3 • SSLv2 • TSLv1. **Default value:** SSLv23

Table 30: Selected SSG Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Default HTTP Page	Specifies the default HTTP page that SSG uses to service incoming outbound requests.	Any string of characters. **Default value:** `SSG`
HTTP Port	Specifies the port that SSG uses to receive HTTP requests from trigger applications.	Any integer. **Default value:** `9800`
HTTPS Port	Specifies the port that SSG uses to receive HTTPS requests from trigger applications.	Any integer. **Default value:** `9801`
`SSG` Section		

FIPS Mode Enabled	Enables FIPS mode in SSG.	`True`
		`False`
		Default value: `False`
		Changes take effect: start/restart

Request Batch Size	Specifies the number of requests that can be fetched from the database into memory in a given cycle. If set to `TotalPorts`, SSG uses the GVP total port capacity received from the SIP Server `EventResourceInfo` as the batch limit. If set to `AvailPorts`, SSG uses the current available port capacity received from the SIP Server `EventResourceInfo` as the batch limit. Otherwise, you can configure any integer value as a string for the batch limit.	A string of characters. **Default value:** `TotalPorts`
Clean Interval	Specifies the time (in seconds) that determines the frequency in which SSG removes expired or completed requests from the database.	An integer in the range of 30–900. **Default value:** `180`
Queue Low Watermark	Specifies when to activate the next fetch cycle from the database. The algorithm uses this value to calculate the percent of the total batch limit. When the in-memory queue falls below this number, the next fetching cycle starts.	An integer in the range of 1—99. **Default value:** `25`

Table 30: Selected SSG Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Initiated Call Retry Flag	Specifies whether the Initiated Call requests are retried when the SSG process starts back up.	• `0` (Do Not Retry Initiated Requests) • `1` (Retry Initiated Requests) • `2` (Purge New And Initiated Requests) • `3` (Purge All Requests Without Notification) Default value: `1`
Max DB Connection Pool Size	Specifies the maximum number of database connections SSG will use.	Any integer. **Default value:** `7`
Min DB Connection Pool Size	Specifies the minimum number of database connections SSG will use.	An integer. **Default value:** `3`
Maximum Attempts Limit	Specifies the upper bound of the `MaxAttempts` parameter that the trigger application is allowed to use in the HTTP requests to SSG.	Any integer. **Default value:** `25`
Time to Live Limit	Specifies the upper bound of the `TimeToLive` parameter, in minutes, that the trigger application is allowed to use in the HTTP requests to SSG.	Any integer. **Default value:** `1440`
Application Slot Calculation	Specifies the number of records that are allotted for an application in each database fetch.	• `Proportionate`—Divide the batch limit among the applications in the same ratio as their pending requests. • `Equal`—Divide the batch limit among the applications equally. **Default value:** `Proportionate`
Equal Priority Between Old and New	Specifies the priority that is given to new and old requests for applications. **Note:** If this option is set to `True`, increased database fetches can result in performance degradation.	• `True`—Give equal priority. • `False`—Do not give equal priority. **Default value:** `False`

Table 30: Selected SSG Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Port Load Factor	Specifies the number of outbound calls that SSG initiates at a time. The value that is specified is a percentage of the current GVP available port capacity.	An integer in the range of 1–100. **Default value:** `100`
Next Retry Interval	Specifies the retry interval, in seconds, for call if temporary internal errors occur.	An integer in the range of 1–65536. **Default value:** `10`
Resource DN Registration Failure Recovery Interval	Specifies the frequency, in seconds, that SSG will attempt to re-register the Resource DN with SIP Server if the Resource DN registration is lost during runtime, or fails at SSG start up.	Any integer. **Default value:** `120`
SIPS Connection Failure Recovery Interval	Specifies the frequency, in seconds, that SSG will attempt to connect to SIP Server if the connection to SIP Server is lost during runtime, or fails at start up.	Any integer. **Default value:** `120`
Request Acceptance Time-Out on Resource DN Registration Failure	Specifies the timeout interval, in seconds, that SSG rejects requests from trigger applications if DN registration with SIP Server fails.	Any integer. **Default value:** `900`
Request Acceptance Time-Out on SIPS Connection Failure	Specifies the timeout interval, in seconds, that SSG rejects requests from trigger applications if the connection with SIP Server fails.	Any integer. **Default value:** `900`
Max Calls/Sec to SIP Server	Specifies the maximum number of calls per second that the Supplementary Services Gateway initiates with SIP Server.	An integer in the range of 1–200. **Default value:** `30`
	`Tenant1` Section	
Trunk Group DN	Specifies the DN for outbound calls to use if using the Next Generation Interpreter. The Trunk Group DN (`TGDN`) option value is used as the Tenant name.	Any string of characters. **Default value:** Empty
Routing Point DN	Specifies the DN for outbound calls to use if using the Legacy GVP Interpreter, or CTI Connector.	Any string of characters. **Default value:** Empty
Name of Access Group	Specifies which access group controls Digest Authentication enabling for the tenant.	Any string of characters. **Default value:** Empty

Note: If SSG is to support multiple tenants, copy the `Tenant1` section to `Tenant2`, `Tenant3`, etc and appropriately configured. See the *Framework 8.1 Genesys Administrator Help* for information on how to copy sections.

Call Progress Detection

There are three distinct call flows that SIP Server uses for making outbound calls.

- CPD is not performed
- CPD is performed on the Media Gateway
- CPD is performed on the Media Server

The Supplementary Services Gateway does not know if Call Progress Detection (CPD) is performed at the Media Gateway or at the Media Server. CPD can be started with the first media packets received, or only after the call is connected.

SIP Server selects a CPD provider (Media Gateway or Media Server), and sets the appropriate CPD mode. If it is configured for both providers, the Media Gateway takes precedence as the CPD provider. SIP Server is also responsible for receiving the CPD result from the CPD provide, and pass it to the Supplementary Services Gateway through the corresponding T-Event.

SIP Server sends the following CPD results to the Supplementary Services Gateway:

- Pre-connect phase
  - GeneralError
  - Busy
  - NoAnswer
  - SitDetected
  - SitInvalidnum
  - SitVacant
  - SitIntercept
  - SitUnknown
  - SitNocircuit
  - SitReorder
- Post-connect phase
  - Unknown
  - AnsweringMachineDetected
  - FaxDetected,
  - Voice

The Supplementary Services Gateway applies any of the following CPD algorithms during a call:

- no_progress_detection
- no_am_detection
- positive_am_detection
- voice_priority_detection (full_positive_am_detection)
- am_priority_detection (accurate_am_detection)
- telephony_preset

The Trigger Application passes CPD control parameters to the Supplementary Services Gateway in the create request. All CPD attributes are optional, and if they are not available in the CreateRequest, the SIP Server uses its default configuration. Table 31 describes the various CPD attributes in the CreateRequest:

Table 31: CPD Attributes

Attribute	Description
record	Specifies whether the CPD part of the call should be recorded. This is mapped to extension `record` in `TMakePredictiveCall` API. • `True` —Record CPD • `False` —Do not record CPD **Default value:** `False`
preconnect	Specifies when to start CPD. This value is mapped to the `cpd-on-connect` extension in `TMakePredictiveCall` request. • `True`—Start CPD when the first RTP packet is received. • `False`—Start CPD when the call is connected. **Default value:** `False`
rnatimeout	Specifies the timeout interval (in seconds) for the Ring No Answer scenario. This value is passed to SIP Server in the `TMakePredictiveCall` request. SIP Server starts the timer after receiving the `180 Ringing` message from the external party. If the timer expires, and the call is not connected, SIP Server disconnects the call, and sends the `EventReleased` TEvent with the `CallState` attribute set to `NoAnswer` to the Supplementary Services Gateway.

Table 31: CPD Attributes (Continued)

Attribute	Description
postconnecttimeout	Specifies the timeout interval (in seconds or milliseconds) for the post connect scenario. This value is passed to SIP Server in the `TMakePredictiveCall` request. SIP Server starts the timer starts when the outbound call is connected. If the timer expires without a call result detected, SIP Server sends the `EventEstablsihed` TEvent with the `CallState` attribute set to `Unknown` to the Supplementary Services Gateway.
detect	Specifies the action that SSG is to take with the outbound call when CPD is detected. • None (default)—Do not request CPD. Start the IVR as soon as call is connected. This maps to no_progress_detection. • All—Start the IVR regardless of the detection result (VOICE/MACHINE/FAX). This maps to full_positive_am_detection. • Voice—Start the IVR only if the detection result is VOICE. The call is re-attempted if the detection result is MACHINE. It is not re-attempted if the detection result is FAX. This maps to full_positive_am_detection. • AM—Start the IVR only if the detection result is MACHINE. The call is re-attempted if the detection result is VOICE. It is not re-attempted if the detection result is FAX. This maps to accurate_am_detection, and requires the following setting on SIP Server (am-detected: connect). • FAX—Start the IVR only if the detection result is FAX. The call is re-attempted if the detection result is VOICE/MACHINE. This maps to no_am_detection, and requires the following setting on SIP Server (fax-detected: connect). • Voice,AM,FAX —Can be combined by using comma separation (for example, voice,am or am,fax or voice,am,fax). If any of the comma-separated values are detected, connect to IVR; otherwise, retry the call. This maps to full_positive_am_detection.

Chapter

12 Configuring the PSTN Connector

The Genesys Voice Platform (GVP) PSTN Connector component provides connectivity to traditional telephony environments, such as Public Switched Telephone Networks (PSTN), and PBXs or ACDs. It uses Dialogic hardware and software to interface with the PSTN network for processing Time Division Multiplexed (TDM) calls.

This chapter contains the following sections:

Task Summary: Configuring the PSTN Connector

Task Summary: Configuring the PSTN Connector summarizes the configuration steps and options to implement PSTN functionality in your GVP deployment.

Task Summary: Configuring the PSTN Connector

Objective	Related Procedures and Actions

Configure the PSTN Connector to manage inbound and outbound calls.	The key configuration parameters are:
	`RouteType`
	`Signaling Type`
	`Channels`
	`SIP Destination IP Address`
	`SIP Destination Port Number`
	`Supported Local Codec Type`
	See Table 32 for other important PSTN Connector options.

Configure Trunk DN.	See the "Post-Installation Configuration of GVP Components" chapter of the *Genesys Voice Platform 8.1 Deployment Guide*.
Configure reporting	See "Configuring Reporting" on page 63.
Configure logging.	See "Configuring Logging" on page 66.
Customize client-side communication ports.	See "Configuring Client-Side Connections" on page 72.

Important PSTN Connector Configuration Options

This section describes the key configuration options that you either must or might want to customize.

Configure the options on Genesys Administrator on the `Provisioning > Environment > Applications > <PSTN Connector> > Options` tab. For the detailed steps to configure option settings, see Procedure: Viewing or modifying GVP configuration parameters, on page 34.

The configurable PSTN Connector parameters are in the following configuration sections:

- `MediaManager`—Parameters determine media behavior.
- `GatewayManager`—Parameters determine SIP behavior.
- `DialogicManager`—Parameters determine Dialogic and CPA behavior.
- `DialogicManager_CPD`—Parameters determine T-Server and CPD behavior.
- `DialogicManager_Route1`—Parameters determine the routing through Dialogic behavior.
- `ems` (see Table 6 on page 64)—Parameters determine Reporting behavior for call detail records (CDRs) and metrics.

- `log` (see "Configuring Logging" on page 66)—Parameters determine behavior for Management Framework logging.

Table 32 provides information about important supplementary PSTN Connector parameters that are not described in Chapter 3 on page 41. Table 32 provides parameter descriptions as well as the default parameter values that are preconfigured in the PSTN Connector `Application` object.

Unless indicated otherwise, all changes take effect on restart.

For information about all the available configuration options for tee PSTN Connector, see the *Genesys Voice Platform 8.1 Configuration Options Reference.*

Table 32: Selected PSTN Connector Configuration Options

Option Name	Description	Valid Values and Syntax
	`MediaManagerSection`	
DTMF Payload Type	Specifies the payload or encoding type of DTMF packets.	Any integer. **Default value:** `101`
Supported Local Codec Type	Specifies the codec that is used by the TDM trunks. The RTP stream generated by PSTN Connector uses the same codec.	• `Mulaw` • `Alaw` **Default value:** `Alaw`
	`GatewayManagerSection`	
PSTN Connector SIP Port	Specifies the local SIP port for PSTN Connector to use for SIP communication.	Any integer. **Default value:** `5170`
SIP Destination IP Address	Specifies the end point to send SIP calls to when PSTN Connector receives TDM calls and translates them to SIP.	`<Host name or IP address>` **Default value:** Empty
SIP Destination Port Number	Specifies the SIP port number of the end point that is configured in the `SIP Destination IP Address` parameter.	Any integer. **Default value:** Empty
Enable Session Timer	Specifies whether to enable session timers.	• `True` • `False` **Default value:** `True`
Session Timer Interval	Specifies the time interval, in seconds, for which a call session is refreshed. If not set, the session will expire.	An integer in the range of 90–86400 **Default value:** `1800`

Table 32: Selected PSTN Connector Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
	`DialogicManager` **Section**	
ATT Conference Sleep Time Before Answer	Specifies the duration, in milliseconds, for which PSTN Connector must wait before connecting the agent, and the caller in an AT&T Conference Transfer. Set this parameter when whisper to agent is required, otherwise, set the value to `0`.	Any integer. **Default value:** `5000`
CPA Failure Timeout	Specifies the maximum time, in milliseconds, to wait for positive answering machine detection.	Any integer. **Default value:** `4000`
CPA Max Inter-ring Timeout	Specifies the maximum time, in milliseconds, to wait between consecutive ring-backs before disconnecting.	Any integer. **Default value:** `8000`
CPA Min Inter-ring Timeout	Specifies the minimum ring duration, in milliseconds, for answering machine detection.	Any integer. **Default value:** `1900`
CPA Option	Specifies whether to choose custom enabled CPA parameters.	• `0`—Enable CPA Detection • `1`—Enable custom CPA detection Springware • `2`—Enable custom CPA detection DMV **Default value:** `0`
CPA PAMD Option	Specifies the level of accurate answering machine detection.	• `0`—Quick AM detection • `1`—Full AM detection • `2`—Accurate AM detection **Default value:** `2`
CPA Qualification Templates	Specifies which template PSTN Connector is to use for AM Detection.	• `0`—Qualification Template1 • `1`—Qualification Template2 **Default value:** `0`
CPA Start Delay in MSec	Specifies the time, in milliseconds, to wait before starting cadence, frequency, or positive voice detection.	Any integer. **Default value:** `250`

Table 32: Selected PSTN Connector Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Disable Custom Tones before CPA	Specifies whether to delete custom tones before performing CPA.	• True • False **Default value:** False Set this value to True for Springware boards.
Minimum Download Size for Play	Specifies the minimum amount of data to be buffered before starting the play on the TDM side.	Any integer. **Default value:** 32768
Ringback Filename	Specifies the audio file to use for playing ringback tone. The file format must be 8Khz PCM Mulaw or A-law, and must contain a single ring with the desired trailing silence. If this parameter is not configured, the value configured for AlawIndexFileName or UlawIndexFileName is used based on the protocol configured (A-law for E1, and Ulaw for T1).	Any string of characters. **Default value:** m12.vox
Default DNIS Value	Specifies default DNIS number when the DNIS information is not available in behind-the-switch configurations.	Any string of characters. **Default value:** NoDNIS
DialogicManager_CPD Section		
IP Address of Primary TServer	Specifies the IP address of the primary T-Server.	<Host name or IP address> **Default value:** Empty
Primary TServer Listening Port	Specifies the primary T-Server's port number.	Any integer. **Default value:** 0
IP Address of Backup TServer	Specifies the IP address of the backup T-Server.	<Host name or IP address> **Default value:** Empty
Backup TServer Listening Port	Specifies the backup T-Server's port number.	Any integer. **Default value:** 0
Use TServer to Make Calls	Specifies whether to use T-Server to make outbound calls. **Note:** If set to False, Dialogic will make outbound calls.	• True • False **Default value:** False

Table 32: Selected PSTN Connector Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
FAX2 Tone as Answering Machine	Specifies whether to accept FAX2 tones as answering machine.	• `True` • `False` **Default value:** `False`
Offhook Delay	Specifies the time, in milliseconds, to wait before going off hook. If negative, go off-hook first, wait the specified time, then dial. If positive, dial first, wait the specified time, then set the channel off-hook. **Note:** This parameter is valid only if `Use TServer to Make Calls` is set to `True`.	Any integer. **Default value:** `100`
Postconnect Priority	Specifies which application has priority if conflicting CPD results are received.	• `TServer` • `Dialogic` **Default value:** `TServer`
Preconnect Priority	Specifies which application has priority for preconnect CPD events.	• `TServer` • `Dialogic` **Default value:** `TServer`
TServer Reconnect Timeout	Specifies the time, in milliseconds, to wait before the reconnecting to the dialer.	Any integer. **Default value:** `20000`
Wait for Offhook Confirmation	Specifies whether to wait for the off-hook confirmation event from T-Server before dialing. **Note:** This parameter is valid only if `Offhook Delay` is set to a negative value.	• `True` • `False` **Default value:** `False`
`DialogicManager_Route1` Section		
Enable CPD Library	Specifies whether CPD results are to be received from T-Server. If set to `False`, CPD results are received from Dialogic.	• `True` • `False` **Default value:** `False`
Route Description	Specifies the description of the route configured.	• `Inbound Route` • `Outbound Route` • `Inbound Outbound Route` **Default value:** `Inbound Route`

Table 32: Selected PSTN Connector Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Dial Prefix	Specifies the number to prepend to the number dialed. **Note:** This parameter is used only when the Network Type is `PSTN`.	Any string of characters. **Default value:** `1`
Range of Directory Numbers	Specifies, in a comma or dash separated list, the DN range for the route. For example, `101-110, 115, 120-130`. **Note:** This parameter is used only if TServer is used for CPD (`Enable CPD Library` is `True`).	Any string of characters. **Default value:** Empty
ISDN Numbering Plan	Specifies the encoding of the Calling/Called Party IE Numbering Plan in the outgoing setup. Used for outbound ISDN routes.	• `0x00`—Unknown • `0x01`—ISDN E.164 • `0x02`—Telephony E.163 • `0x09`—Private **Default value:** `0x01`
ISDN Numbering Type	Specifies the encoding of the Calling/Called Party IE Numbering Type in the outgoing setup. Used for outbound ISDN routes.	• `0x00`—Unknown • `0x01`—International Number • `0x02`—National Number • `0x04`—Subscriber Number **Default value:** `0x02`
Max Digits to Dial	Specifies the number of digits to dial. If Network Type is set to `PSTN`, then this value must be `7`,`10`, or `11`. If Network Type is set to `Enterprise`, then this parameter can have any value. If this value is set to `0`, there is no maximum number of digits to dial. If this value is missing or invalid, the default is used.	Any integer. **Default value:** `7`

Table 32: Selected PSTN Connector Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Media Resource Board to Use for CSP	Specifies the board number used for Continuous Speech Processing (CSP) when using JCT boards. If no value is specified, it defaults to the same board as used for the network port. For ISDN JCT boards, a different board must be configured for CSP other than the network board. **Note:** Routes with this parameter configured must be on a single board.	Any integer. **Default value:** `0`
Network Type	Specifies the type of telephony network the route is connected to.	• `0`—PSTN • `1`—Enterprise (PBX/ACD) **Default value:** `0`
New Call Confirmation	Specifies when to collect digits for inbound calls, and when to start CPA for outbound calls. For inbound calls, if set to After Answer, the call is accepted and answered before collecting the DNIS. If set to Before Answer, the ANI and DNIS are collected before the called is answered. For outbound calls, if set to After Answer, CPA is started immediately after dialing. If set to Before Answer, CPA is started after the call is connected. **Note:** If you are using the groundstart protocol, you must set New Call Confirmation to After Answer.	• `0`—Before Answer • `1`—After Answer **Default value:** `0`
Max Digits to Receive in Overlap Receive Mode	Specifies the maximum number of digits (ANI + DNIS + delimiters) to receive in overlap receive mode.	Any integer. **Default value:** `0`
Enable ISDN Overlap Receive	Specifies whether to enable ISDN Overlap Receive mode.	• `0`—True • `1`—False **Default value:** `0`
Channels	Specifies the ports used for this route.	`<Card:PortRange, Card:PortRange>` **Default value:** Empty

Table 32: Selected PSTN Connector Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Signaling Type	Specifies the Route Signaling type.	• 0—T1-ISDN PRI • 1—Analog • 2—E1-ISDN PRI • 3—T1-RobbedBit • 4—E1-CAS **Default value:** Empty
T1-RB ANI DNIS Delimiter	Specifies the character that separates ANI from DNIS in the incoming call data.	Any single character. **Default value:** *
T1-RB ANI/DNIS Order	Specifics which order to receive the ANI and DNIS. **Note:** This parameter is ignored if the signaling protocol is not T1-RobbedBit.	• 0—No ANI/DNIS • 1—DNIS only • 2—DNIS followed by ANI • 3—ANI followed by DNIS **Default value:** 1
T1-RB Protocol File	Specifies the Dialogic T1 configuration file to use. For example, use `us_mf_loop_io` for loopback testing. **Note:** This parameter is mandatory for T1 robbed-bit signaling.	Any string of characters. **Default value:** `pdk_dmv`
T1-RB Remove ANI/DNIS Delimiter	Specifies whether the ANI/DNIS delimiters are to be removed. **Note:** This parameter is ignored if the signaling protocol is not T1 robbed-bit signaling.	• `True` • `False` **Default value:** `True`

Table 32: Selected PSTN Connector Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Two Channel Transfer Type	Specifies the type of two channel transfer to use.	• Empty • `NortelRLT` • `TBCT` • `ECTExplicit` • `ECTExplicit_AUS` • `ECTExplicit_NZ` • `ECTExplicit_UK` • `QSigPathReplace` **Default value:** Empty
Route Type	Specifies the type of call direction to use.	• Empty • `0`—Inbound • `1`—Outbound • `2`—In/Out **Default value:** Empty

Note: To configure additional routes, copy the `DialogicManager_Route1` section to `DialogicManager_Route2`, `DialogicManager_Route3`, etc and configure appropriately. See the *Framework 8.1 Genesys Administrator Help* for information on how to copy sections.

Chapter

13 Configuring the Fetching Module and Squid Proxy

This chapter describes the requirements to configure the Fetching Module and Third-Party Squid caching proxy in your Genesys Voice Platform (GVP) deployment.

It contains the following sections:

Task Summary: Configuring the Fetching Module and Squid

Task Summary: Configuring the Fetching Module and Squid summarizes the configuration steps and options to configure the Fetching Module and Squid Caching Proxy in your GVP deployment.

Notes: You do not need to change the default configuration for the Fetching Module to work.

With GVP version 8.1.2 and higher, the Fetching Module functionality has been included in the Media Control Platform, and Squid is an optional component.

Task Summary: Configuring the Fetching Module and Squid

Objective	Related Procedures and Actions
Modify the Squid caching proxy configuration, if required for the following reasons: • To configure for a second-level proxy. • You cannot configure the Web Server to deliver `Expires` headers, and you need to change the Squid refresh-pattern rules. • You are following the recommended practice of denying access to all ports except those that you have identified as safe, but the ports you are using for HTTP or, if applicable, HTTPS and SSL are not the ports that are configured as safe ports and SSL ports, respectively, in the default Squid configuration file.	See "Configuring the Squid Caching Proxy" on page 272. For more information about configuring Squid, which is an open-source product, see online sources.
Schedule a task to rotate the Squid Caching Proxy service logs.	See the chapter about post-installation activities in the *Genesys Voice Platform 8.1 Deployment Guide.*

Important Fetching Module Configuration Options

This section describes the key configuration options that you may want to customize.

Configure the options in Genesys Administrator on the `Provisioning > Environment > Applications > <Fetching Module> > Settings` tab of each Fetching Module `Application` in your deployment. For the detailed steps to configure option settings, see Procedure: Viewing or modifying GVP configuration parameters, on page 34.

Except for some `ems` options, all changes to Fetching Module options take effect on restart.

Note: If you restart the Fetching Module, you must also stop and then restart the associated Media Control Platform or Call Control Platform.

The Fetching Module configuration options are in the following configuration sections:

- `ems` (see Table 6 on page 64)—Parameters determine Reporting behavior for call detail records (CDRs) and metrics.
- `log` (see "Configuring Logging" on page 66)—Parameters determine behavior for Management Framework logging.
- `iproxy`—Parameters determine the behavior of the `pwproxy` process (the Fetching Module as an HTTP or HTTPS proxy).
- `snmp` (see "Configuring SNMP" on page 72)—Parameters determine the behavior of SNMP.

Table 33 provides information about some important Fetching Module parameters in the `iproxy` section. Table 33 provides parameter descriptions as well as the default parameter values that are preconfigured in the Fetching Module `Application` object.

For information about all the available configuration options for the Fetching Module, see the *Genesys Voice Platform 8.1 Configuration Options Reference.*

Table 33: Selected Fetching Module Configuration Options

Option Name	Description	Valid Values and Syntax
	`iproxy` Section	
HTTP Proxy HTTPS Proxy	The IP address and port that the HTTP or HTTPS proxy will use. • If `HTTP_proxy` is disabled (empty), the `pwproxy` will not use an HTTP proxy. • If `HTTPS_proxy` is disabled (empty), the `pwproxy` will not use an HTTPS proxy.	`<Proxy IP address>:<port>` **Default value:** • HTTP proxy—`127.0.0.1:3128` • HTTPS proxy—Empty
No Cache URL Substring	A space-separated list of substrings that, if contained in a URL, will ensure that the page is not cached.	`<Substring1> [<Substring2>...]` **Default value:** `cgi-bin`
ssl_*	Various options starting with `ssl_`, to configure aspects of Secure Socket Layer (SSL) functioning. For more information about configuring secure communications in GVP, see "Enabling Secure Communication" on page 46.	

Configuring the Squid Caching Proxy

In general, the default Squid configuration file should be suitable for most installations. However, there are three reasons why you might need to modify the Squid configuration file:

- You need to configure for a second-level proxy.
- You cannot configure your Web Server to deliver `Expires` headers, and you wish to change the Squid defaults for the expressions Squid tries to match in SIP `request-URI` headers to control refresh behavior.
- You need to configure non-standard "safe" ports or SSL ports for HTTP and SSL.

  By default, the Squid configuration file:
  - Identifies the following as SSL ports: port 443 563.
  - Identifies the following as a "safe" port for HTTP: port 80.
  - Denies requests to unknown ports (in other words, ports that are not identified as "safe").
  - Denies CONNECT to other than SSL ports.

The following procedure describes how to modify the Squid configuration file.

Procedure: Modifying the Squid Configuration

Purpose: To modify the configuration file of the caching proxy to enable a second-level proxy, to specify different refresh-pattern rules for matching `Request-URI` expressions, or to enable non-standard "safe" and SSL ports.

Perform this procedure on each Media Control Platform and Call Control Platform host in your deployment whose behavior you want to modify.

Prerequisites

- You have the required permissions to modify files in the Squid configuration directory.

Start of procedure

1. Back up the original configuration file in case you need to restore it later.
2. Open the Squid configuration file in a text editor.

   `c:\squid\etc\squid.conf` (Windows)

   `/usr/local/squid/etc/squid.conf` (Linux)

Second-Level Proxy

3. To configure for a second-level proxy, add the following lines:

```
cache_peer <parentcache.yourdomain.com> parent <port> 0 noquery
default
acl local-servers dstdomain <yourdomain.com>
acl all src 0.0.0.0/0.0.0.0
never_direct deny local-servers
never_direct allow all
```

Where:

- `<parentcache.yourdomain.com>` is the next proxy in the chain.
- `<port>` is the port number on which the parent cache is listening.
- `<yourdomain.com>` identifies the domains that should not go through the parent proxy.

Refresh-Pattern Rules

4. To modify the Squid refresh-pattern rules, add or reorder as many lines as you require, to specify the refresh-patterns in the order in which you want Squid to consider them. Use the following format for each line:

```
refresh_pattern [-i] regex <min> <percent> <max> [<options>]
```

Where:

- `<min>` is the amount of time, in minutes, that an object without an explicit expiry time should be considered fresh. The recommended value is `0`. Any non-negative values may cause dynamic applications to be erroneously cached unless the application designer has taken the appropriate actions.
- `<percent>` is a percentage of the age of the object (where age is the time since last modification) that an object without an explicit expiry time will be considered fresh.
- `<max>` is the upper limit, in minutes, for how long objects without an explicit expiry time will be considered fresh.
- `<options>` are one or more of the following:
  - `override-expire`—Enforces `min` age even if the server sent an `Expires:` header. Doing this violates the HTTP standard. Enabling this feature could make you liable for problems, which it causes.
  - `override-lastmod`—Enforces `min` age even on objects that were modified recently.
  - `reload-into-ims`—Changes client `no-cache` or `reload` to `If-Modified-Since` requests. Doing this violates the HTTP standard. Enabling this feature could make you liable for problems, which it causes.
  - `ignore-reload`—Ignores a client `no-cache` or `reload` header. Doing this violates the HTTP standard. Enabling this feature could make you liable for problems, which it causes.

The default is:

```
refresh-pattern. 0 20% 4320
```

Configure "safe" and SSL ports

5. In the `ACCESS CONTROLS` section:
   a. Add or modify access control lines as required to ensure that the following lines match the applicable port configurations in your deployment:
      - `acl Safe_ports port <safe port> #http`
      - `acl Safe_ports port <safe port> #https`
      - `acl SSL_ports port <SSL port>`

   b. To deny requests to unknown ports (in other words, ports that have not been identified as "safe"), verify that the following line has not been commented out or deleted:

      ```
      http_access deny !Safe_ports
      ```

   c. To deny connections to other than SSL ports, verify that the following line has not been commented out or deleted:

      ```
      http_access deny CONNECT !SSL_ports
      ```

6. Save the file.

Execute the Update

7. Do one of the following to execute the update:
   - Execute the following command to force a re-read of the configuration file:

     `C:\squid\sbin\squid.exe -k reconfigure -n squidNT` (windows)

     `/usr/local/squid/bin/squid - k reconfigure` (linux)
   - Restart Squid.

   Restarting Squid will not affect the Fetching Module. However, if a fetch is in progress, it may fail.

Note: Changes to the configuration file are not reflected in the running configuration until you execute this command

End of procedure

Chapter

14 Configuring the Reporting Server

This chapter provides information about how to configure the Genesys Voice Platform (GVP) Reporting Server. It contains the following sections:

Task Summary: Configuring the Reporting Server

Task Summary: Configuring the Reporting Server summarizes the configuration steps and options to set up the Reporting Server and to customize EMS Reporting behavior in your GVP deployment.

Task Summary: Configuring the Reporting Server

Objective	Related Procedures and Actions
Verify directory paths for: • Java Message Service (JMS) for CDR, OR summary, and call events reporting. • The Atomikos distributed transactions processing engine.	If necessary, modify settings for options in the following configuration sections: • `messaging`. In particular, verify the path to the directory that ActiveMQ uses for persistent queuing (`activemq.dataDirectory`).

Task Summary: Configuring the Reporting Server (Continued)

Objective	Related Procedures and Actions
Configure Reporting Server to operate without a database.	If required, set the `rs.nodb.enabled` parameter in the `persistence` section to `true`.
Configure logging.	See "Configuring Logging" on page 66.
Configure the maximum size of reports for different levels of granularity.	If necessary, modify settings for the `rs.query.limit.<granularity period>` options in the `reporting` configuration section. For more information, including a summary of the default maximums, see "Configuring Reporting, by Granularity".
Configure the maximum size of Call Detail Record (CDR) and Call Events reports.	• If necessary, modify the `cdr.max-page-size` option, to configure a suitable value for your deployment, for the maximum number of CDR or metrics records per page The default is `100`. • Consider also the `cdr.max-page-count` option, for the maximum number of pages per report. The default is `10`.
Configure database retention policies.	Use the Database Retention Policy Wizard (see "Data Retention Policy Wizard" on page 138) to configure the database retention policies. For more information, see "Configuring Database Retention Policies" on page 278.
Configure Reporting Server behavior in general.	See "Important Reporting Server Configuration Options" on page 280.
Customize client-side communication ports.	See "Configuring Client-Side Connections" on page 72.
(Optional) Configure HTTPS to secure access to Reporting Services.	See "Controlling Access to Reporting Services" on page 287.
Verify that Genesys Administrator displays GVP reports requested from the `Monitoring > Voice Platform` navigation panel.	If necessary, configure or modify the connection between Genesys Administrator and the Reporting Server. For more information, see the *Genesys Voice Platform 8.1 Deployment Guide*.
If running on Oracle in partitioned mode, optimize performance.	Disable the `GATHER_STATS_JOB` before installing the RS database to ensure that inaccurate statistics are not associated with the staging tables. For more information, see the *Genesys Voice Platform 8.1 Deployment Guide*.

Configuring Reporting, by Granularity

Granularity refers to the degree of aggregation in a given summary report. For example, a request for a Call Peak report at the granularity level of `month` will return a peak value for each month in the requested time range. A request for a Call Peak report at the granularity level of `week` will return a peak value for each week in the requested time range.

The Reporting Server supports reporting at the following levels of granularity:

- 5-minute
- 30-minute
- Hour
- Day
- Week
- Month

If the requested time period does not encompass an integral unit that matches the specified granularity level, then the Reporting Server expands the time to cover an integral number. For example, if the granularity is `day,` a request for a report from 2008/01/01 00:00 – 2008/01/01 14:00 will be expanded to 2008/01/01 00:00 – 2008/01/02 00:00.

The Reporting Server normalizes `From` and `To` parameters that specify the time range in a reporting request, so that they lie on time unit boundaries that match the granularity level. For example, if the granularity is `hour`, then the Reporting Server normalizes the start time and end time of the report so that they point to the beginning of an hour. In this case, a start or end time request for 11:30 will be normalized to 11:00.

Ensure that the values that are set for the `rs.query.limit.<granularity>` configuration options in the `reporting` section (see page 286) are appropriate for your reporting purposes and environment.

Table 34 summarizes the default values for the `reporting.rs.query.limit.<granularity>` configuration options. Each of these options specifies the maximum number of units of a particular aggregation period that will be included in reports at that aggregation period's level of granularity. For example, if you request a 5-minute report covering 2008/11/17–2008/11/19, the range is truncated to cover the day maximum (2008/11/17 00:00–2008/11/18 00:00).

Table 34: Default Maximum Units, by Granularity Level

Aggregation Period	Maximum Number of Units
5 minutes	288 (5-minute periods, equals 1 day)
30 minutes	48 (30-minute periods, equals 1 day)
Hour	168 (hours, equals 1 week)
Day	92 (days)
Week	53 (weeks)
Month	36 (months)

Configuring Database Retention Policies

By default, the database maintenance process runs daily to purge data in accordance with database retention policies. The database retention policies are defined in the following options in the `dbmp` configuration section:

- On the Reporting Server, the `rs.db.retention.*.default` options—These set the default retention periods for the GVP deployment overall.
- On the IVR Profile, equivalent `rs.db.retention.*` options—These override the default retention periods, for data relating to the specific VoiceXML or CCXML application.

Table 35 summarizes the default Reporting Server database retention periods for data at the varying levels of granularity (aggregation periods).

Before you modify the default retention periods, consider your reporting requirements and the reporting results you expect. Ensure that your default database retention period settings are consistent with settings for the `reporting.rs.query.limit.<granularity>` configuration options, so that data that you expect to include in reports at various granularity levels is not purged prematurely from the database.

For information on using the Data Retention Policy Wizard to configure these policies, see "Data Retention Policy Wizard" on page 138.

Note: When using the default parameter, there is a possibility that one maintenance execution may be skipped when Day Light Savings occurs. This will cause missing data on the VAR reports.

Table 35: Default Database Retention Periods

Type of Data	Granularity	Option Name in dbmp Section	Minimum Valid Value (Integer)	Default Value, in Days
CDRs	N/A	rs.db.retention.cdr.default	> 0	30
Operational data	5-minute	rs.db.retention.operations.5min.default	> 0	1
	30-minutes	rs.db.retention.operations.30min.default	> 0	7
	Daily	rs.db.retention.operations.daily.default	> 30	90
	Hourly	rs.db.retention.operations.hourly.default	> 0	7
	Weekly	rs.db.retention.operations.weekly.default	> 30	364 (52 weeks)
	Monthly	rs.db.retention.operations.monthly.default	> 30	1095 (36 Months)
VAR summary statistics (Call Summary and IVR Action statistics)	5-minute	rs.db.retention.var.5min.default	> 0	1
	30-minutes	rs.db.retention.var.30min.default	> 0	7
	Daily	rs.db.retention.var.daily.default	> 30	90
	Hourly	rs.db.retention.var.hourly.default	> 0	7
	Monthly	rs.db.retention.var.monthly.default	> 30	1095 (36 months)
	Weekly	rs.db.retention.var.weekly.default	> 30	364 (52 weeks)
Service Quality Latency Summary data	Daily	rs.db.retention.latencies.daily.default	> 30	90
	Hourly	rs.db.retention.latencies.hourly.default	> 0	7
	Weekly	rs.db.retention.latencies.weekly.default	> 30	364 (52 weeks)
	Monthly	rs.db.retention.latencies.monthly.default	> 30	1095 (36 Months)
Service Quality Failures	N/A	rs.db.retention.sq.failures.default	>0	365

Table 35: Default Database Retention Periods (Continued)

Type of Data	Granularity	Option Name in dbmp Section	Minimum Valid Value (Integer)	Default Value, in Days
Service Quality Summary data	Daily	rs.db.retention.sq.daily.default	> 30	90
	Hourly	rs.db.retention.sq.hourly.default	> 0	7
	Weekly	rs.db.retention.sq.weekly.default	> 30	364 (52 weeks)
	Monthly	rs.db.retention.sq.monthly.default	> 30	1095 (36 Months)

Disabling CDR Storage for Resource Manager and Media Control Platform

You can use the Reporting Server configuration option `[cdr]media-service-cdrs.reduce` to disable storage of Resource Manager (RM) and Media Control Platform (MCP) CDRs that have certain media service types.

Set the option to `true` to disable storage of these CDRs in the remote database:

- Any RM CDR and MCP CDR with its media service type set to: `media`, `record` or `conference`.
- Any RM CDR or MCP CDR that has the media service type CPD and VXML resource flag set to false and media service type set to CPD.

Notes:
- RM and MCP must provide the correct media service type in the first CDR message for a given call session.
- A new VXML field for RM CDRs is set by RM, to satisfy the above requirement.
- See RS.OR.MS.x for more details about media service types.

Important Reporting Server Configuration Options

This section describes the key configuration options that you either must or may want to customize.

Configure the options in Genesys Administrator on the `Provisioning > Environment > Applications > <Reporting Server> > Options` tab. For the detailed steps to configure option settings, see Procedure: Viewing or modifying GVP configuration parameters, on page 34.

The configurable Reporting Server parameters are in the following configuration sections:

- agentx—Parameters that determine the behavior of the connection attempts and delays between the SNMP subagent and the SNMP Master Agent.
- cdr—Parameters determine behavior for processing and reporting on call detail records (CDRs).
- dbmp—Parameters determine database retention policies (see Table 35 on page 279).
- https (see "Controlling Access to Reporting Services" on page 287) — Parameters determine HTTPS settings.
- https_key—Parameters determine HTTPS key settings.
- imdb—Parameters determine database query behavior.
- latency—Parameters determine latency threshold behavior.
- log (see "Configuring Logging" on page 66)—Parameters determine logging behavior.
- messaging—Parameters specify the paths for the ActiveMQ JMS broker that receives Reporting Server messages.
- persistence—Parameters configure behavior for Hibernate interactions with the database.
- reporting—Parameters determine the number of records that will be considered for different levels of granularity.
- schedule—Parameters provide the cron expressions for scheduling tasks.
- sqa—Parameters determine service quality behavior.
- transaction—Parameters provide the directory paths for the Atomikos distributed transactions processing engine.

Table 36 provides information about important Reporting Server parameters that are not described in Chapter 3 on page 41. Table 36 provides parameter descriptions as well as the default parameter values that are preconfigured in the Reporting Server Application object.

Except for changes in the dbmp and log sections, all changes take effect on restart.

For information about all the available configuration options for the Reporting Server, see the *Genesys Voice Platform 8.1 Configuration Options Reference.*

Table 36: Selected Reporting Server Configuration Options

Option Name	Description	Valid Values and Syntax
	`agentx` **Section**	
Subagent Connection Attempts	Specifies the maximum connection attempts to be made by the SNMP subagent to the SNMP Master Agent. If the option value if not set or if the value is less than or equal to `0`, there is no limit on the number of attempts.	An integer greater than `0`. **Default value:** `0`
	`cdr` **Section**	
Call Timeout	The amount of time, in minutes, until a call is considered timed out from the perspective of VAR and CDR reporting. The Reporting Server may receive no CDR call-termination update because: • The call was dropped from the platform (for example, because a component shut down unexpectedly).	An integer in the range of 1-1440. **Default value:** `180` (3 hours)
Call Timeout (continued)	• The Reporting Server is simply not receiving updates from the component (for example, because the network connection is down). The component queues data that it cannot send to the Reporting Server, so the Reporting Server may eventually receive a CDR update for a call that was previously assumed to be timed out. In these cases, the Reporting Server will appropriately update the CDR. The interval at which the timeout process runs is configurable (see `schedule.quartz.rs.calltimeout` on page 287). The timeout process uses the value of the `call-timeout` parameter to identify calls that have timed out since the process last ran.	
Limit of Disk Storage for Messages Handled by the ActiveMQ Broker	Specifies a limit to the amount of disk storage that can be used for messages handled by the ActiveMQ broker.	String. **Default value:** `256 gb`
Max Page Count	The maximum number of pages that will be returned in any given CDR or Call Events report request.	An integer in the range of 1–100. **Default value:** `10`

Table 36: Selected Reporting Server Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
Max Page Size	The maximum number of records that will be returned in a single page in any given CDR or Call Events report request. This limit prevents users from overloading the system by requesting unreasonably large numbers of CDRs or metrics in a report.	An integer in the range of 1-10000. **Default value:** `100`
`imdb` Section		
Max Concurrent CDR Queries	The maximum number of concurrently executed CDR in-progress queries.	An integer in the range of 1–15. **Default value:** `3`
Max Query Lock Timeout	The maximum time, in milliseconds, for the real-time query to wait before locking the in-memory storage.	An integer in the range of 100–5000. **Default value:** `1000`
`latency` Section		
Threshold Criteria for <given> Latency	Specifies the latency threshold, in milliseconds, and the percentile for the given latency. For every Service Quality period, the Report Server calculates the actual latency for the specified percentile. It that number exceeds the threshold, an error is logged.	`<Threshold|Percentile>` **Default values:** • `Page Fetch—1500|95` • `Audio Fetch—1000|95` • `Grammar Fetch—1000|95` • `Day Fetch—2000|95` • `JavaScript Fetch—1000|95` • `Page Compile—100|95` • `JavaScript Execution—50|99` • `Initial Response—4000|95` • `Call Answer—2000|95` • `Call Reject—2000|95` • `First Prompt Inbound—2000|95`

Table 36: Selected Reporting Server Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
		• First Prompt Outbound—2000|95 • Inter Prompt—2000|95 • Cumulative Response—2000|95 • DTMF Prompt—2000|95 • ASR Input—2000|95 • No Input Response—2000|95 · Recording Response· 2000|95 • Transfer Response—2000|95 • MRCP ASR Session Establish—100|95 • MRCP TTS Session Establish—100|95 • MRCP ASR Set Params—100|95
Threshold Criteria for <given> Latency (continued)		• MRCP ASR Stop—100|95 • MRCP Define Grammar—500|95 • MRCP Recognize—500|95 • MRCP Speak—100|95 • MRCP TTS Set Params· 100|95 • MRCP TTS Stop—100|95

Table 36: Selected Reporting Server Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
	`messaging` **Section**	
ActiveMQ Broker Connectors	Specifies the type of connector that will be enabled on the ActiveMQ broker.	An integer equal to 0, 1, or 2. • `0` = Unencrypted connections only. • `1` = SSL connections only. (Not supported for GVP 8.1.3 or earlier clients.) • `2` = SSL-enabled for GVP 8.1.4 and earlier clients, that will connect in unencrypted mode. (GVP 8.1.4 and later clients will connect in encrypted mod.) **Default value:** `0`
ActiveMQ JMS Broker Port for SSL	Specifies the SSL listening port for the ActiveMQ JMS broker that receives incoming data from Reporting Clients.	An integer greater than 0. **Default value:** `61617`
ActiveMQ Keystore for SSL Private Key and Certificate	Specifies the path to the Java Keystore file that contain the cryptographic key and trusted certificate entries that ActiveMQ broker requires to provide TLS/SSL support.	A string of characters. **Default value:** `keystore.ks`
ActiveMQ Keystore Password	Specifies the password that is required to open the keystore used by the ActiveMQ broker.	A string of characters. **Default value:** `""`
Local Listening Address for the ActiveMQ's Broker	Specifies the IP address for the listening port that is used by the ActiveMQ broker (an unencrypted connector).	A string of characters. **Default value:** `""`
Local Listening Address for the ActiveMQ's Broker (TLS)	Specifies the IP address for the TLS (encrypted) listening port that is used by the ActiveMQ broker (an unencrypted connector).	A string of characters. **Default value:** `""`

Table 36: Selected Reporting Server Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
persistence Section		
Read-Only Mode	Specifies read-only mode for the Reporting Server. The Reporting Server does not write to the remote database, but continues to support report queries. • If this option value is true, read-only mode is enabled. • If this option value is false, standard read-write mode is enabled	• true • false **Default value:** false
reporting Section		
Maximum configured units of <time period>	The maximum number of <granularity> periods that are included in any report with a granularity of <granularity>, where <granularity> is: • 5min—5-minute periods • 30min • day • hour • month • week If a reporting request at a particular granularity level specifies a time range that is greater than the configured maximum, the request is truncated to cover the maximum allowed time period, starting from the From time specified in the request.	See Table 35 on page 279. **Example:** If rs.query.limit.5min=288 (1 day), a request for a report at 5-minute granularity for the time period 2008/01/01 00:00 – 2008/01/02 12:00 will be truncated to 2008/01/01 00:00 – 2008/01/02 00:00.
Summarization Buffer Time	Specifies the buffer time, in minutes, that ensures that the summarization process runs after this time has elapsed. Records will be summarized before this time.	An integer range from 0–44640. **Default value:** 60
The maximum allowed database query running time (in seconds) before the RS sends cancel query request to database server	Specifies the maximum time, in seconds, for Reporting Server to query the database before cancelling the request.	Any integer range from 0–65535. **Default value:** 60

Table 36: Selected Reporting Server Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
	`schedule` **Section**	
Call Timeout Process	The `cron` schedule for Quartz to execute the Call Timeout Process, which is responsible for timing out Resource Manager, Media Control Platform, Call Control Platform, and VAR CDRs, so that they do not get stuck as open calls in the database. By default, the process runs every 50 minutes. A configurable option specifies the timeout interval that determines when a call is considered timed out (see the `cdr.call-timeout` option).	**Default value:** `0 50 * * * ?`
RS DB Maintenance Process	The `cron` schedule for Quartz to purge data from the database, in accordance with data retention policies. By default, the process runs at 1 a.m. every day.	**Default value:** `0 0 1 * * ?`
	`sqa` **Section**	
Minimum Calls for Service Quality	Specifies the minimum number of calls that need to be recorded before the service quality alarm is issued at the critical level.	Any integer value. **Default value:** `100`
Minimum Latency Measurements for Threshold Warning	Specifies the minimum number of latency measurements that need to be recorded before a warning is logged.	Any integer value. **Default value:** `100`

Controlling Access to Reporting Services

GVP 8.1 leverages the HTTPS features to secure users access to Reporting Services.

GVP 8.1 does not support selective, role-based access for different categories of Reporting services.

The following procedure describes how to configure Reporting Server to use HTTPS.

Procedure:
Enabling HTTPS for Reporting

Purpose: To enable HTTPS for Reporting.

Start of procedure

1. Obtain a server certificate that is signed by a third-party authority (for example, CAcert, Comodo, or VeriSign).
2. Install the server certificate.

   Use the PKCS12Import utility to import the server certificate into the Jetty keystore with the following command:

   ```
   java -classpath <jar> org.mortbay.jetty.security.PKCS12Import
   <source> <keystore>
   ```

   Where:

   <jar> is the path to the ems-rs.jar file.

   <source> is the path to the PKCS12 file that contains the keys and certificates.

   <keystore> is the path to the keystone file where the keys and certificates are installed.

   For example,

   ```
   java -classpath ems-rs.jar
   org.mortbay.jetty.security.PKCS12Import rs_example_com.pkcs12
   keystore.jks
   Enter input keystore passphrase: secret123
   Enter output keystore passphrase: secret123
   Alias 0: 1
   Adding key for alias 1
   ```

3. Configure the Reporting Server application.
   a. In Genesys Administrator, go to the Provisioning > Environment > Applications > <Reporting Server> > Options tab.
   b. Under the reporting section, add the following parameters:

      hostname = the FQDN of the host to which the server certificate is assigned

      protocol = https
   c. Under the https section, modify the following parameters:

      https.keystore.path = the path to the keystore file.

      https.protocol = SSL.

      password = the keystore password.
   d. Under the https_key section, set the password parameter to the keystore password.
   e. Click Save.
4. Restart Reporting Server.

5. To test HTTPS authentication, navigate to `https://<FQDN>:8080/ems-rs/components`, where `<FQDN>` is the fully qualified domain name of the host to which the server certificate is assigned.

   Reporting Server will now service all web service requests through HTTPS, including those from the Management Framework Reporting UI (Genesys Administrator).

6. Use the Trusted Root Certification Authorities MMC snap-in to verify that the certificate is trusted by Windows.

7. Verify that GVP reports are properly displayed in Genesys Administrator when you request them from the `Monitoring > Voice Platform` navigation panel.

End of procedure

Chapter

15 Configuring GVP in Multi-Site Environments

This chapter describes the requirements to configure Genesys Voice Platform (GVP) to support multi-site environments.

It contains the following sections:

Overview

GVP supports multi-site configurations in large scale environments. Typically, a single site is represented by a Resource Manager (RM) instance (or a pair of redundant RM instances), a Reporting Server (RS) instance (or a pair of redundant RS instances), and a pool of Media Control Platform (MCP) instances. Within a single site, scalability is limited by the number of call attempts-per-second (CAPS) supported by the Reporting Server.

GVP scalability has gone beyond a single site and is now scalable across multiple physical sites. The Resource Manager can facilitate resource-sharing between sites and consistently enforce usage policies across all sites. The Reporting Server can generate historical and real-time reports that are filtered to produce site reporting or system-wide reporting. In addition, SIP Server instances within the same site can use all of the available resources within the site.

Site Identification

In GVP multi-site environments, sites are identified by a *folder* object in Genesys Management Framework. The folder is created and configured in Genesys Administrator on the `Provisioning` tab. For example, `Provisioning` >

`Environment > Application > Site folder`. The `Options` tab in the configuration properties of the `Site folder`, the `Advanced View (Annex)` contains a configuration section called `gvp.site`. You can add various options to this section to control resource-sharing, geo-location, and the types and weight (or number) of calls that can be routed to this site. See Table 37.

Management Framework obtains the site name from the site folder within the `Applications` folder. The DBID of the site folder is used as the site ID, so it must be unique.

The only GVP `Applications` in the site folder are the Resource Manager and Reporting Server.

Configuring the Site Folder

Configure the `Site` folder in Genesys Administrator, by using the options described in Table 37 and the Procedure: Configuring a Site folder by using Genesys Administrator.

Table 37: Site Folder Configuration Options

Configuration option	Option description	Valid values
contact (mandatory)	Specifies the SIP route address for this site. This configuration option is used for call forwarding and site monitoring.	A list of IP addresses and port numbers.
geo-location	Specifies a list of geographic locations that are supported by this site.	A comma-separated list.
resource-sharing	Specifies whether or not resource sharing is enabled for this site.	True or False
weight	Specifies the relative weight that is allocated to this site.	Any unsigned integer.

Procedure:
Configuring a Site folder by using Genesys Administrator

Purpose: To configure a Site folder to support multi-site environments.

Start of procedure

1. In Genesys Administrator, go to `Provisioning > Environment > Applications`.
2. In the task bar, click `New Folder`.

   The `Folder name` dialog box appears.
3. Enter the site folder name and click `OK`.
4. Right-click on the folder and select `Edit`.

   The `Configuration` tab appears.
5. On the `Options` tab, select `New`.
6. In the `Section` field enter, gvp.site.
7. In the `Name` field:
   a. Enter `contact`. This is the SIP route address in the format: `10.10.10:5060`

      Which represents either the virtual IP and Resource Manager proxy port (when RM is clustered) or the network interface that RM binds to and the RM proxy port (when RM is standalone). (This configuration option is mandatory.)
   b. Enter the following additional options, if required:
      — `weight`—Enter a weight. (If a value is not specified, default = `100.`)
      — `geo-location`—Enter a location. (Optional, can be left blank.)
      — `resource-sharing`—Enter `true` or `false`. (If a value is not specified, default = `true`).
8. In the `Value` field, enter an appropriate value.
9. To add additional options, click `Save & New`.
10. To move an existing Resource Manager or Reporting Server `Application` into the folder:
    - Highlight the `Application`.
    - In the task bar, select `Move to`.
    - In the `Browse` window, select the Site folder you created in Step 2.
11. In the `Confirm` dialog box, click `Yes`.

End of procedure

Next Steps

- No further steps are required.

Genesys

Part

2 Monitoring GVP

This part of the Guide describes the available real-time and historical reports in Genesys Administrator.

This information appears in the following chapters:

Chapter

16 Reporting Overview

This chapter describes how to use Genesys Administrator to create real-time and historical reports. It contains the following sections:

Reports—Using GA vs. Using GAX

Genesys Administrator Extension (GAX) can now generate all reports that are available in Genesys Administrator (GA), and some new reports that GA does not offer. GVP 8.1.7 configuration, as well as the ability to generate most reports, remains in Genesys Administrator (GA).

Below is a breakdown of reports that you can generate with GAX vs. with GA.

Functionality Exclusive to GAX

Generating these new reports:

- VoiceXML Call Arrivals, VoiceXML Call Peaks, Media Service Call Arrivals, Media Service Call Peaks.
- Call Durations for VoiceXML, Media Service, and ASR/TTS.

Functionality Exclusive to GA

- Configuring GVP.

Functionality Common to both GAX and GA

- Generating Call Browser Reports: Historical Call Status, In Progress Call Status.
- Generating VAR reports: VAR Call Completion, VAR IVR Action Usage, VAR Last IVR Action.
- Generating these Operational and Dashboard reports:
  - Real-time Call Browser, IVR Profile Call Arrivals, IVR Profile Call Peaks, Tenant Call Arrivals, Tenant Call Peaks.
  - Component Call Arrivals (RM, MCP, CCP, PSTNC, CTIC, ASR, TTS).
  - Component Call Peaks (RM, MCP, CCP, PSTNC, CTIC, ASR, TTS).
  - Call Dashboard, SSG Dashboard, Fetch Dashboard, PSTNC, CTIC Dashboard.
- Generating Service Quality Reports: CallFailures, Call Summary, Latency Details, Latency Dashboard.

Genesys Administrator is the legacy tool that you use to monitor your call-center activity; it enables you to analyze call volumes, trends, and the effectiveness of your voice and call-control applications.

For more information on how to use Genesys Administrator, see the *Framework 8.1 Genesys Administrator Help* file.

Warning! The tenant that is defined as the *parent* becomes the reference entry point in the tenant hierarchy. The parent tenant with read permissions can view their child tenants and their configurations and reports, but cannot view the child tenants below them (their *grandchild* tenants).

Generating a Report with GA

The following procedures explain how to generate a report using Genesys Administrator or Genesys Administrator Extension.

Procedure: Generating a Report Using Genesys Administrator

Purpose: To generate a report by using Genesys Administrator.

Prerequisites

- The valid URL for Genesys Administrator—for example, `http://<Genesys Administrator host>/wcm/`.

- The username and password with the correct permissions for running reports.
- The name of Genesys Administrator application—for example, `default`.

Note: The `default` application object is automatically created, and is seen when Genesys Administrator is invoked.

- The host name and port of the Genesys Configuration Server.

Start of procedure:

1. In the web browser's address bar, enter `http://<Genesys Administrator host>/wcm/`.

   The login to `Genesys Administrator` dialog box appears.
2. Enter the following parameters:
   - `User Name`
   - `Password`
   - `Application`
   - `Host Name`
   - `Port`
3. Click `Login`.

   The `Genesys Administrator` screen appears (see Figure 8) with the `Monitoring` tab active.

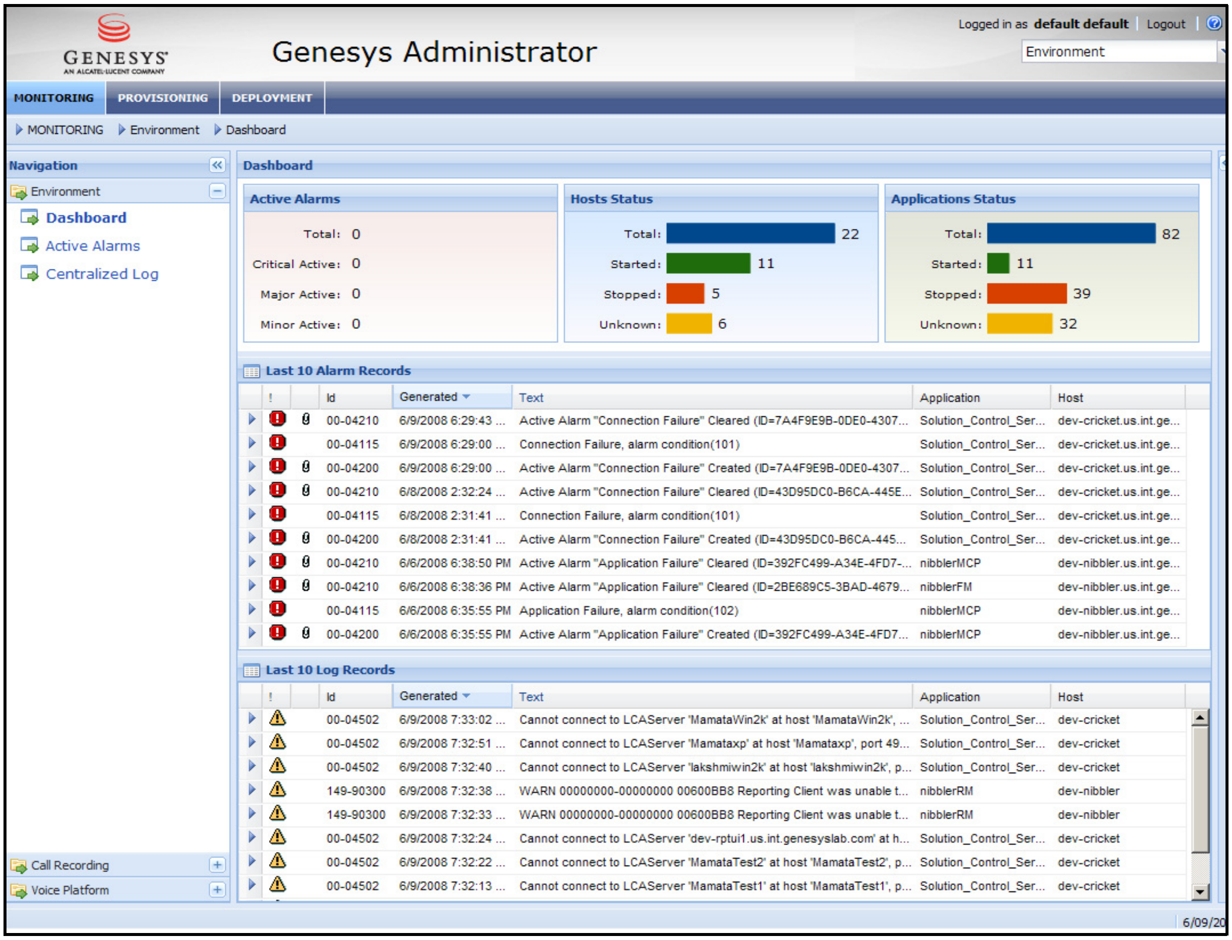


Figure 8: Genesys Administrator

4. From the Navigation panel, select `Voice Platform`.

   The reporting categories are visible (see Figure 9).

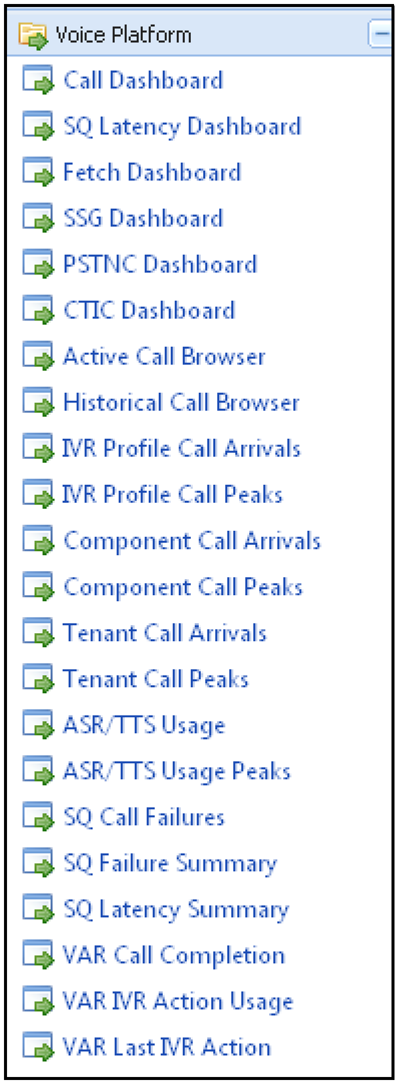


Figure 9: Genesys Administrator Voice Platform

5. Select the required report.
6. The filter screen appears with the possible filter criteria (see Figure 10). A different set of filters appear for each report type.

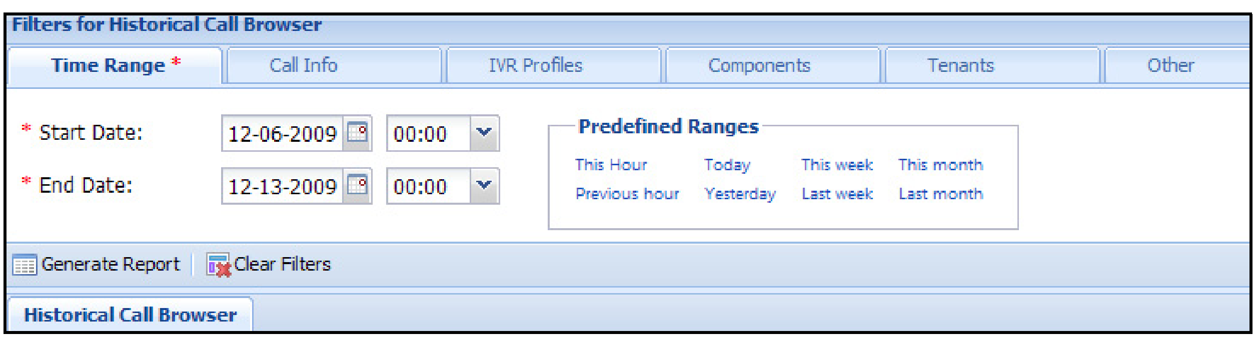


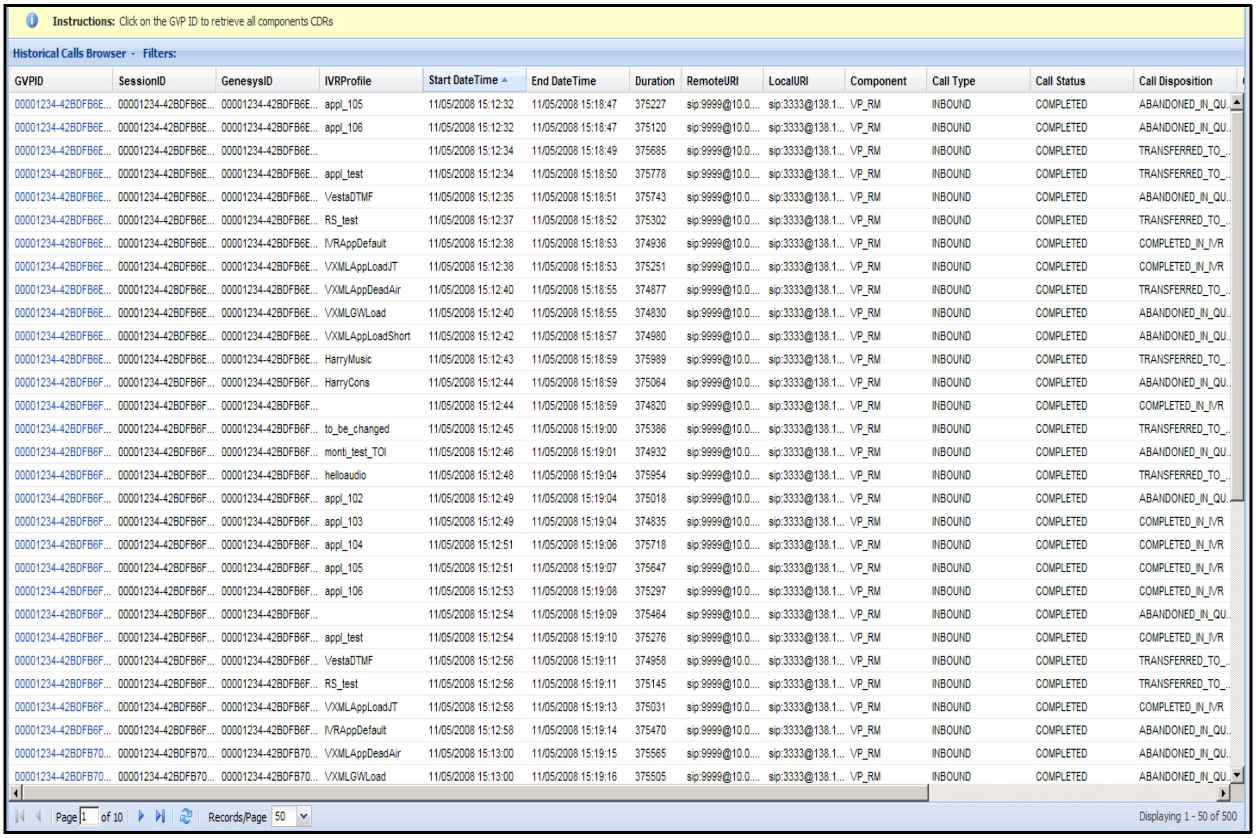
Figure 10: Configure Filters for the Historical Call Browser Screen

7. Select the required filters.

   For information on how to select the Active Call Browser filters, see Procedure: Generating the Active Call Browser Report with GA, on page 334.

8. Click `Generate Report`.

   The `Report Results` screen appears (see Figure 11).



Instructions: Click on the GVP ID to retrieve all components CDRs

Historical Calls Browser - Filters:

GVPID	SessionID	GenesysID	IVRProfile	Start DateTime	End DateTime	Duration	RemoteURI	LocalURI	Component	Call Type	Call Status	Call Disposition
00001234-42BDFB6E...	00001234-42BDFB6E...	00001234-42BDFB6E...	appl_105	11/05/2008 15:12:32	11/05/2008 15:18:47	375227	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	ABANDONED_IN_QU...
00001234-42BDFB6E...	00001234-42BDFB6E...	00001234-42BDFB6E...	appl_106	11/05/2008 15:12:32	11/05/2008 15:18:47	375120	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	ABANDONED_IN_QU...
00001234-42BDFB6E...	00001234-42BDFB6E...	00001234-42BDFB6E...		11/05/2008 15:12:34	11/05/2008 15:18:49	375685	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	TRANSFERRED_TO_...
00001234-42BDFB6E...	00001234-42BDFB6E...	00001234-42BDFB6E...	appl_test	11/05/2008 15:12:34	11/05/2008 15:18:50	375778	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	TRANSFERRED_TO_...
00001234-42BDFB6E...	00001234-42BDFB6E...	00001234-42BDFB6E...	VestaDTMF	11/05/2008 15:12:35	11/05/2008 15:18:51	375743	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	ABANDONED_IN_QU...
00001234-42BDFB6E...	00001234-42BDFB6E...	00001234-42BDFB6E...	RS_test	11/05/2008 15:12:37	11/05/2008 15:18:52	375302	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	TRANSFERRED_TO_...
00001234-42BDFB6E...	00001234-42BDFB6E...	00001234-42BDFB6E...	IVRAppDefault	11/05/2008 15:12:38	11/05/2008 15:18:53	374936	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	COMPLETED_IN_IVR
00001234-42BDFB6E...	00001234-42BDFB6E...	00001234-42BDFB6E...	VXMLAppLoadJT	11/05/2008 15:12:38	11/05/2008 15:18:53	375251	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	COMPLETED_IN_IVR
00001234-42BDFB6E...	00001234-42BDFB6E...	00001234-42BDFB6E...	VXMLAppDeadAir	11/05/2008 15:12:40	11/05/2008 15:18:55	374877	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	TRANSFERRED_TO_...
00001234-42BDFB6E...	00001234-42BDFB6E...	00001234-42BDFB6E...	VXMLGWLoad	11/05/2008 15:12:40	11/05/2008 15:18:55	374830	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	ABANDONED_IN_QU...
00001234-42BDFB6E...	00001234-42BDFB6E...	00001234-42BDFB6E...	VXMLAppLoadShort	11/05/2008 15:12:42	11/05/2008 15:18:57	374980	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	ABANDONED_IN_QU...
00001234-42BDFB6E...	00001234-42BDFB6E...	00001234-42BDFB6E...	HarryMusic	11/05/2008 15:12:43	11/05/2008 15:18:59	375989	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	TRANSFERRED_TO_...
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...	HarryCons	11/05/2008 15:12:44	11/05/2008 15:18:59	375064	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	ABANDONED_IN_QU...
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...		11/05/2008 15:12:44	11/05/2008 15:18:59	374820	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	COMPLETED_IN_IVR
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...	to_be_changed	11/05/2008 15:12:45	11/05/2008 15:19:00	375386	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	TRANSFERRED_TO_...
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...	monti_test_TOI	11/05/2008 15:12:46	11/05/2008 15:19:01	374932	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	ABANDONED_IN_QU...
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...	helloaudio	11/05/2008 15:12:48	11/05/2008 15:19:04	375954	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	TRANSFERRED_TO_...
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...	appl_102	11/05/2008 15:12:49	11/05/2008 15:19:04	375018	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	ABANDONED_IN_QU...
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...	appl_103	11/05/2008 15:12:49	11/05/2008 15:19:04	374835	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	COMPLETED_IN_IVR
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...	appl_104	11/05/2008 15:12:51	11/05/2008 15:19:06	375718	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	COMPLETED_IN_IVR
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...	appl_105	11/05/2008 15:12:51	11/05/2008 15:19:07	375647	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	COMPLETED_IN_IVR
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...	appl_106	11/05/2008 15:12:53	11/05/2008 15:19:08	375297	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	COMPLETED_IN_IVR
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...		11/05/2008 15:12:54	11/05/2008 15:19:09	375464	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	ABANDONED_IN_QU...
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...	appl_test	11/05/2008 15:12:54	11/05/2008 15:19:10	375276	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	COMPLETED_IN_IVR
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...	VestaDTMF	11/05/2008 15:12:56	11/05/2008 15:19:11	374958	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	TRANSFERRED_TO_...
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...	RS_test	11/05/2008 15:12:56	11/05/2008 15:19:11	375145	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	TRANSFERRED_TO_...
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...	VXMLAppLoadJT	11/05/2008 15:12:58	11/05/2008 15:19:13	375031	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	COMPLETED_IN_IVR
00001234-42BDFB6F...	00001234-42BDFB6F...	00001234-42BDFB6F...	IVRAppDefault	11/05/2008 15:12:58	11/05/2008 15:19:14	375470	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	ABANDONED_IN_QU...
00001234-42BDFB70...	00001234-42BDFB70...	00001234-42BDFB70...	VXMLAppDeadAir	11/05/2008 15:13:00	11/05/2008 15:19:15	375565	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	ABANDONED_IN_QU...
00001234-42BDFB70...	00001234-42BDFB70...	00001234-42BDFB70...	VXMLGWLoad	11/05/2008 15:13:00	11/05/2008 15:19:16	375505	sip:9999@10.0....	sip:3333@138.1...	VP_RM	INBOUND	COMPLETED	ABANDONED_IN_QU...

Page 1 of 10 Records/Page 50 Displaying 1 - 50 of 500

Figure 11: Example of Historical Call Browser Report Results

9. Click `Clear Filters` if you want to erase the existing filters.

End of procedure

Generating a Report with GAX

Genesys Administrator Extension, part of the Genesys Framework, is a web-based graphical user interface (GUI) that provides advanced administrative and operational functionality targeted primarily providing access to GVP reporting and Operation Parameter Management.

Procedure:
Generating a report using GAX

Purpose: To generate a report by using Genesys Administrator Extension (GAX).

Prerequisites

- The valid URL for GAX—for example, http://<GAX_host>:<port>/gax/
- The username and password with the correct permissions for running reports.

Start of procedure:

1. In the web browser's address bar, enter http://<GAX_host>:<port>/gax/.

   The GAX login dialog box appears.
2. Complete the following fields:
   - User Name
   - Password
3. Click `Login`.

   The GAX home screen appears.
4. In the Navigation panel across the top of the home screen, select `Reports`.

   The VP Reporting drop-down menu appears.
5. Select the required report: the list Call Browser, Dashboard, Operational Report, Service Quality Report, VAR Report.
6. The report screen appears; the filter criteria occupy the left third. A different set of filters appears for each report type.
7. Select the appropriate filters. Required filters have a red asterisk(*) next to their name.

   Filters for each of the four report types are described in the online help:
   - Call Browser Report
   - Dashboard Report
   - Operation Report
   - Service Quality Report
   - VAR Report

   Filters may appear in these formats:
   - Radio button list (examples: Call Status, Report Type)
   - Check box list (examples: Query Data From, Media Control Platform Components, IVR Profiles)
   - Drop-down list (examples: Component Type, Call Type)

- Text entry field (examples: DIDs, Remote URI)
- Pop-up calendar / text entry field (examples: From Date, To Date)

8. Click `Generate`.

The `Report Results` window appears and occupies the right two-thirds of the screen. See the table GAX Report Display Controls in the online help.

End of procedure

Next Steps

- Which choices generate a specific report? See “GAX Report Generation Table” on page 304.
- What is in each report? See Chapter 19, “Historical Reports,” on page 341.

The following Reporting descriptions appear in GAX online help:

- Overview.
- Applying report filters for each report type (see also: “Report Filters” on page 310).
- Reading the reports.

GAX Report Generation Table

The table “To Generate a Report, You Must Make These Selections” lists the menu and filter choices that you must make to generate each report. These choices are not the only choices available to you—they are the specific choices that you *must* make to generate that particular report.

Other filter choices—such as Query Data From*, From Date*,To Date*, Granularity, Session Type, and others—help you refine the data that each report presents. Some are mandatory(*). These filters are common to multiple reports; they are described in the GAX online help and in the table “Report Filters” on page 310.

Table 38: To Generate a Report, You Must Make These Selections

Report to generate selection	Reports Menu selection	Report Type selection	Filter Type selection	Item selection
Real-time Call Browser	Call Browser	In Progress	—	—
Historical Call Browser	Call Browser	Historical	—	—

Table 38: To Generate a Report, You Must Make These Selections (Continued)

Report to generate selection	Reports Menu selection	Report Type selection	Filter Type selection	Item selection
Call Dashboard	Dashboard	Call Dashboard	—	—
SSG Dashboard	Dashboard	SSG Dashboard	—	—
Fetch Dashboard	Dashboard	Fetch Dashboard	—	—
Connector Dashboard	Dashboard	Connector Dashboard	—	—
SQ Latency Dashboard	Dashboard	SQ Latency Dashboard	—	—
IVR Profile Call Arrivals	Operational Report	Call Arrivals	IVR Profile	Select IVR Profile(s)
IVR Profile Call Peaks	Operational Report	Call Peaks	IVR profile	Select IVR Profile(s)
***Component* Call Arrivals**	Operational Report	Call Arrivals	Component	Select one: RM, MCP, VXML, Media Service, CCP, PSTNC, CTIC
ASR / TTS Call Arrivals	Operational Report	Call Arrivals	Component	Select one: ASR, TTS
***Component* Call Peaks**	Operational Report	Call Peaks	Component	Select one: RM, MCP, VXML, Media Service, CCP, PSTNC, CTIC
ASR / TTS Call Peaks	Operational Report	Call Arrivals	Component	Select one: ASR, TTS
Tenant Call Arrivals	Operational Report	Call Arrivals	Tenant	Select Tenant(s)
Tenant Call Peaks	Operational Report	Call Peaks	Tenant	Select Tenant(s)
IVR Profile Call Durations	Operational Report	Call Durations	IVR Profile	Select IVR Profile(s)
Component Call Durations	Operational Report	Call Durations	Component	Select Media Control Platform (MCP) Component(s)

Table 38: To Generate a Report, You Must Make These Selections (Continued)

Report to generate selection	Reports Menu selection	Report Type selection	Filter Type selection	Item selection
Tenant Call Durations	Operational Report	Call Durations	Tenant	Select Tenant(s)
SQ Call Failures	Service Quality (SQ) Report	Call Failures	—	—
Call Summary	SQ Report	Call Summary	—	—
Latency Details	SQ Report	Latency Details	—	—
VAR Call Completion	VAR Report	Call Completion	—	—
VAR IVR Action Usage	VAR Report	IVR Action Usage	—	—
VAR Last IVR Action	VAR Report	Last IVR Action	—	—

Report Groups

Reports that you can generate using the GAX-GVP Reporting Plugin fall into these categories:

Call Detail Record Browsing

Generate real-time and historical reports of status for calls processed by different components in the GVP deployment.

Dashboard

Generate real-time reports that monitor in-progress calls from the perspective of IVR Profiles or GVP components.

Operational Reporting

Generate reports on the rate of call arrivals, call durations, and peak call volume by IVR Profile or GVP component.

Call Duration Operational Reports for MCP VXML and Media Service

The Operational Report type Call Duration generates total call duration data for specific time periods.

The total call duration for a given hour is the sum of time spent by every call in that hour. A call that spans multiple time periods contributes the appropriate percentage of its duration to each of the time periods. For example, a 3-hour call that starts at 3:45 spans four different hour periods, and thus contributes—proportionally—to four call duration records.

Report granularities are HOUR, DAY, WEEK and MONTH.

MCP VXML Duration

MCP VXML Duration reports generate historical call duration summaries for MCP calls that have the VXML resource usage flag set to `used`. These reports contain:

- Total duration of MCP VXML calls per tenant-id.
- Total duration of MCP VXML calls per application-id
- Total duration of MCP VXML calls per component-id
- Total duration of MCP VXML calls for the entire site (single RS deployment).

Media Service Duration

Media Service Duration reports generate historical call duration summaries for call sessions that use media services (sessions that have the media-service CDR parameter set) These reports contain:

- Total duration of MEDIA RM calls per tenant-id and for the entire RS site.
- Total duration of CPD RM calls per tenant-id and for the entire RS site.
- Total duration of RECORDING RM calls per tenant-id and for the entire RS site.
- Total duration of CONFERENCE RM calls per tenant-id and for the entire RS site.
- Total duration of TREATMENT MCP (MCP sessions with TREATMENT and with no VXML) calls per tenant-id and for the entire RS site.

ASR Call Duration

ASR Call Duration reports generate historical call duration summaries for MCP calls that have the ASR resource usage flag (see RS.MCP.CDR.2) set to used. These reports contain:

- Total call duration of MCP calls that used an ASR resource for a given IVR application-id.
- Total call duration of MCP calls that used an ASR resource for a given tenant-id.
- Total call duration of MCP calls that used an ASR resource for a given component-id.
- Total call duration of MCP calls that used an ASR resource for the entire site (single RS deployment).

TTS Call Duration

TTS Call Duration Reporting provides historical call duration summaries for MCP calls that have the TTS resource usage flag (see RS.MCP.CDR.2) set to `used`. These reports contain:

- Total call duration of MCP calls that used an TTS resource for a given IVR application-id.
- Total call duration of MCP calls that used an TTS resource for a given tenant-id.
- Total call duration of MCP calls that used an TTS resource for a given MCP (component-id).
- Total call duration of MCP calls that used an TTS resource for the entire site (single RS deployment).

Service Quality Reporting

Generate service quality reports of call failures, call summaries, and latency details for MCP components, from call analysis data provided by the Reporting Server.

Voice Application Reporting

Generate reports on the logical success and failure rates for calls and IVR Actions in a given IVR Profile.

Note: VAR reporting data is available only for applications that leverage the VAR <log> interfaces described in the Reporting Server Functional Specification.

Report Filters

Table 39 describes the filter criteria that you can use to retrieve call detail records, IVR action data, or summary data.

Table 39: Report Filters

Filter Name	Description
Time Range	Filters the data by start date, start time, end date, and end time. The results will display calls that started on or after the start time and ended before the end time. **Note:** Selecting from the `Predefined Ranges` automatically populates the `Start` and `End` dates with common time ranges. • `This Hour` (granularity = five minutes) • `Today` (granularity = hour) • `Yesterday` (granularity = hour) • `This Week` (granularity = day) • `Last Week` (granularity = day) • `This Month` (granularity = day) • `Last Month` (granularity = day) The following Time Ranges are for the Active Call Browser report only. • `Last Five Minutes` • `Last Fifteen Minutes` • `Last Thirty Minutes` • `Last Hour` • `Last Day` **Default:** `Today`.
Granularity	Presents the data at various levels of aggregation: • `Five Minutes` • `Thirty Minutes` • `Hour` • `Day` • `Week` • `Month` • `Six week moving average` **Default:** `Hour`

Table 39: Report Filters (Continued)

Filter Name	Description
IVR Profile	Filters the data by IVR Profile. You can choose more than one IVR Profile for some of the reports. For more information on IVR Profiles, see Chapter 6 on page 107.
Component Type	Filters the data by Component Type. The possible Components are: • `RM`—Resource Manager • `MCP`—Media Control Platform • `CCP`—Call Control Platform • `CTIC`—CTI Connector • `PSTNC`—PSTN Connector
Component	Filters the data by the component. A component is a provisioned Resource Manager, MCP, CCP, CTIC, or PSTNC application. You can choose more than one component for some of the reports. **Note:** All selected components must be of the same type.
Tenant	Filters the data by Tenant.
DID	Filters the data by DID.
Call Type	Filters the data by call type. The possible call types are: • `Inbound`—Applicable for MCP and RM components. • `Outbound`—Applicable for MCP and RM components. • `Bridged`—Applicable for MCP components only. • `New Call`—Applicable for CCP components only. • `Createccxml`—Applicable for CCP components only. • `External`—Applicable for CCP components only. • `Unknown`—Applicable for RM components only.
Call Length	Filters the data by the length of time, in milliseconds, of the call. Minimum and maximum durations can be specified. If only a minimum duration is specified, calls that exceeded this duration are displayed. If only a maximum duration is specified, calls that lasted for less than or equal to this duration are displayed.

Table 39: Report Filters (Continued)

Filter Name	Description
Call Failure Type	Filters the data by the reason for the failed call. The possible call failure types are: • Call answer • Call reject • Inbound first prompt latency • Outbound first prompt latency • Inter-prompt latency • Cumulative response latency • Audio gap latency • Application error • System error • Unknown Failure
ID	Filters the data by the call ID. The possible IDs are: • `Session ID`—The GVP Component specific ID that is generated by the component to identify the call leg. • `GVP GUID`—The globally unique ID that identifies a complete interaction with GVP. This ID is generated by the Resource Manager, and is passed to all the resources that provide service for the call. • `Genesys UUID`—The Genesys CallUUID that is generated by T-Server or SIP Server. For more information on these IDs, see Chapter 1, "Introduction," on page 21.
Call State	Filters the data by Call States. The possible Call States are: • `Accepted`—The call has been received by Resource Manager, but has not yet landed on a VoiceXML platform, or been transferred to an agent. • `IVR`—The call has landed on a VoiceXML platform. • `Transferring`—The call is being transferred to an agent. • `Transferred`—The call was successfully completed.

Table 39: Report Filters (Continued)

Filter Name	Description
Call Disposition	Filters the data according to the outcome of the call. The possible call dispositions are: • `Completed in IVR`—The call completed in self service. • `Abandoned in Queue`—The caller hung up while waiting in the queue. This is available only for those call flows that include IVR Server for CTI. • `Transferred to Agent`—The call was send to an agent. • `Rejected`—The call cannot be routed to a media resource.
Call End State	Filters the data by Call End State. The possible Call End States are: • `Application End`—The voice application hung up. • `System Error`—The call did not end properly. • `Unknown`—The platform did not log an end state. • `User End`—The caller hung up.
Call Result	Filters the date by Call Results. The possible Call Results are: • `Success`—The call was processed successfully. • `Failed`—A failure occurred that prevented the call from being processed properly • `Rejected`—The platform rejected the call. • `Unknown`—Some unknown reason that the call ended abruptly.
Virtual Reporting Object	Filters the data by the user defined Virtual Reporting Object.
Remote URI	Filters the data by the full URI of the remote party that is involved in the session. **Note:** Accepts the `*` wildcard.
Local URI	Filters the data by the URI of the local service. **Note:** Accepts the `*` wildcard.

Note: Certain filters are only available when a particular component is selected. For example, `Call End State` is displayed only when MCP is selected.

The data on the reports that use the granularity filter are stored in the database for the length of time that is given for the dbmp.rs.db.retention.cdr.default parameter. Granularity works with the data reporting limits that are configured in the Reporting Server. These limits are the maximum amount of data that the Reporting Server returns based on the which granularity level is selected. The Report Server options are:

- rs.query.limit.5mins
- re.query.limit.30mins
- rs.query.limit.hour
- rs.query.limit.day
- rs.query.limit.week
- rs.query.limit.month

For more information, see “Configuring Reporting, by Granularity” on page 277.

Genesys

Chapter

17 Voice Platform Dashboards

This chapter describes the Genesys Voice Platform (GVP) Dashboards. The Voice Platform Dashboards display real-time and historical information for selected IVR Profiles, components, and tenants. It contains the following sections:

Overview

The Voice Platform Dashboards display a high-level summary of the current usage for IVR Profiles and Resource Manager (RM), Media Control Platform (MCP), Call Control Platform (CCP), Supplementary Services Gateway (SSG), and PSTN Connector (PSTNC) components. Each dashboard can be configured to auto-update its display at regular intervals. The data that is displayed represents current values. How current the data is depends on two factors:

- In-progress session counts are derived from CDRs. There can be delays in the delivery of CDRs to Reporting Server. The dashboard reflects the CDRs that are currently available to Reporting Server.
- Calls this hour/day; Peaks today are derived from operational summary data. By default operational data is submitted once per minute. The hover ToolTip indicates how current the reported values are.

The following procedures describe how to filter the dashboard layout.

Notes: The Voice Platform Dashboard is unrelated to the Dashboard under `Monitoring > Environment`. This dashboard displays alarm and log information for all applications in the environment.

When the Reporting Server is switching from back up mode to primary mode, the dashboard will display inaccurate data for a short period of time.

Procedure: Filtering the Voice Platform Dashboard with GA

Note: In GAX, see the onscreen help about filtering and configuration.

Purpose: To filter the Voice Platform Dashboard using Genesys Administrator.

Prerequisites

- The valid URL for Genesys Administrator—for example, `http://<Genesys Administrator host>/wcm/`.
- The username and password with the correct permissions for running reports.
- The name of Genesys Administrator application—for example, default.

  Note: The default application object is automatically created, and is seen when Genesys Administrator is invoked.

- The host name and port of the Genesys Configuration Server.

Start of procedure:

1. In the web browser's address bar, enter `http://<Genesys Administrator host>/wcm/`.

   The login to `Genesys Administrator` dialog box appears.
2. Enter the following parameters:
   - `User Name`
   - `Password`
   - `Application`
   - `Host Name`
   - `Port`

3. Click `Login`.

   The `Genesys Administrator` screen appears (see Figure 8 on page 300) with the `Monitoring` tab active.

4. From the Navigation panel, select `Voice Platform`.

5. Select the required dashboard—for example, Call Dashboard.

   The selected dashboard appears (for example, see Figure 12).

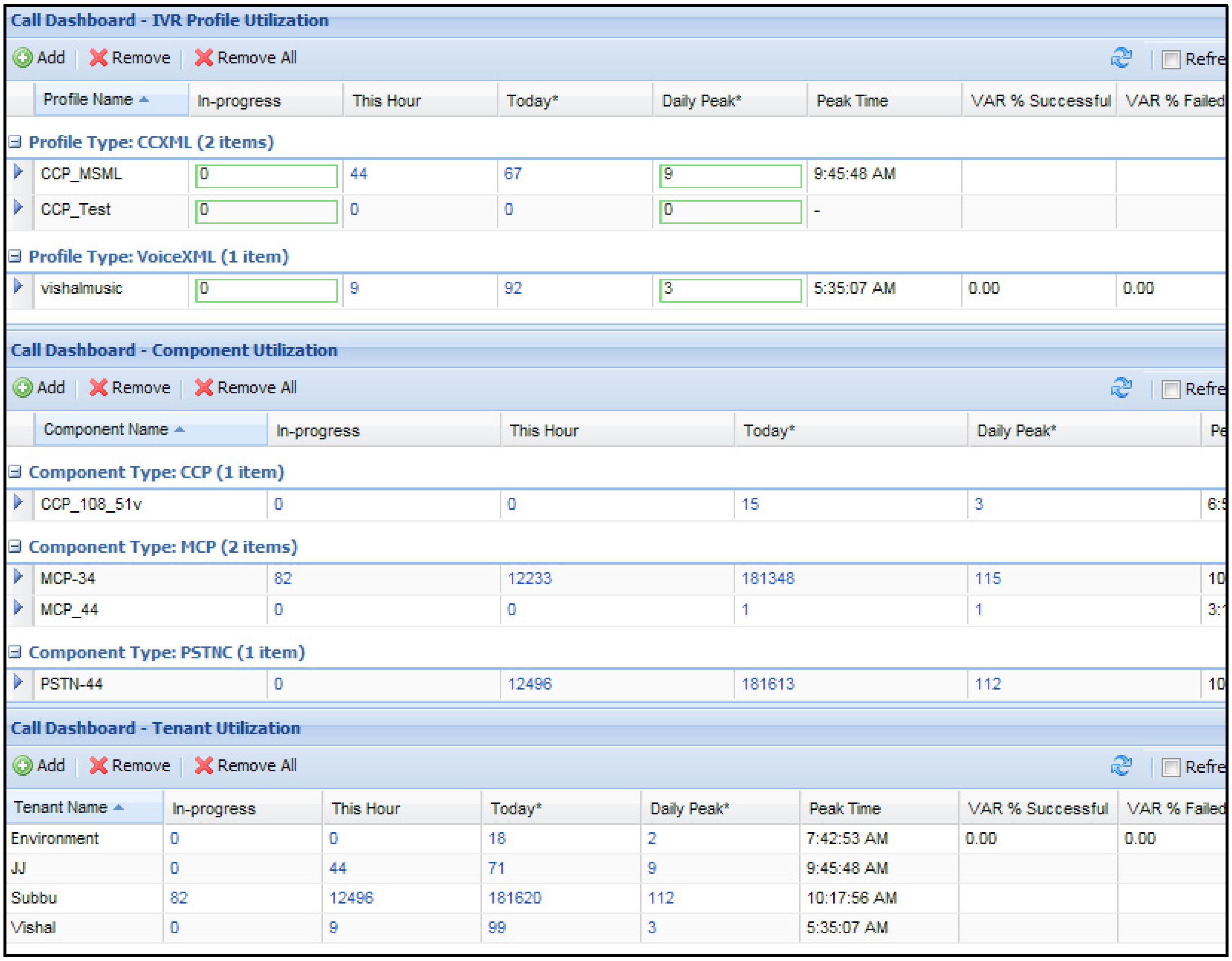


Figure 12: The Call Dashboard

The dashboards enable you to do the following:

- `Add`—Add new IVR Profiles by service type, Components by component type, or Tenants to the Voice Platform Dashboard.
- `Remove`—Remove the selected IVR Profile, Component, or Tenant from the dashboard display.

- Remove All—Remove all IVR Profiles, Components, or Tenants from the dashboard display.

To add IVR Profiles, Components, or Tenants:

a. Click Add. The Select IVR Profiles, the Select Components, or the Select Tenants dialog box appears.

b. Select one or more IVR Profiles, Components, or Tenants by either clicking the check boxes, or selecting the individual rows. When clicking rows, you can also use the Shift/Ctrl keys for multiple selections.

Note: By default, you can select a maximum of 50 IVR Profiles, Components, or Tenants at a time. Use the dashboard.max.filtered.items parameter in the rptui section of the default application to change this limit. For more information on configuring options, see “Configuring GVP Processes in Genesys Administrator” on page 34.

c. Click OK.

To remove IVR Profiles, Component, or Tenants:

a. Select the desired IVR Profiles, Components, or Tenants from the Dashboard.

b. Click Remove.

To remove all of the IVR Profiles, Components, or Tenants:

- Click Remove All.

6. To enable refreshing for the Voice Platform Dashboard, select the Refresh every check box, and enter the desired rate in either seconds or minutes.

End of procedure

Note: In order for the GVP Dashboard to display data, you must connect each GVP application to the SNMP Master Agent application.

Call Dashboard

This section describes the Call Dashboard. It contains the following sections:

IVR Profile Utilization

The IVR Profile Utilization section of the Call Dashboard (see Figure 13) displays current IVR Profile activity for the current day and time up to and including the time that is spent viewing the dashboard. The dashboard pictorially represents the current burst levels for each IVR Profile. They are flagged with the following colors:

- Green—Level 1
- Yellow—Level 2
- Red—Level 3

The level of the colored bar depicts the utilization progress which is also displayed in the hover ToolTip.

Note: Usage limits and bursting limits apply to IVR Profile utilization only.

For more information on usage limits and bursting levels see Chapter 14, "Configuring the Reporting Server," on page 275.

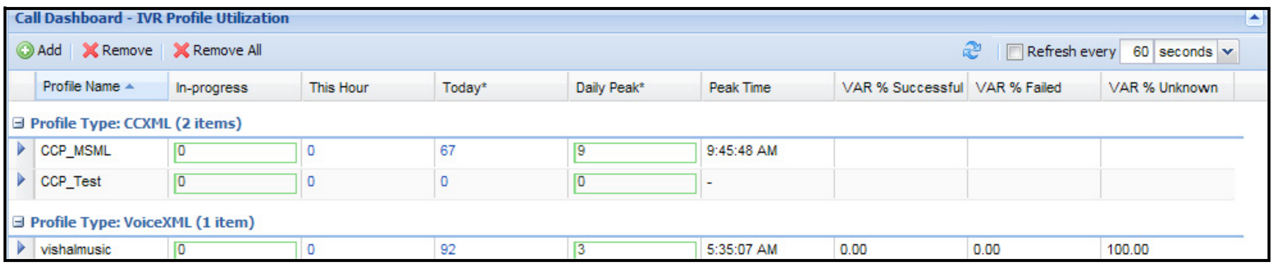


Figure 13: Call Dashboard—IVR Profile Utilization

Table 40 lists and describes the IVR Profile Utilization columns.

Table 40: IVR Profile Utilization Columns

Column	Description
Profile Name	The name of the IVR Profile as provisioned in Genesys Administrator.
In-progress	The number of calls, broken down by Resource Manager call type, that are currently in progress. This value is as current as the CDRs in the Reporting Server database. **Note:** Select the hyperlinked value to display the Active Call List report.

Table 40: IVR Profile Utilization Columns (Continued)

Column	Description
This Hour	The number of calls, broken down by Resource Manager call type, that were processed in the last hour including those calls processed at the last updated time as shown in the Tooltip. **Note:** Select the hyperlinked value to display the Historical Call Browser Report.
Today	The number of calls, broken down by Resource Manager, that were processed today including those calls processed at the last updated time as shown in the tool tip. **Note:** Select the hyperlinked value to display the Historical Call Browser report.
Daily Peak	The greatest number of in-progress calls that are registered for this IVR Profile. **Note:** Select the hyperlinked value to display the IVR Profile Call Peaks report.
Peak Time	The time for which today's peak calls registered.
VAR % Successful	The percentage of calls that had a successful VAR call result.
VAR % Failed	The percentage of calls that had a failed VAR call result.
VAR % Unknown	The percentage of calls that did not have the VAR call result.

Component Utilization

The Component Utilization section of the Call Dashboard (see Figure 14) displays current activity for GVP Components (RM, MCP,CCP, SSG, and PSTNC platforms) for the current day and time up to and including the time that is spent viewing the dashboard.

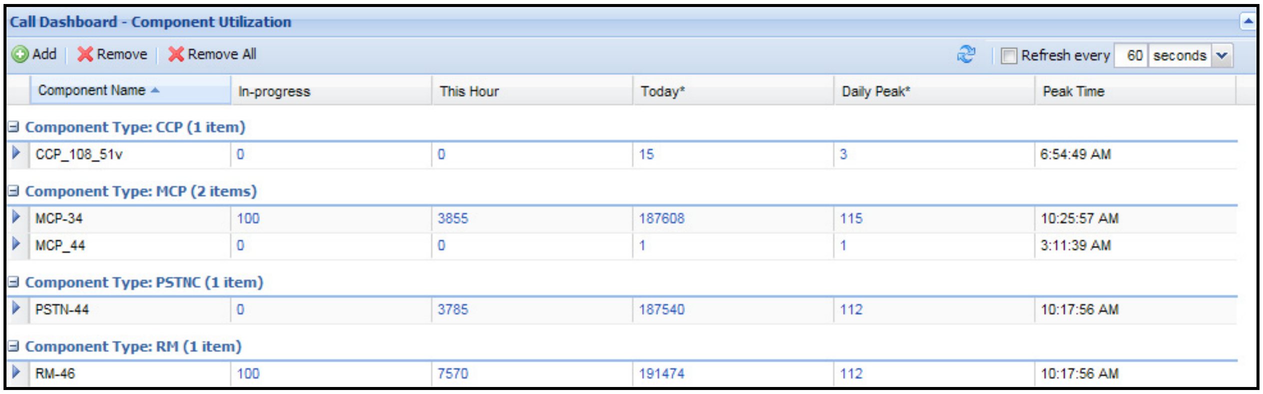


Figure 14: Call Dashboard—Component Utilization

Table 41 lists and describes the Component Utilization columns.

Table 41: Component Utilization Columns

Column	Description
Component Name	The name of the Component as provisioned in Genesys Administrator. **Note:** Component names are listed in alphabetical order according to the component type (CCP, MCP, RM,CTIC, PSTNC).
In-progress	The number of calls, broken down by call type, that are currently in progress. **Note:** Select the hyperlinked value to display the Active Call List report.
This Hour	The number of calls, broken down by call type, that were processed in the last hour. This values is current as of the time that is displayed in the ToolTip. **Note:** Select the hyperlinked value to display the Historical Call Browser report.
Today	The number of calls, broken down by call type, that were processed today including those calls that were processed at the last updated time as shown in the ToolTip. **Note:** Select the hyperlinked value to display the Historical Call Browser report.

Table 41: Component Utilization Columns (Continued)

Column	Description
Daily Peak	The greatest number of in-progress calls that are registered for this Component. **Note:** Select the hyperlinked value to display the Component Call Peaks report.
Peak Time	The time for which today's peak calls registered.

Tenant Utilization

The Tenant Utilization section of the Call Dashboard (see Figure 15) displays current activity for tenants for the current day and time up to and including the time that is spent viewing the dashboard.

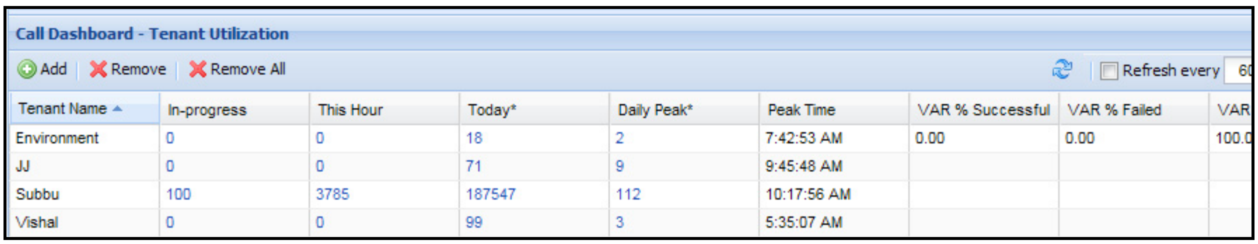


Figure 15: Call Dashboard—Tenant Utilization

Table 42 lists and describes the Tenant Utilization columns.

Table 42: Tenant Utilization Columns

Column	Description
Tenant Name	The name of the Tenant as provisioned in Genesys Administrator.
In-progress	The number of calls, broken down by Resource Manager call type, that are currently in progress. This value is as current as the CDRs in the Reporting Server database. **Note:** Select the hyperlinked value to display the Active Call List report.
This Hour	The number of calls, broken down by Resource Manager call type, that were processed in the last hour including those calls processed at the last updated time as shown in the Tooltip. **Note:** Select the hyperlinked value to display the Historical Call Browser Report.

Table 42: Tenant Utilization Columns (Continued)

Column	Description
Today	The number of calls, broken down by Resource Manager, that were processed today including those calls processed at the last updated time as shown in the tool tip. **Note:** Select the hyperlinked value to display the Historical Call Browser report.
Daily Peak	The greatest number of in-progress calls that are registered for this IVR Profile. **Note:** Select the hyperlinked value to display the IVR Profile Call Peaks report.
Peak Time	The time for which today's peak calls registered.
VAR % Successful	The percentage of calls that had a successful VAR call result.
VAR % Failed	The percentage of calls that had a failed VAR call result.
VAR % Unknown	The percentage of calls that did not have the VAR call result.

SQ Latency Dashboard

The SQ Latency Dashboard (see Figure 16) displays service quality latency data for a set of selected MCP components. The latency data is grouped by latency category, which contains many latency types. The data is broken down for comparison by today's data, this week's data, and this month's data. For more information on SQ Latency data, see Chapter 20, "Service Quality Reports," on page 371.

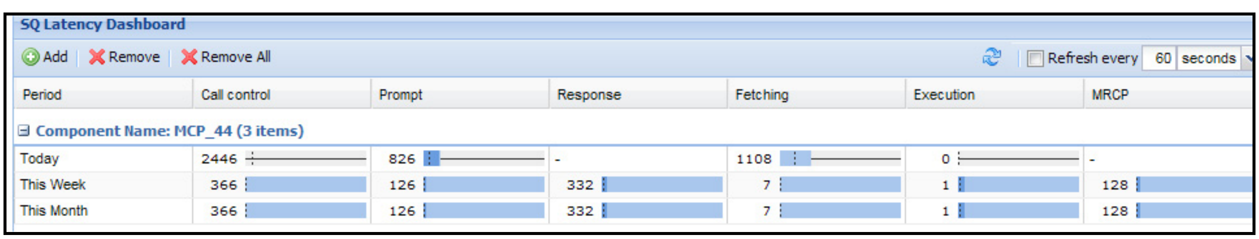


Figure 16: SQ Latency Dashboard

Table 43 lists and describes the SQ Latency Dashboard Columns.

Table 43: SQ Latency Dashboard Columns

Column	Description
Period	The time of the reporting period.
Call Control	The aggregated number of call type latency thresholds. Possible types are: • `CALL_ANSWER` • `CALL_REJECT`
Prompt	The aggregated number of prompt latency thrsholds.Possible types are: • `INBOUND_FIRST_PRMOPT` • `OUTBOUND_FIRST_PROMPT` • `INTERPROMPT` • `INITIAL_RESPONSE`
Response	The aggregated number of response latency thresholds. Possible types are: • `CUMULATIVE_RESPONSE` • `DTMF_INPUT_RESPONSE` • `ASR_INPUT_RESPONSE` • `NOINPUT_RESPONSE` • `RECORDING_RESPONSE` • `TRANSFER_RESPONSE`
Fetching	The aggregated number of fetch latency thresholds. Possible types are: • `PAGE_FETCH` • `AUDIO_FETCH` • `GRAMMAR_FETCH` • `DATA_FETCH` • `JAVA_SCRIPT_FETCH`

Table 43: SQ Latency Dashboard Columns (Continued)

Column	Description
Execution	The aggregated number of execution latency thresholds. Possible types are: • PAGE_COMPILE • JAVA_SCRIPT_EXECUTION
MRCP	The aggregated number of MRCP latency thresholds. Possible types are: • MRCP_ASR_SESSION_ESTABLISH • MRCP_TTS_SESSION_ESTABLISH • MRCP_ASR_SET_PARAMS • MRCP_TTS_SET_PARAMS • MRCP_ASR_STOP • MRCP_TTS_STOP • MRCP_DEFINE_GRAMMAR • MRCP_RECOGNIZE • MRCP_SPEAK

Fetch Dashboard

The Fetch Performance Dashboard (see Figure 17) displays near-real-time statistics of the Media Control Platform and Call Control Platform fetching processes. This dashboard pulls this data from the Reporting Server through SNMP from the Media Control Platform and the Call Control Platform components.

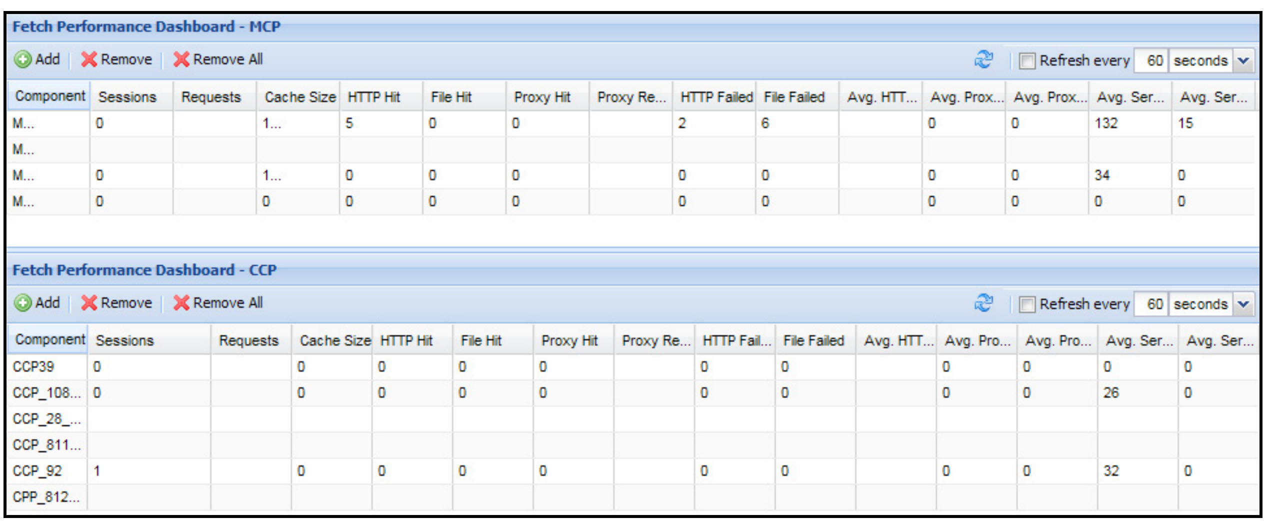


Figure 17: Fetch Performance Dashboard

Table 44 lists and describes the Fetch Performance Dashboard columns.

Table 44: Fetch Performance Dashboard Columns

Column	Description
Component	The name of the component as provisioned in Genesys Administrator.
Sessions	The total number of action sessions per component.
Requests	The total number of active fetch requests.
Cache Size	The current size, in megabytes, of the cache memory.
HTTP Hit	The total number of HTTP cache hits.
File Hits	The total number of file cache hits.
Proxy Hit	The total number of HTTP proxy cache hits.
Proxy Reval	The total number of HTTP proxy cache re-validations.
HTTP Failed	The total number of failed HTTP fetches.
File Failed	The total number of failed file fetches.
Avg. HTTP Resp	The average HTTP response time.
Avg. Proxy Hit	The average HTTP proxy cache hit response time.
Avg. Proxy Reval	The average HTTP proxy cache re-validation response time.
Avg. Server Resp	The average HTTP server response time.
Avg. Server Reval	The average HTTP server re-validation response time.

SSG Dashboard

The SSG Dashboard display near-real-time utilization of the Supplementary Services Gateway (SSG) components. This dashboard pulls this data from the Reporting Server through SNMP from the SSG components. The SSG Dashboard has the following panes:

- SSG IVR Profile Utilization, page 327
- SSG Component Utilization, page 327
- SSG Tenant Utilization, page 328

SSG IVR Profile Utilization

The SSG IVR Profile Utilization dashboard (see Figure 18) displays SSG usage statistics summarized by IVR profile.

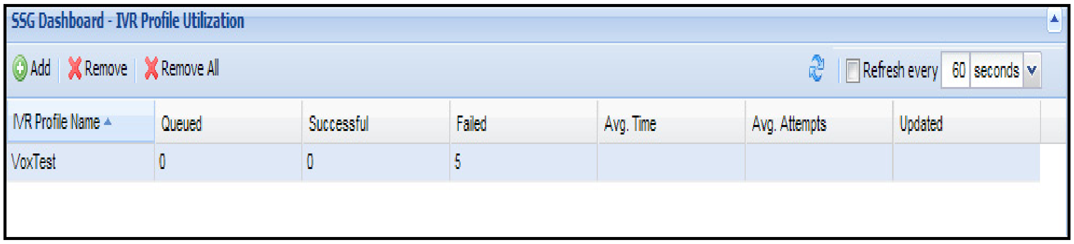


Figure 18: SSG Dashboard—IVR Profile Utilization

Table 45 lists and describes the SSG IVR Profile Utilization columns.

Table 45: SSG IVR Profile Utilization Columns

Column	Description
IVR Profile Name	The name of the IVR Profile.
Queued	The total number of calls that are waiting in queues.
Successful	The total number of successful calls.
Failed	The total number of calls that failed.
Avg. Time	The average time to complete a call.
Avg. Attempts	The average number of attempts to complete a call.

SSG Component Utilization

The SSG Component Utilization dashboard (see Figure 19) displays SSG usage statistics summarized by Component.

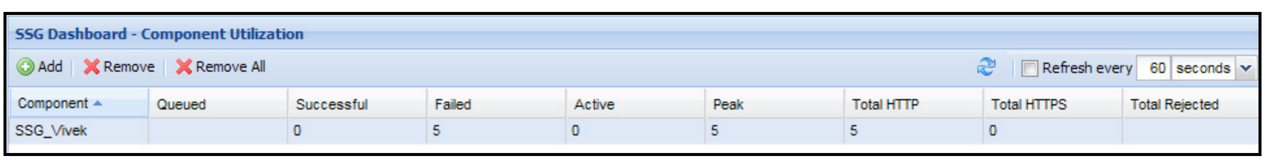


Figure 19: SSG Dashboard—Component Utilization

Table 46 lists and describes the SSG Component Utilization columns.

Table 46: SSG Component Utilization Columns

Column	Description
Component Name	The name of the SSG component.
Queued	The total number of calls that are waiting in queues.
Successful	The total number of successful calls.
Failed	The total number of calls that failed.
Active	The total number of active calls.
Peak	The peak number of active calls.
Total HTTP	The total number of HTTP requests.
Total HTTPS	The total number of HTTPS requests.
Total Rejected	The total number of rejected requests.

SSG Tenant Utilization

The SSG Tenant Utilization dashboard (see Figure 20) displays SSG usage statistics grouped by tenants.

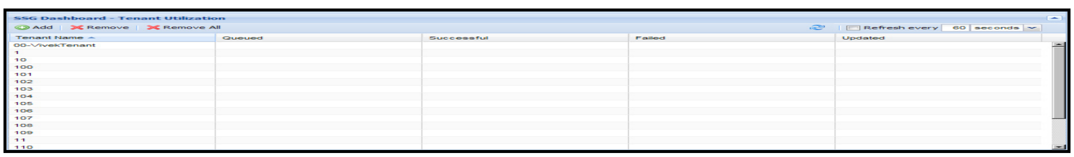


Figure 20: SSG Dashboard—Tenant Utilization

Table 47 lists and describes the SSG Tenant Utilization columns.

Table 47: SSG Tenant Utilization Columns

Column	Description
Tenant Name	The name of the tenant.
Queued	The total number of calls that are waiting in queues.
Successful	The total number of successful calls.
Failed	The total number of calls that failed.

PSTNC Dashboard

The PSTN Connector (PSTNC) dashboard (see Figure 21) displays near-real-time utilization data of PSTNC components and the PSTN boards that they manage. This dashboard pulls this data from the Reporting Server through SNMP from the PSTN Connector components.

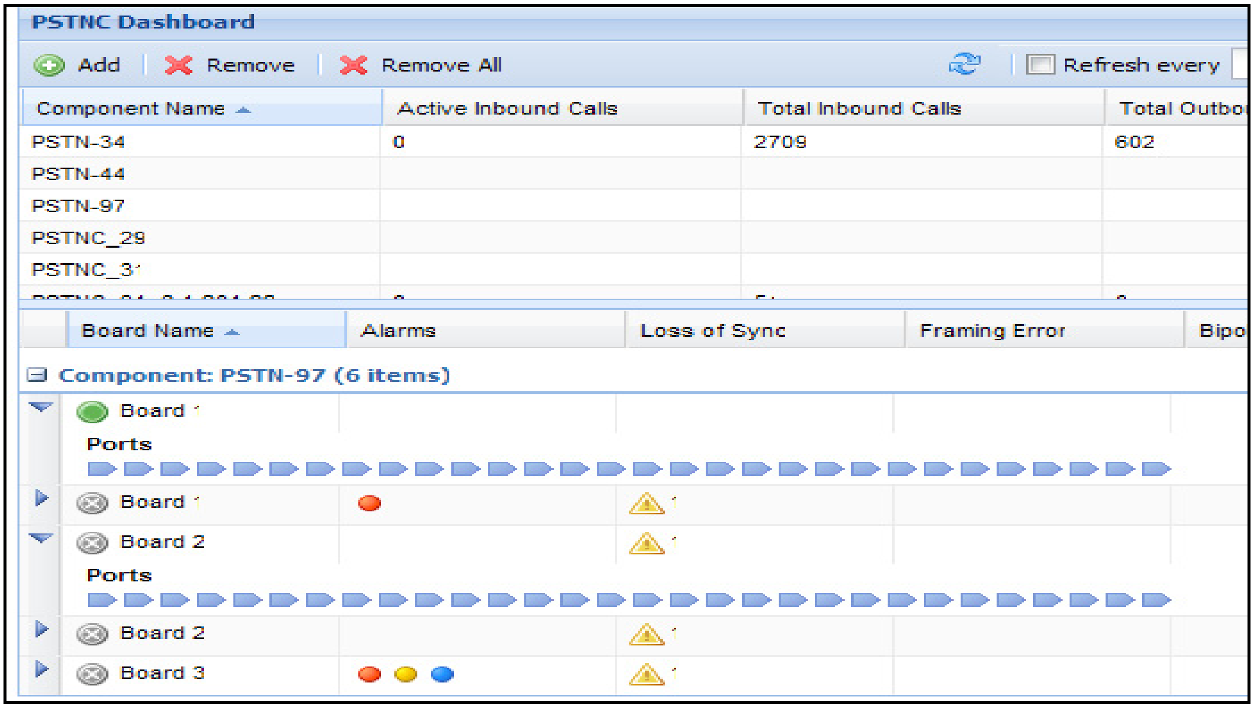


Figure 21: PSTNC Dashboard

Table 48 lists and describes the PSTNC dashboard columns.

Table 48: PSTNC Dashboard Columns

Column	Description
Component Name	The name of the PSTN Connector component.
TDM Inbound Total	The total number of TDM inbound calls received.
TDM Outbound Total	The total number of TDM outbound calls attempted.
TDM Inbound Active	The total number of TDM inbound calls received that are currently active.
TDM Outbound Active	The total number of TDM outbound calls attempted that are currently active.
SIP Inbound Total	The total number of SIP inbound calls received.

Table 48: PSTNC Dashboard Columns (Continued)

Column	Description
SIP Outbound Total	The total number of SIP outbound calls initiated.
SIP Inbound Active	The number of SIP inbound calls that are currently active.
SIP Outbound Active	The number of SIP outbound calls that are currently active.
Board Name	The name of the PSTN board. **Note:** Hover over the board name to display the board description in the tool tip. The D-channel status is indicated by the green (up) or grey (down) icon beside the board name.
Alarms	The status of the PSTN board. This is indicated with a colored icon: • red—Indicates that the incoming signal is corrupt. On the Dialogic card, the red LED is lit. • yellow—Indicates that the network has failed. On the Dialogic card, the yellow LED is lit. • blue—Indicates the absence of an incoming signal. On the Dialogic card, the red and green LEDs are lit. **Note:** The lack of an icon means that there are no active alarms of that color for this board. Multiple icons of different colors can appear at the same time.
Loss of Sync	The PSTN board is not synchronized.
Framing Error	The PSTN board has encountered a framing error.
Bipolar Violation	The PSTN board has encountered a bipolar violation.

Note: Select the arrow (>) beside the board to view each port. Hover over each port number to view the individual details of that port. Right-click a port to reset or disable it.

CTIC Dashboard

The CTI Connector (CTIC) dashboard (see Figure 22) displays near real-time CTI Connector component and ICM connection statistics, which are polled by the Reporting Server by using SNMP.

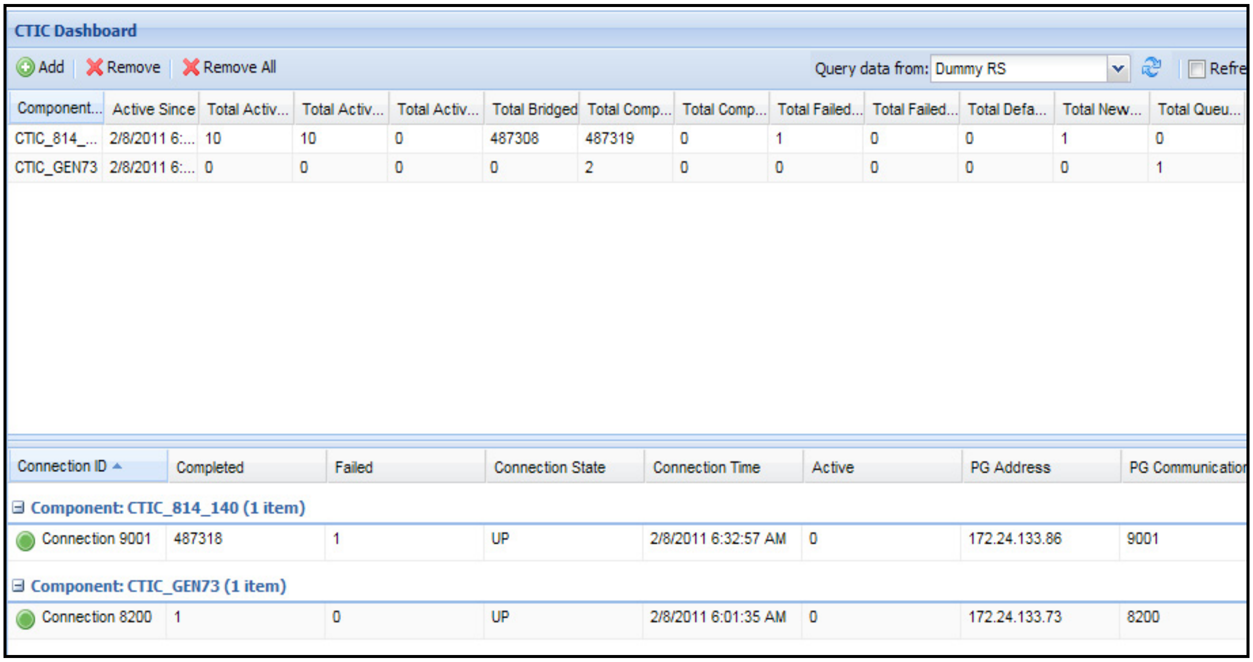


Figure 22: CTIC Dashboard

Table 49 lists and describes the CTIC dashboard columns.

Table 49: CTIC Dashboard Columns

Column	Description
Component Name	The name of the CTI Connector component.
Active Since	The time that the component became active.
Total Active Calls	The total number of active calls.
Total Active Inbound Calls	The total number of active inbound calls.
Total Active Outbound Calls	The total number of active outbound calls.
Total Bridged	The total number of bridged calls.
Total Completed	The total number of completed calls.
Total Completed in IVR	The total number of calls completed in IVR.
Total Failed at CTI	The total number of calls failed at CTI.
Total Failed at GVP	The total number of calls failed at the Voice Platform.

Table 49: CTIC Dashboard Columns (Continued)

Column	Description
Total Default Agent	The total number of Default Agent responses received.
Total NewCall Failed	The total number of failed NewCall requests.
Total Queued Calls	The total number of queued calls.
Total Route Request Failed	The total number of failed Route requests.
Total Route Responses	The total number of Route responses received.
Connection ID	The connection ID of the CTI Connector.
Completed	The total number of calls successfully completed on this port.
Failed	The total number of failed calls on this port.
Connection State	The connection state of the CTI Connector (either UP or DOWN).
Connection Time	The time at which the connection request from Intelligent Contact Management (ICM) is accepted by CTI Connector.
Active	The current number of active calls on this port.
PG Address	The IP address of the ICM from which the connection request is accepted.
PG Communication Port	The listening port on which CTIC accepts new connection requests from ICM.
Tenant	The name of the tenant for which CTIC is handling the calls on this port.

Chapter

18 Real-Time Reports

This chapter describes the available reports that display real-time data. It contains the following sections:

Overview

The real-time reports display statistics of the current call that is in progress. However, real-time data updates are not instantaneous, because there may be a slight delay while the Media Control Platform (MCP), the Call Control Platform (CCP), or Resource Manager (RM) sends data to the Reporting Server.

Call detail records (CDR) and call events are delivered in batches to the Reporting Server. By default the batch size is 500 CDRs or ten seconds. This means that a message will be sent either when 500 CDR updates are queued, or ten seconds has expired, whichever occurs first. You can reconfigure the system to be more *real-time* by changing the batch size—for example, changing it to 1. This means that a CDR update or the call event will be delivered to the Reporting Server as soon as it is raised by the component.

There are performance implications to changing the batch size. For more information, see "Configuring the Reporting Server" on page 275.

Active Call Browser

The Active Call Browser report (see Figure 23) displays the list of calls that currently are being processed by GVP. It also includes any call that the Reporting Server has not marked as timed out.

A call is considered as timed out if the call processing component has unexpectedly shut down, or if the connection between the call processing component and the Reporting Server is broken. In either case, the Reporting Server has not received the update indicating that the call ended. If the connection is down, the Reporting Server will eventually receive the update; however, if the processing component unexpectedly shut down, the call will stay as timed out.

The Reporting Server processes timed out calls once hourly (by default), and marks calls as timed out when they have been in progress for more than a configured period of time (by default 3 hours).

The following procedure describes how to generate the Active Call Browser report.

Procedure: Generating the Active Call Browser Report with GA

Note: In GAX, see the onscreen help about filtering and configuration.

Purpose: To generate the Active Call Browser report by using Genesys Administrator.

Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instructions:
   a. On the `Time Range` tab, select the appropriate `Time Range` (on page 310).
   b. On the `Call Info` tab, select the appropriate `Call Type` (on page 312) and `Call State` (on page 312).
   c. On the `IVR Profile` tab, specify the `IVR Profiles` (on page 336) to include in the report. To modify the list of IVR Profiles, do any of the following:
      i. Too add an IVR Profile, click `Add`. Select the IVR Profiles from the list.
      ii. To remove an IVR Profile from the list, click `Remove`.
      iii. To remove all IVR Profiles the list. Click `Remove All`.
   d. On the `Components` tab, select the `Component Type` (on page 311).To build a list of `Components` (on page 311) for which active calls are to be reported on:
      i. To add a component, click `Add`.
      ii. To remove a component, select it and click `Remove`.

  iii. To remove all components, select `Remove All`.

  e. On the `Tenants` tab, select the `Tenant` (on page 311).To build a list of `Tenants` (on page 311) for which active calls are to be reported on:

    i. To add a tenant, click `Add`.

    ii. To remove a tenant, select it and click `Remove`.

    iii. To remove all tenants, select `Remove All`.

  f. On the `Other` tab, enter the `Virtual Reporting Object`, the `Remote URI`, and/or the `Local URI` (on page 313).

3. Click `Generate Report`. Continue from Step 8 on page 302.

   Click the specific GVP GUID link to display all component specific call detail records for that GVP GUID.

End of procedure

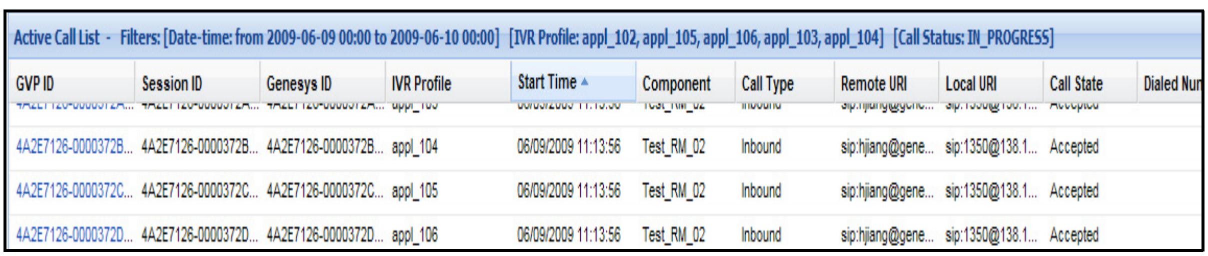
Notes: If you do not select an IVR Profile, all IVR Profile data is displayed. If you do not select a Component, only RM call detail records will display. You must select the MCP component to view MCP call detail records.

Data is returned if no filter is selected.

When the backup Reporting Server switches to primary mode, the Active Call Browser report will show inaccurate data for a short period of time.

You can select multiple IVR Profiles and multiple Components.

For more information on the details of completed calls, see "Component Call Peaks" on page 351.


Active Call List - Filters: [Date-time: from 2009-06-09 00:00 to 2009-06-10 00:00] [IVR Profile: appl_102, appl_105, appl_106, appl_103, appl_104] [Call Status: IN_PROGRESS]

GVP ID	Session ID	Genesys ID	IVR Profile	Start Time	Component	Call Type	Remote URI	Local URI	Call State	Dialed Nur
4A2E7126-0000372B...	4A2E7126-0000372B...	4A2E7126-0000372B...	appl_104	06/09/2009 11:13:56	Test_RM_02	Inbound	sip:hjiang@gene...	sip:1350@138.1...	Accepted	
4A2E7126-0000372C...	4A2E7126-0000372C...	4A2E7126-0000372C...	appl_105	06/09/2009 11:13:56	Test_RM_02	Inbound	sip:hjiang@gene...	sip:1350@138.1...	Accepted	
4A2E7126-0000372D...	4A2E7126-0000372D...	4A2E7126-0000372D...	appl_106	06/09/2009 11:13:56	Test_RM_02	Inbound	sip:hjiang@gene...	sip:1350@138.1...	Accepted	

Figure 23: Active Call Browser Report

Table 50 describes the fields for the summary level for the Active Call Browser report.

Table 50: Active Call List Report Summary Fields

Field	Description
GVP GUID	The globally unique ID that identifies a complete interaction with GVP. This ID is generated by the Resource Manager, and is passed to all the resources that provide service for the call. For more information on the GVP Session ID, see "Session Identifiers" on page 26.
Site ID	The site ID from which the call originated.
Session ID	The GVP Component ID that is generated by the component to identify the call leg. For more information on the GVP Component ID, see "Session Identifiers" on page 26.
Genesys UUID	The Genesys CallUUID that is generated by T-Server or SIP Server. For more information on the Genesys CallUUID, see "Session Identifiers" on page 26.
IVR Profile	The name of the IVR Profile that is selected.
Start Time	The start date and start time of the call.
Component	The name of the component (RM, MCP, or CCP) that is selected.
Call Type	The type of the call. Valid call types are the following: • Inbound • Outbound • Bridged • Unknown • New Call • Createccxml • External
Remote URI	The SIP address of the calling party.
Local URI	The SIP address of the component that received the call.

Table 50: Active Call List Report Summary Fields (Continued)

Field	Description
Call State	The state of the call. Valid call states are the following: • IVR—The call is being processed by MCP. • Accepted—The call has been received by RM, but not landed on the VoiceXML platform or transferred to an agent. • Transferring—The call is being transferred to an agent. • Transferred—The call was successfully transferred to an agent.
Profile Usage	The number of calls in progress for the given IVR Profile. This is recorded when the Resource Managers associates the call with the IVR Profile. The burst level is indicated with a color bar. The colors green, yellow, and red correspond to burst levels 1, 2, and 3 respectively.
Tenant Usage	The number of calls in progress for the tenant. The burst level is indicated with a color bar. The colors green, yellow, and red correspond to burst levels 1, 2, and 3 respectively.

The Active Call List Details report breaks down the detail recorded of the selected call according to component type (see Figure 24).

Compone...	GVP ID	Genesys ID	Session ID	Appli
CCP	00000123-00000123-00000123-0000...	A111Q01T1H1C12O1LQM9LFUR0000...	00000068-00000068-00000068-0000...	RPTU
MCP	00000123-00000123-00000123-0000...	A111Q01T1H1C12O1LQM9LFUR0000...	00000067-00000067	RPTU
RM	00000123-00000123-00000123-0000...	A111Q01T1H1C12O1LQM9LFUR0000...	00000123-00000123-00000123-0000...	RPTU

Figure 24: Active Call List Details Report

Table 51 describes the fields for the Active Call Browser Details report.

Table 51: Active Call List Details Report Fields

Fields	Description
Component Type	The type of component. The possible components are: • RM (Resource Manager) • MCP (Media Control Platform) • CCP (Call Control Platform)

Table 51: Active Call List Details Report Fields (Continued)

Fields	Description
GVP ID	The globally unique ID that identifies a complete interaction with GVP. This ID is generated by the Resource Manager, and is passed to all the resources that provide service for the call. For more information on the GVP Session ID, see "Session Identifiers" on page 26.
Genesys ID	The Genesys CallUUID that is generated by T-Server or SIP Server. For more information on the Genesys CallUUID, see "Session Identifiers" on page 26.
Session ID	The GVP Component ID that is generated by the component to identify the call leg. For more information on the GVP Component ID, see "Session Identifiers" on page 26.
IVR Profile	The name of the IVR Profile as seen in Genesys Administrator or Configuration Manager.
Component	The name of the Component application as seen in Genesys Administrator or Configuration Manager.
Start Time	The start date and start time of the call.
End Time	The end date and end time of the call if the call has completed.
Call Status	The state of the call. Valid call states are the following: • Completed • Timed Out • In Progress
Call Type	The type of call. Valid call types are the following: • Inbound • Outbound • Bridged • Unknown • New Call • Createccxml • External
Remote URI	The remote Uniform Resource Identifier.
Local URI	The local Uniform Resource Identifier.

Table 51: Active Call List Details Report Fields (Continued)

Fields	Description
Call State	The state of the call. Valid call states are the following: • IVR—The call is being processed by MCP. • Accepted—The call has been received by RM, but not landed on the VoiceXML platform or transferred to an agent. • Transferring—The call is being transferred to an agent. • Transferred—The call was successfully transferred to an agent.
Dialed Number	The number dialed.
Profile Usage	The number of calls in progress for the given IVR Profile. This is recorded when the Resource Manager associates the call with the IVR Profile. The burst level is indicated with a color bar. The colors green, yellow, and red correspond to burst levels 1, 2, and 3 respectively.
Tenant Usage	The number of calls in progress for the tenant. The burst level is indicated with a color bar. The colors green, yellow, and red correspond to burst levels 1, 2, and 3 respectively.
Session Start Origin	The source from which the call started.
Parent Session ID	The unique identifier of the parent session.

Chapter

19 Historical Reports

This chapter describes the available historical reports. It contains the following sections:

Overview

The historical reports display call detail records, call arrival and summary information over a selected period of time, of the specified IVR Profiles, Components, and Tenants.

The Historical Call Summary and Historical Peaks reports display the data in both a pictorial graph and a table. The graph provides the following navigation features:

- To zoom in, drag from left to right on a selected area.

- To pan around the graph, right-click and drag the graph.
- To restore to normal view (100%), double-click the graph.
- To turn on or off the visibility of the individual series, click the *eye* icon in the chart legend.
- Service Quality reports apply to NGi VoiceXML applications, and are found in Genesys Administrator. GVP 8.1.5 and thereafter are NGi-only platforms unless you run MCP 8.1.4 to incorporate support for GVPi applications.

Note: If you are viewing reports in Microsoft Internet Explorer, Genesys recommends that you use 1280 x 1024 (or greater) screen resolution, or use Mozilla or Firefox, in order for the graphs to resize and repaint quickly.

IVR Profile Call Arrivals

The IVR Profile Call Summary report (see Figure 25) lists a summary of call arrival data that is submitted for each IVR Profile selected, for a given period of time.

The following procedure describes how to generate the IVR Profile Call Arrivals report.

Procedure: Generating the IVR Profile Call Arrivals Report with GA

Note: In GAX, see the onscreen help about filtering and configuration.

Purpose: To generate the IVR Profile Call Arrivals report using Genesys Administrator.

Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instruction:
   a. On the `Time Range` tab, select the appropriate `Time Range` (on page 310) and the `Granularity` (on page 310).

b. On the `IVR Profile` tab, specific the `IVR Profiles` (on page 311) to include in the report. To modify the list of IVR Profiles, do any of the following:

   i. Too add an IVR Profile, click `Add`. Select the IVR Profiles from the list.

   ii. To remove an IVR Profile from the list, click `Remove`.

   iii. To remove all IVR Profiles the list. Click `Remove All`.

3. Click `Generate Report`. Continue from Step 8 on page 302.

End of procedure

To view the matching Historical Peaks data, select the `Historical Peaks` link from the `Related Reports` section of the `Tasks` panel.

Note: You can select up to a maximum of eight IVR Profiles; however, you cannot access the IVR Profile Peaks report from the `Tasks` panel.

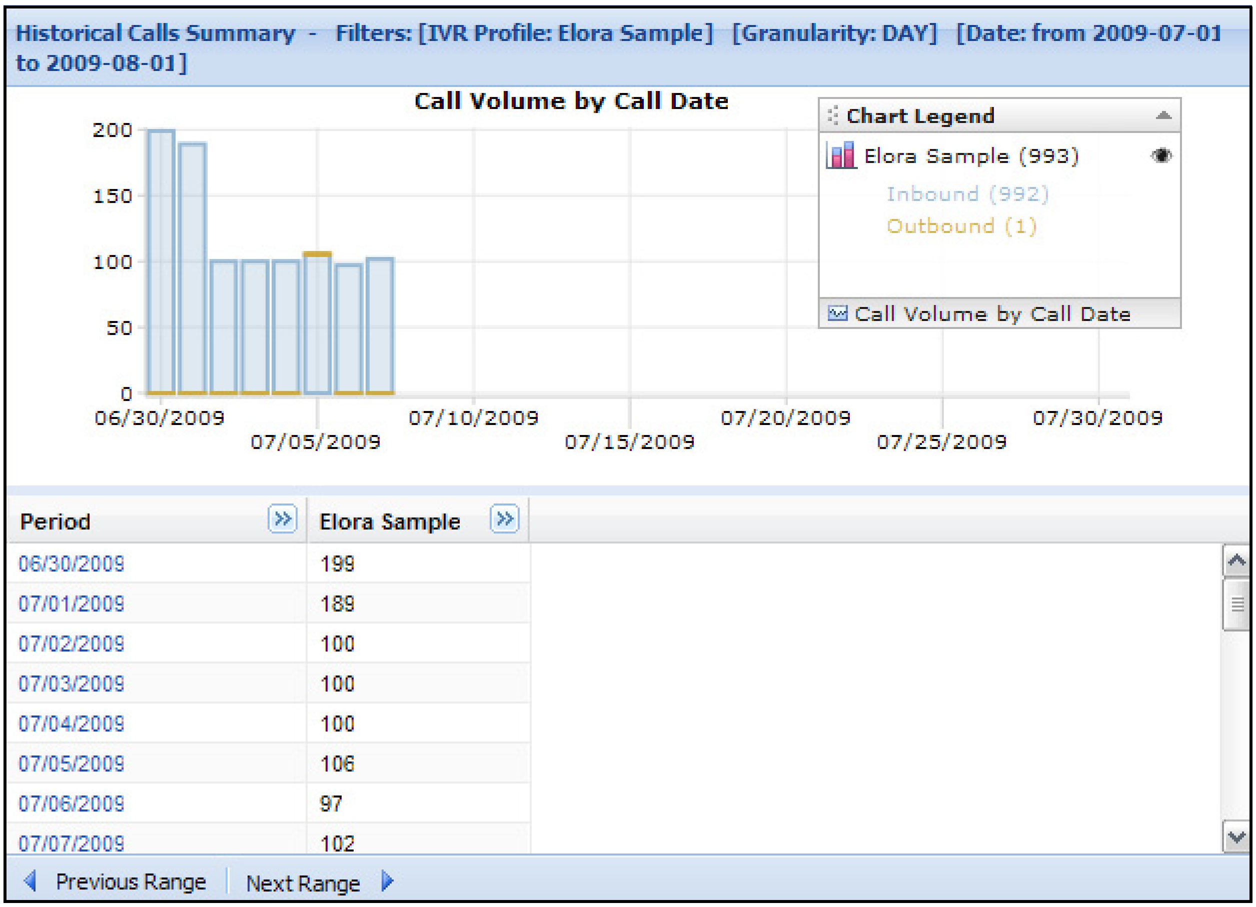


Figure 25: IVR Profile Call Arrivals Report

Table 52 describes the fields for the IVR Profile Call Arrivals report.

Table 52: IVR Profile Call Arrivals Report Fields

Field	Description
Period	The date and time of the call for the given granularity period. Click >> to view the `End of Period` column.
IVR Profile Name	The total number of calls for the given IVR Profile.
IVR Profile Name (Inbound)	The total number of inbound calls for the given IVR Profile.
IVR Profile Name (Outbound)	The total number of outbound calls for the given IVR Profile.
IVR Profile Name (Unknown)	The total number of unknown calls for the given IVR Profile.

Component Call Arrivals

The Component Call Arrivals report (see Figure 26) lists a summary of call arrival data that is submitted for each component selected, for a given period of time.

The following procedure describes how to generate the Component Call Arrivals report.

Procedure:Generating the Component Call Arrivals Report

Purpose: To generate the Component Call Arrivals report using Genesys Administrator.

Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instruction:
   a. On the `Time Range` tab, select the appropriate `Time Range` (on page 310) and the `Granularity` (on page 310).

b. On the `Components` tab, select the `Component Type` (on page 311).To build a list of `Components` (on page 311) for which active calls are to be reported on:

   i. To add a component, click `Add`.

   ii. To remove a component, select it and click `Remove`.

   iii. To remove all components, select `Remove All`.

3. Click `Generate Report`. Continue from Step 8 on page 302.

End of procedure

To view the matching Component Peaks data, select the `Historical Peaks` link from the `Related Reports` section of the `Tasks` panel.

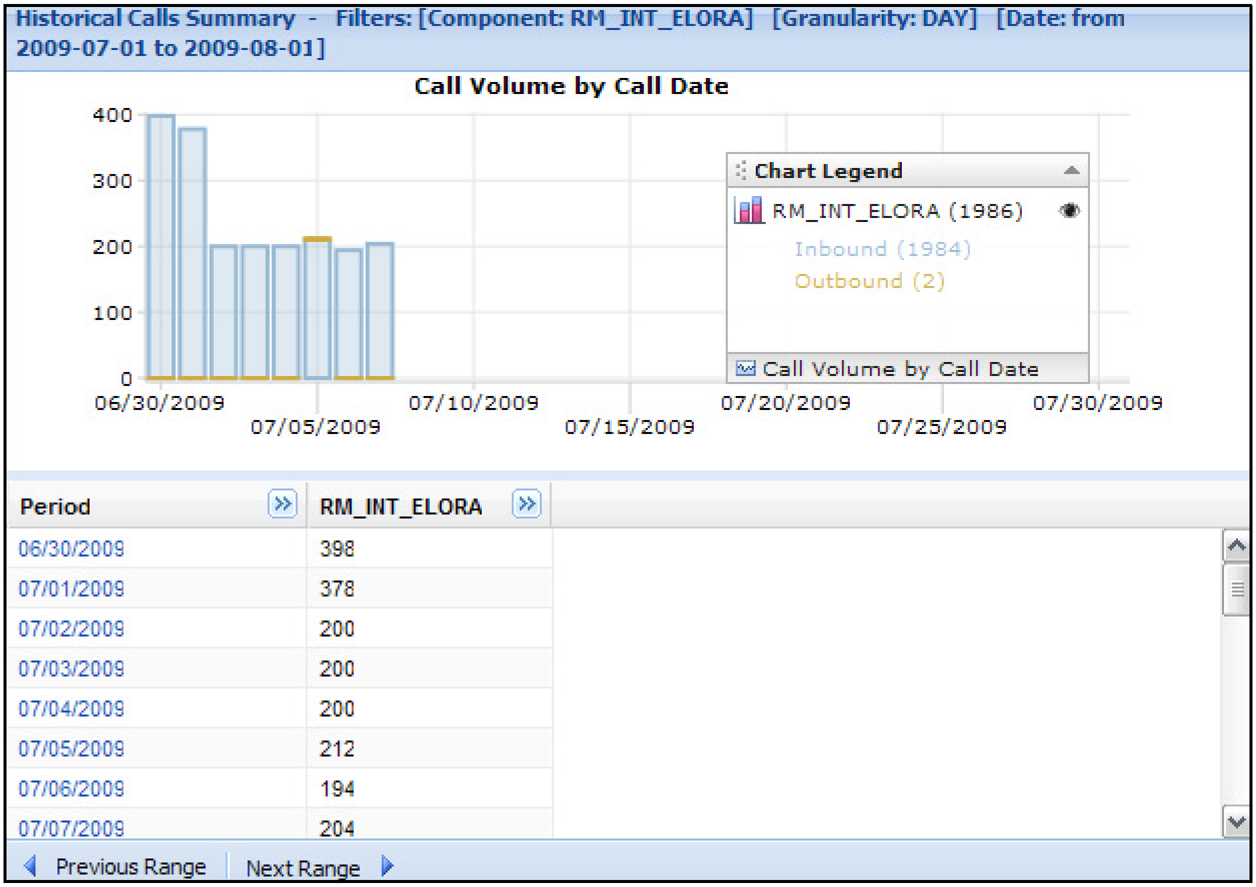


Figure 26: Component Call Arrivals Report

Table 53 describes the fields for the Component Call Arrivals report.

Table 53: Component Call Arrivals Report Fields

Field	Description
Period	The date and time of the call for the given granularity period. Click >> to view the End of Period column.
Component Name	The total number of calls for the given IVR Profile.
Component Name (Inbound)	The total number of inbound calls for the given IVR Profile.
Component Name (Outbound)	The total number of outbound calls for the given IVR Profile.
Component Name (Unknown)	The total number of unknown calls for the given IVR Profile.

Tenant Call Arrivals

The Tenant Call Arrivals report (see Figure 27) lists a summary of call arrival data that is submitted for each tenant selected, for a given period of time.

The following procedure describes how to generate the Tenant Call Arrivals report.

Procedure:Generating the Tenant Call Arrivals Report

Purpose: To generate the Tenant Call Arrivals report using Genesys Administrator.

Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instruction:
   a. On the Time Range tab, select the appropriate Time Range (on page 310) and the Granularity (on page 310).
   b. On the Tenants tab, select the Tenant (on page 311).To build a list of Tenants (on page 311) for which active calls are to be reported on:
      i. To add a tenant, click Add.
      ii. To remove a tenant, select it and click Remove.
      iii. To remove all tenants, select Remove All.

3. Click `Generate Report`. Continue from Step 8 on page 302.

End of procedure

To view the matching Tenant Peaks data, select the `Historical Peaks` link from the `Related Reports` section of the `Tasks` panel.

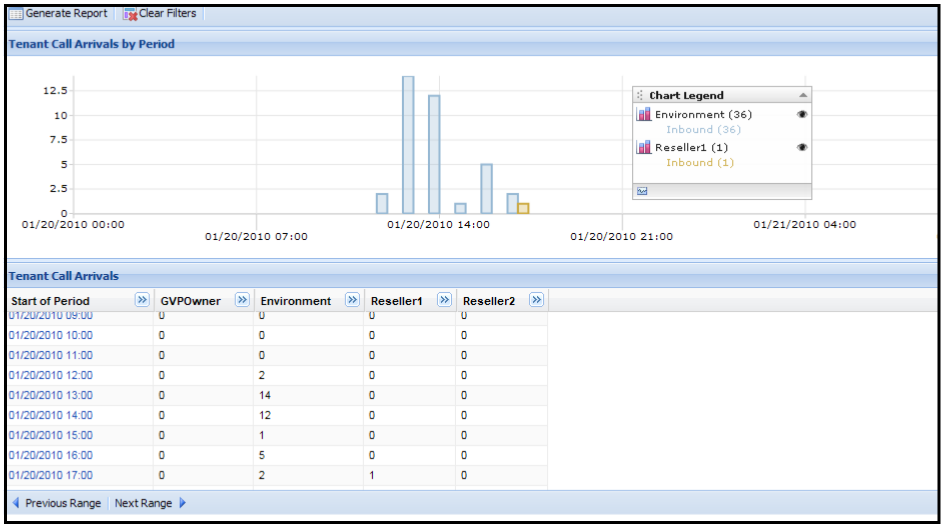


Figure 27: Tenant Call Arrivals Report Get

Table 54 describes the fields for the Tenant Call Arrivals report

Table 54: Tenant Call Arrivals Report Fields

Field	Description
Start of Period	The date and time of the call for the given granularity period. Click >> to view the `End of Period` column.
Tenant Name	The total number of calls for the given Tenant.
Tenant Name (Inbound)	The total number of inbound calls for the given Tenant.
Tenant Name (Outbound)	The total number of outbound calls for the given Tenant.
Tenant Name (Unknown)	The total number of unknown calls for the given Tenant.

Media Service Call Arrivals

The Genesys Media Server (GMS) collects, summarizes and reports on session arrivals for each media service requested by SIP Server over MSML.

For each of the services in Table 55, the GMS provides session arrival (total sessions) for the RS deployment or for each tenant if in a multi-tenant environment. The time period for each arrival count can be hour or higher granularity (see Time Period in “Report Filters” on page 310).

The report generates data for the following types of calls:

Table 55: Media Service Call Arrival Reporting

Media Service	Report Details
GMS Treatment	Total arrivals for 'Treatment' sessions (excluding Play Application use of VoiceXML)
GMS CPD	Total arrivals for 'CPD' sessions
GMS Record	Total arrivals for 'Recording' sessions
GMS Conference	Total arrivals for 'Conferencing' sessions
GMS Media	Total arrivals for 'Media' sessions

This report can be generated for specific tenants or for the entire deployment.

- See “Generating a Report with GA” on page 298 for the procedure steps.
- See “GAX Report Generation Table” on page 304 for the specific choices to make, to generate this report (and others).

IVR Profile Call Peaks

The IVR Profile Call Peaks report (see Figure 28) provides the peak volume of calls during a given period of time for a given IVR Profile.

The following procedure describes how to generate the IVR Profile Call Peaks report.

Procedure:Generating the IVR Profile Call Peaks Report

Purpose: To generate the IVR Profile Call Peaks report using Genesys Administrator.

Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instruction:
   a. On the `Time Range` tab, select the appropriate `Time Range` (on page 310) and the `Granularity` (on page 310).
   b. On the `IVR Profile` tab, specific the `IVR Profiles` (on page 311) to include in the report. To modify the list of IVR Profiles, do any of the following:
      i. Too add an IVR Profile, click `Add`. Select the IVR Profiles from the list.
      ii. To remove an IVR Profile from the list, click `Remove`.
      iii. To remove all IVR Profiles the list. Click `Remove All`.
3. Click `Generate Report`. Continue from Step 8 on page 302.

End of procedure

Note: You can select one IVR Profile at a time.

To view the matching IVR Profile Call Arrivals data, select the `Historical Arrivals` link from the `Related Reports` section of the `Tasks` panel.

The Peaks Volume, which is shown on the graph, counts the peak number of calls that is observed during the specified time range, according to the selected granularity level.

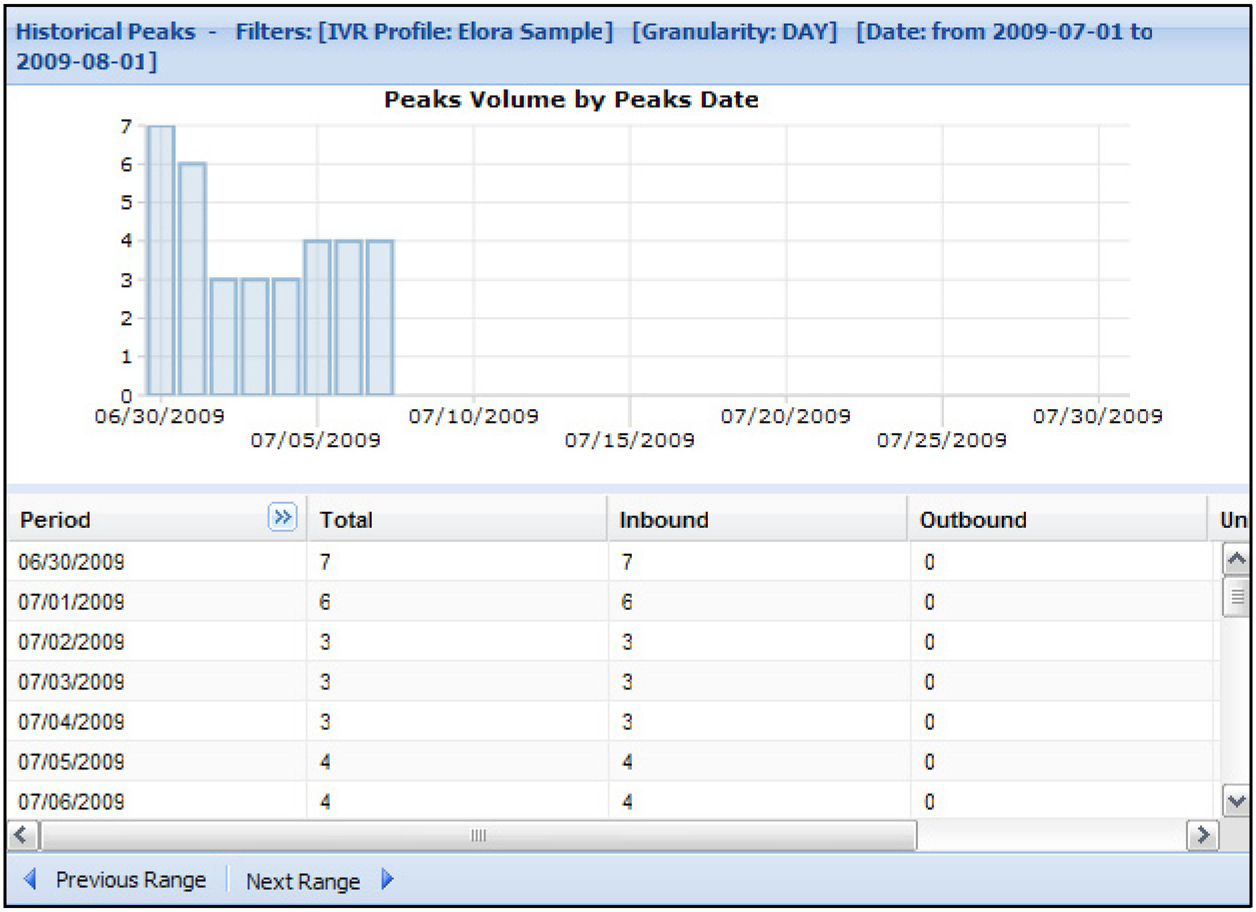


Figure 28: IVR Profile Call Peaks Report

Table 56 describes the fields for the IVR Profile Call Peaks report.

Table 56: IVR Profile Call Peaks Report Fields

Field	Description
Period	The date and time of the call for the given granularity period. Click >> to view the `End of Period` column.
Total	The total number of calls for the given peak period.
Inbound	The total number of inbound calls for the given peak period.
Outbound	The total number of outbound calls for the given peak period.
Unknown	The total number of unknown calls for the given peak period.

Component Call Peaks

The Component Call Peaks report (see Figure 29) provides the peak volume of calls during a given period of time for a given Component.

The following procedure describes how to generate the Component Call Peaks report.

Procedure:Generating the Component Call Peaks Report

Purpose: To generate the Component Call Peaks report using Genesys Administrator.

Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instruction:
   a. On the `Time Range` tab, select the appropriate `Time Range` (on page 310) and the `Granularity` (on page 310).
   b. On the `Components` tab, select the `Component Type` (on page 311).To build a list of `Components` (on page 311) for which active calls are to be reported on:
      i. To add a component, click `Add`.
      ii. To remove a component, select it and click `Remove`.
      iii. To remove all components, select `Remove All`.
3. Click `Generate Report`. Continue from Step 8 on page 302.

End of procedure

- To view the matching Component Call Arrivals data, select the `Historical Arrivals` link from the `Related Reports` section of the `Tasks` panel.

Note: You can select up to eight components at a time.

The Peaks Volume, which is shown on the graph, counts the peak number of calls that is observed during the specified time range, according to the selected granularity level.

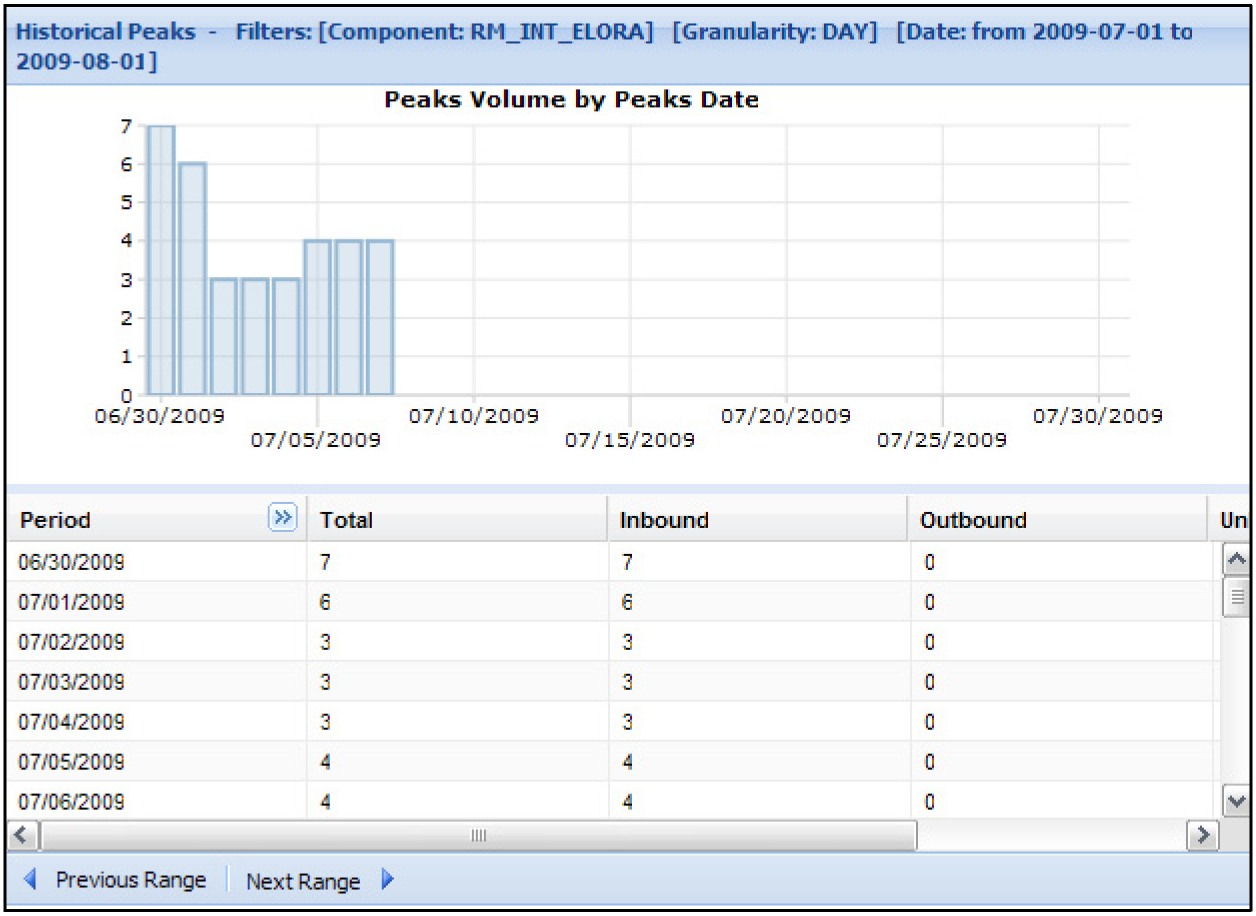


Figure 29: Component Call Peaks Report

Table 57 describes the fields for the Component Call Peaks report.

Table 57: Component Call Peaks Report Fields

Field	Description
Period	The date and time of the call for the given granularity period. Click >> to view the `End of Period` column.
Total	The peak number of concurrent calls reached during the time period.
Inbound	The total number of inbound calls for the given peak period.
Outbound	The total number of outbound calls for the given peak period.
Unknown	The total number of unknown calls for the given peak period.

Tenant Call Peaks

The Tenant Call Peaks report (see Figure 30) provides the peak volume of calls during a given period of time for a given Tenant.

The following procedure describes how to generate the Tenant Call Peaks report.

Procedure:Generating the Tenant Call Peaks Report

Purpose: To generate the Tenant Call Peaks report using Genesys Administrator.

Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instruction:
   a. On the `Time Range` tab, select the appropriate `Time Range` (on page 310) and the `Granularity` (on page 310).
   b. On the `Tenant` tab, select the `Tenant` (on page 311).To build a list of `Tenants` (on page 311) for which active calls are to be reported on:
      i. To add a tenant click `Add`.
      ii. To remove a tenant, select it and click `Remove`.
      iii. To remove all tenants, select `Remove All`.
3. Click `Generate Report`. Continue from Step 8 on page 302.

End of procedure

- To view the matching Tenant Call Arrivals data, select the `Historical Arrivals` link from the `Related Reports` section of the `Tasks` panel.

Note: You can select up to eight components at a time.

The Peaks Volume, which is shown on the graph, counts the peak number of calls that is observed during the specified time range, according to the selected granularity level.

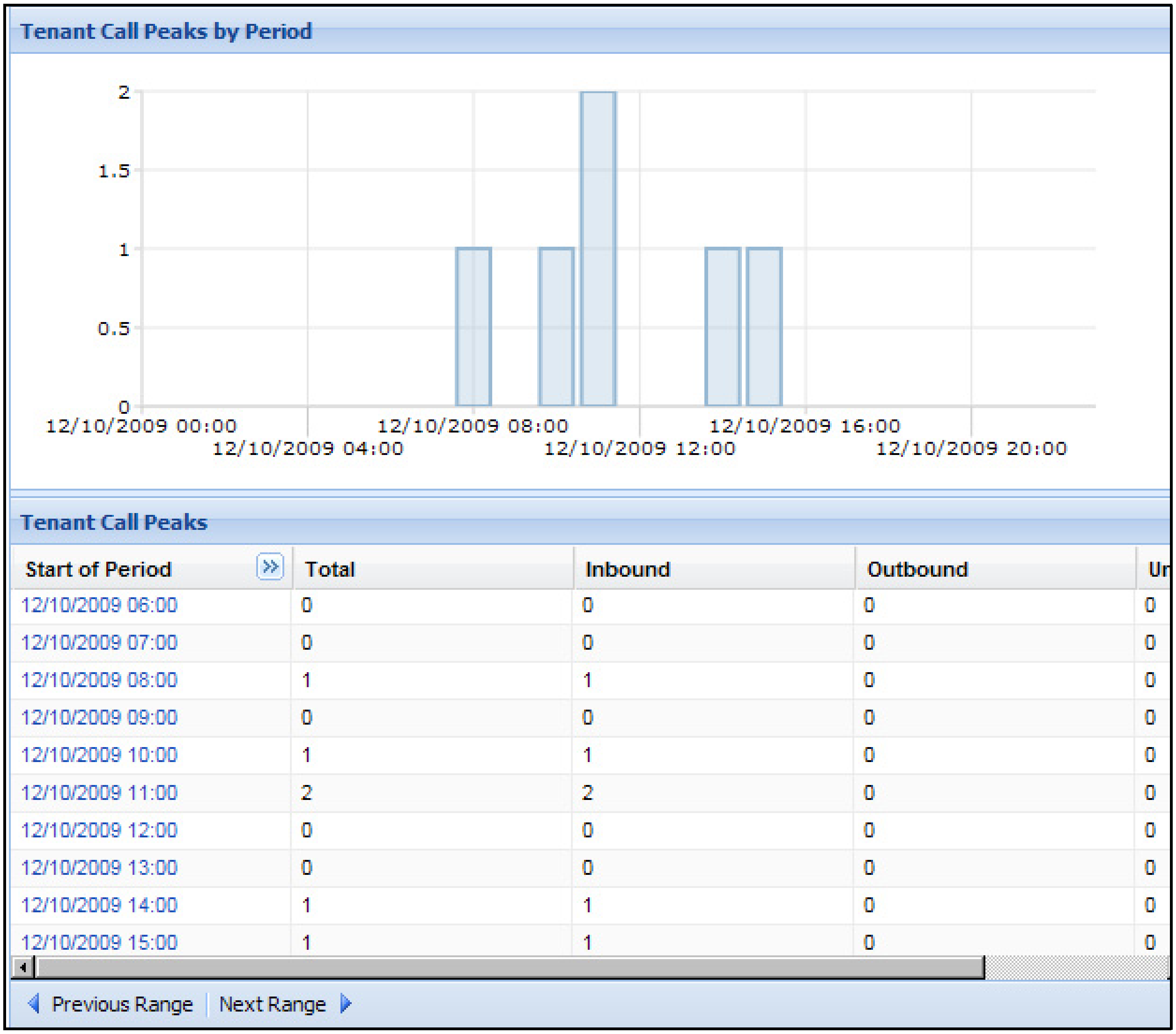


Figure 30: Tenant Call Peaks Report

Table 58 describes the fields for the Tenant Call Peaks report.

Table 58: Tenant Call Peaks Report Fields

Field	Description
Start of Period	The date and time of the call for the given granularity period. Click >> to view the `End of Period` column.
Total	The peak number of concurrent calls reached during the time period.
Inbound	The total number of inbound calls for the given peak period.

Table 58: Tenant Call Peaks Report Fields (Continued)

Field	Description
Outbound	The total number of outbound calls for the given peak period.
Unknown	The total number of unknown calls for the given peak period.

Media Service Call Peaks

The Genesys Media Server (GMS) collects, summarizes and reports on concurrent session peaks for each media service requested by SIP Server over MSML.

For each of the services in Table 59, the GMS provides peaks (simultaneous/concurrent sessions) with a timestamp of when the peak occurred, and for each tenant if in a multi-tenant environment. The time period for each peak can be hours or a finer granularity (see Time Period in "Report Filters" on page 310).

Table 59: Media Service Call Peak Reporting

Media Service	Report Details
GMS Treatments	Peak Daily Use of `Treatment` sessions and peak timestamp. This may be used as the daily maximum call parking use of GMS (excluding Play Application use of VoiceXML).
GMS CPD	Peak Daily Use of `CPD` sessions and peak timestamp. Call Progress Detection Peaks can be used for outbound calling.
GMS Record	Peak Daily Use of `Recording` sessions and peak timestamp. This may be useful for CRQM billing or at least loading.
GMS Conference	Peak Daily Use of `Conferencing` sessions and peak timestamp. This may be useful for loading statistics of the system.
GMS Media	Peak Daily Use of `Media` sessions and peak timestamp. This is for completeness where the service is not categorized.

The reference interval is daily peak with a timestamp of the peak.

These peaks are available from Reporting Server, by tenant for reporting in Genesys Administrator and for Genesys License Reporting Manager (LRM) access via the Reporting Server web services API.

This report can be generated for specific tenants, or for the entire RS deployment.

- See “Generating a Report with GA” on page 298 for the procedure steps.
- See “GAX Report Generation Table” on page 304 for the specific choices to make, to generate this report (and others).

MCP VXML Call Arrivals

The MCP Call Arrivals report provides a summary of MCP session arrivals using a GVP VoiceXML application (SIP Invite) during a given period of time. Play Application (VoiceXML) will be monitored and counted as a VoiceXML session. So VoiceXML session arrivals are total use count of VoiceXML irrespective of how it was requested. The time period for each arrival count can be hour or higher granularity (see Time Period in “Report Filters” on page 310).

This report can be generated for specific tenants, IVR profiles, or MCP components, or for the entire deployment.

- See “Generating a Report with GA” on page 298 for the procedure steps.
- See “GAX Report Generation Table” on page 304 for the specific choices to make, to generate this report (and others)

MCP VXML Call Peaks

The MCP Call Peaks report provides the peak volume of calls using a GVP VoiceXML application (SIP Invite) during a given period of time. Play Application (VoiceXML) will be monitored and counted as a VoiceXML session. So VoiceXML session peaks are the simultaneous total use of VoiceXML irrespective of how it was requested. The time period for each peak can be hours or a finer granularity.

This report can be generated for specific tenants, IVR profiles, or MCP components, or for the entire deployment.

- See “Generating a Report with GA” on page 298 for the procedure steps.

- See "GAX Report Generation Table" on page 304 for the specific choices to make, to generate this report (and others).

Note: MCP CDR data is used to calculate the VXML Call Peaks. CDR data that arrives late may be incorporated into hourly VXML peak statistics, but a late CDR is not incorporated if it arrives late by more than the operations counts retention parameter (rs.db.retention.operations.counts.default).

ASR/TTS Usage

The ASR/TTS Usage report provides the overall ASR and TTS usage on calls on a per-component (ASR or TTS Server), IVR Profile, Tenant, and deployment basis.

Procedure:Generating the ASR/TTS Usage Report

Purpose: To generate the ASR/TTS Usage report using Genesys Administrator Extension.

Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instruction:
   a. On the `Time Range` tab, select the appropriate `Time Range` (on page 310) and the `Granularity` (on page 310).
   b. On the `Components` tab, select the `Component Type` (on page 311).To build a list of `Components` (on page 311) for which active calls are to be reported on:
      i. To add a component, click `Add`.
      ii. To remove a component, select it and click `Remove`.
      iii. To remove all components, select `Remove All`.
3. Click `Generate Report`. Continue from Step 8 on page 302.

End of procedure

Table 60 describes the fields for the ASR/TTS Usage report.

Table 60: ASR/TTS Usage Report Fields

Field	Description
Start of Period	The date and time of the call for the given granularity period. Click >> to view the End of Period column.
Session	The number of ASR or TTS sessions that are used during a time period for a specific ASR Server or TTS Server, IVR Profile, or tenant.
Session Count	The total number of inbound, outbound, and unknown calls for the given peak period.

ASR/TTS Usage Peaks

The ASR/TTS Usage Peaks report provides the peak ASR and TTS usage on calls on a per-component (ASR or TTS Server), IVR Profile, or Tenant basis.

Note: GVP does not provide usage peaks for MRCPv2;
GVP does provide usage peaks for MRCPv1.

Procedure:Generating the ASR/TTS Usage Peaks Report

Purpose: To generate the ASR/TTS Usage Peaks report using Genesys Administrator.

Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instruction:
   a. On the `Time Range` tab, select the appropriate `Time Range` (on page 310) and the `Granularity` (on page 310).
   b. On the `Components` tab, select the `Component Type` (on page 311).To build a list of `Components` (on page 311) for which active calls are to be reported on:
      i. To add a component, click `Add`.
      ii. To remove a component, select it and click `Remove`.
      iii. To remove all components, select `Remove All`.

3. Click `Generate Report`. Continue from Step 8 on page 302.

End of procedure

You can select and modify the list of IVR Profiles and Tenants in the same way as you did the Components, by clicking on their respective tabs. The Resource Type drop-down menu is not displayed on the IVR Profiles or Tenants tab.

Table 61 describes the fields for the ASR/TTS Usage Peaks report.

Table 61: ASR/TTS Usage Peaks Report Fields

Field	Description
Start of Period	The date and time of the call for the given granularity period. Click `>>` to view the `End of Period` column.
Session	The peak number of concurrent ASR or TTS sessions that are used during a time period for a specific ASR Server or TTS Server, IVR Profile or Tenant.
Session Count	The total number of inbound, outbound, and unknown calls for the given peak period.

Media Services Usage and GVP Ports Peaks

Resource Manager (RM) and Media Control Platform (MCP), along with SIP Server, form a Genesys Media Server which is part of the Genesys SIP Server offering. These same components are partly found in GVP. Consequently, this section applies to both environments on the GVP platform. When routing requests the use of a media service via MSML, the RM/MCP can track the peak simultaneous number of sessions used by a Tenant of that service. A Tenant can have a single IVR Profile defined for each service type. If the system is used without multiple Tenants, then the single Tenant is the Environmental Tenant.

The peak use of a given service (see “Media Service Call Peaks” on page 355) may occur at a different time than another service. The peak uses are of value for capacity planning purposes, and in some cases, for billing purposes (for example, Pay Per Use).

The Services are as follows:

Peak Daily Use of Treatment sessions and peak timestamp

This may be used as the daily maximum call parking and call qualification use of media services. As this can be a separate saleable and billable item with

Genesys (unless you have GVP ports), it will eventually tracked by the Genesys License Reporting Manager (LRM) product. Hence this statistic may be used for pay per use (PPU) tracking by Genesys for customer billing contracts, via the LRM server. This value is also useful for system capacity planning. LRM tracks this peak value.

Peak Daily Use of CPD sessions and peak timestamp

Call Progress Detection Peaks might be used for monitoring loads for outbound call progress detection. This peak tracks the number of simultaneous sessions doing Call Progress Detection (CPD) using the MCP's CPD capability. Currently this value is provided for system capacity planning.

Peak Daily Use of Recording sessions and peak timestamp

This peak tracks the number of simultaneous sessions doing recording by a given tenant. Note that a given call may have many recording legs—caller, agent, supervisor—such that the recording sessions level will exceed the number of calls in the system if most calls record. Currently this value is provided for system capacity planning.

Peak Daily Use of Conferencing sessions and peak timestamp

This peak may be useful for capacity planning and knowing how much load the system is receiving.

Peak Daily Use of Media sessions and peak timestamp

This peak is for completeness where the service is not categorized, and can contain special media functions outside of GVP type deployments.

Special Service: Peak Daily Use of VoiceXML = GVP Ports

If a call arrives using a GVP VoiceXML application (SIP Invite), the VoiceXML session is included in the simultaneous peak use of VoiceXML. Peak Daily Use of GVP VoiceXML sessions (see “MCP VXML Call Peaks” on page 356) includes the use of MSML sessions initiated for a PLAY APPLICATION (VoiceXML) request and not counted as a Treatment session. Play Application (VoiceXML) is monitored and counted as a VoiceXML session. So VoiceXML session peaks are the simultaneous total use of VoiceXML irrespective of how it was requested. This peak is tracked and measured by LRM using the Reporting Server. It can be a contributing metric value for Pay Per Use (PPU) for GVP ports.

Daily Peak and Timestamp

The peak value reached on any given day, for any service, are tracked by the GVP Reporting Server and the data is displayed in the GAX reporting tool.

LRM and PPU

The daily peak values for media services and VoiceXML sessions have a timestamp for use by LRM. LRM may collect this daily peak, and after a month of such collections, produce a final monthly peak statistic Pay Per Use statistic for Genesys entitlement tracking.

At this time, LRM collects only VoiceXML session and Treatment Session peaks.

Capacity Planning and Resource Allocations

Based on the media service requests for all service types including VoiceXML sessions, different services may demand more resources and processing than others. By examining the various service loadings and using the GVP and Media portion of the Genesys Hardware Sizing Guide, an operational manager of media services can determine if more MCPs are required to handle a particular load or all loads.

Important Play Application Limitations

Some call center architects might prefer to use routing to control the interaction with a caller while still using a VoiceXML application through the Play Application media service. Several important limitations should be noted:

- GVP ports are required.
- The Play Applications will be reported as individual events for the tenant, not as one whole call. That is, if the tenant has several applications in routing using different VoiceXML snippets, they will all be reported simply in GVP historical reporting as short tenant VoiceXML sessions.
- Routing plus Play Application is NOT a substitute for GVP applications where session preservation is important: for example, you can not use speech resources more than once per call, and hence only once in a Play Application per call.
- If multiple Play Applications are used where back end communication exists (web services), there will be NO cookie preservation or data persistence between Play Application requests. This is particularly true for CTI integrations with third party routing systems; GVP, not Play Application, should be used.

Best Practices use of Play Application is to use ONE Play Application VoiceXML for conducting a series of interactions with a caller, including back end and speech resources, and when that interaction sequence is done, return the data and call control to routing.

Historical Call Browser

The Historical Call Browser report (see Figure 31) displays a list of completed calls. It provides the ability to search for and browse call detail records. These records represent calls that either completed successfully or eventually were timed out by the Reporting Server.

A call is considered as timed out if the call processing component has unexpectedly shut down, or if the connection between the call processing component and the Reporting Server is broken. In either case, the Reporting Server has not received the update indicating that the call ended. If the connection is down, the Reporting Server will eventually receive the update; however, if the processing component unexpectedly shut down, the call will stay as timed out.

The Reporting Server processes timed out calls once hourly (by default), and marks calls as timed out when they have been in progress for more than a configured period of time (by default 3 hours).

The following procedure describes how to generate the Historical Call Browser report.

Procedure:
Generating the Historical Call Browser Report

Purpose: To generate the Historical Call Browser report using Genesys Administrator.

Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instructions:
   a. On the `Time Range` tab, select the appropriate `Time Range` (on page 310).
   b. On the `Call Info` tab:
      i. Enter the `DID` (on page 311).
      ii. Select the appropriate `Call Type` (on page 311).
      iii. Select the `Call Disposition` (on page 313).
      iv. Enter the minimum and maximum `Call Lengths` (on page 311).
      v. Enter the `Call End State` (on page 313).
      vi. Enter the `Call Result` (on page 313).
      vii. Select the appropriate `ID` (on Page 312), and enter its value.

c. On the `IVR Profile` tab, specific the `IVR Profiles` (on page 311) to include in the report. To modify the list of IVR Profiles, do any of the following:

   i. Too add an IVR Profile, click `Add`. Select the IVR Profiles from the list.

   ii. To remove an IVR Profile from the list, click `Remove`.

   iii. To remove all IVR Profiles the list. Click `Remove All`.

d. On the `Components` tab, select the `Component Type` (on page 311).To build a list of `Components` (on page 311) for which active calls are to be reported on:

   i. To add a component, click `Add`.

   ii. To remove a component, select it and click `Remove`.

   iii. To remove all components, select `Remove All`.

e. On the `Tenants` tab, select the `Tenant` (on page 311).To build a list of `Tenants` (on page 311) for which active calls are to be reported on:

   i. To add a tenant, click `Add`.

   ii. To remove a tenant, select it and click `Remove`.

   iii. To remove all tenants, select `Remove All`.

f. On the `Other` tab, enter the `Virtual Reporting Object`, the `Remote URI`, and/or the `Local URI` (on page 313).

3. Click `Generate Report`. Continue from Step 8 on page 302.

End of procedure

Note: You can select multiple IVR Profiles, Components, and Tenants.

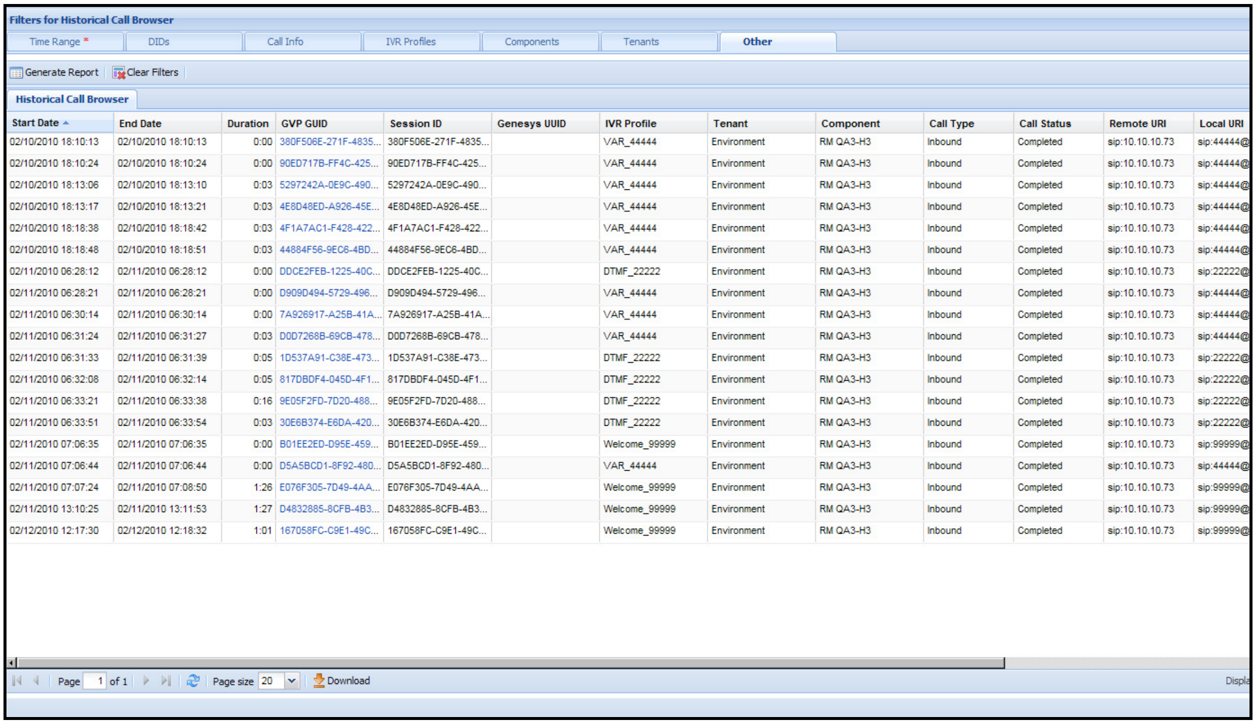


Figure 31: Historical Call Browser Report

Table 62 describes the fields for the Historical Call Browser report.

Table 62: Historical Call Browser Report Fields

Field	Description
Start Date	The start date and time of the call.
End Date	The end date and time of the call.
Duration	The length (in milliseconds) of the call.
GVP GUID	The globally unique ID that identifies a complete interaction with GVP. This ID is generated by the Resource Manager, and is passed to all the resources that provide service for the call. For more information on the GVP Session ID, see "Session Identifiers" on page 26.
Session ID	The GVP Component ID that is generated by the component to identify the call leg. For more information on the GVP Component ID, see "Session Identifiers" on page 26.
Genesys UUID	The Genesys CallUUID that is generated by T-Server or SIP Server. For more information on the Genesys CallUUID, see "Session Identifiers" on page 26.

Table 62: Historical Call Browser Report Fields (Continued)

Field	Description
IVR Profile	The name of the IVR Profile that is selected.
Tenant	The name of the Tenant.
Component	The name of the application (RM, CCP, MCP) that is selected.
Call Type	The type of call. Valid call types are the following: • Inbound • Outbound • Bridged • Unknown • New Call • Createccxml • External
Call Status	The state of the call. Valid call states are the following: • IVR—The call is being processed by MCP. • Accepted—The call has been received by RM, but not landed on the VoiceXML platform or transferred to an agent. • Transferring—The call is being transferred to an agent.
Remote URI	The remote Uniform Resource Identifier.
Local URI	The local Uniform Resource Identifier.
Call Disposition	The outcome of the call. Valid call states are the following: • Unknown—The outcome was not specified by the Resource Manager. • Completed in IVR—The call completed in self service. • Transferred to Agent • Abandoned in Queue—The caller hung up while waiting in the queue. **Note:** The Abandoned in Queue disposition is available only for those call flows that involve IVR Server for CTI. This column is only visible when the Component type is RM.

Table 62: Historical Call Browser Report Fields (Continued)

Field	Description
Queue Wait Time	The amount of time (in milliseconds) that the caller waited in queue. This value is displayed only for those calls with the Transferred to Agent or Abandoned in Queue disposition. **Note:** Because calls can be transferred to an agent without waiting in queue, the Queue Wait Time may display no values (empty) for the Transferred to Agent disposition. This column is only visible when the Component type is RM.
Dialed Number	The number that was dialed. This column is visible only when the Component type is RM.
Site ID (RM only)	The site ID from which the call originated.
Session Start Origin (CCP only)	The source from which the call started.
End Reason (CCP only)	The reason for the ending the call.
Parent Session ID (MCP only)	The unique identifier of the parent session.
SQ (MCP only)	The link to the SQ Call Failure Dashboard for the selected MCP record.
End State (MCP only)	The end state of the call. Valid states are: • Application End—The application hung up. • System Error—The call did not end properly. • Unknown—The MCP did not log an end state. • User End—The caller hung up.
End Result (MCP only)	The end result of the call, as reported by the application. Valid results are: • Success • Failed
Resource Type (MCP only)	The type of resources that were used on the call.

Table 62: Historical Call Browser Report Fields (Continued)

Field	Description
Reason (MCP only)	The reason for the call.
Notes (MCP only)	Any other notes that are associated with the call.
Profile Usage	The number of calls that are in progress for the given IVR Profile. This is recorded when the Resource Manager associates the call with the IVR Profile. The burst level is indicated with a color bar. The colors green, yellow, and red correspond to burst levels 1, 2, and 3 respectively. **Note:** This column is only visible when the Component type is RM.
Tenant Usage	The number of calls that are in progress for the tenant. The burst level is indicated with a color bar. The colors green, yellow, and red correspond to burst levels 1, 2, and 3 respectively. **Note:** This column is visible only when the Component type is RM.

Clicking on a GVP GUID link results in displaying the Historical Call Browser details that breaks down the selected record according to component type. It displays the call detail records for all components that were involved in handling the call. There can be multiple call detail records for each component if there was more than one leg in the call.

Clicking on a Session ID of an MCP call results in displaying the VAR events associated with the call.

Expanding an MCP call row reveals a table of custom VAR variables associated with that call.

Per-Call IVR Actions Report

Within the Historical Call Browser you can generate the Per-Call IVR Action Report. The Per-Call IVR Actions Report (see Figures 32 and 33) lists the VAR actions and custom variables handled during a single Media Control Platform session. The following procedure describes how to generate the Per-Call IVR Actions Report.

The following procedure describes how to generate the Per-Call IVR Actions Report.

Procedure:
Generating the Per-Call IVR Actions Report

Purpose: To generate the Per-Call IVR Actions Report using Genesys Administrator.

Summary

The Per-Call IVR Actions Report displays Media Control Platform data only.

Start of procedure

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instructions:
   a. On the `Time Range` tab, select the appropriate `Time Range` (on page 310) or a predefined range and the `Granularity` (on page 310).
3. On the `Component` tab, select the Media Control Platform for which you require data.
4. Click `Generate Report`.
5. In the `IVR Action` column, click the arrows in the row for which you require data.

   The reporting data is displayed.

End of procedure

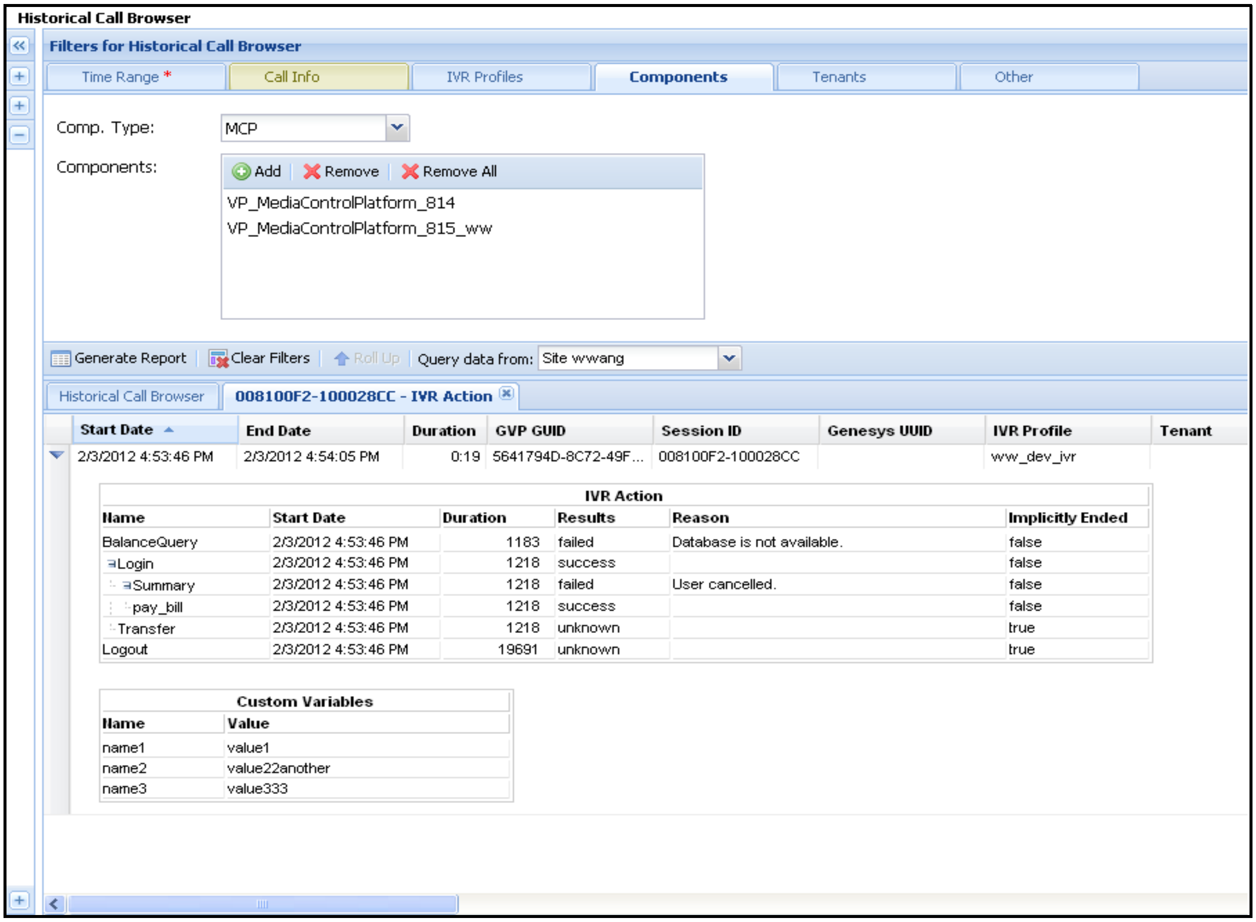


Figure 32: Per-Call IVR Actions Report

The report also provides usage information in the CDR. See Figure 33.

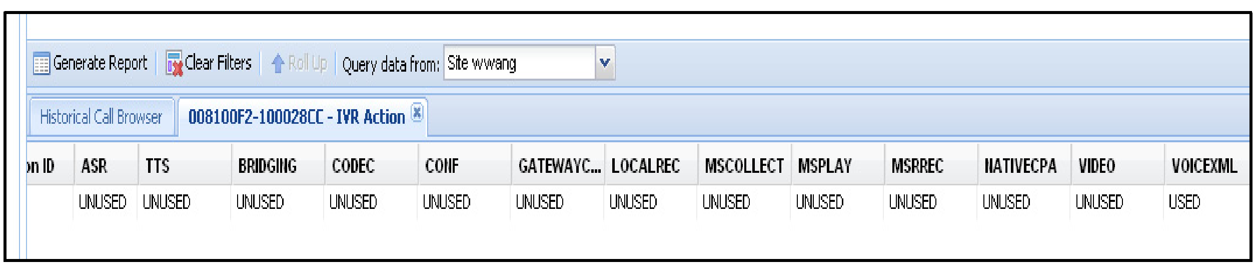


Figure 33: Per-Call IVR Actions Report —Usage Metrics

Table 63 describes the fields for the Per-Call IVR Actions Report.

Table 63: Per-Call IVR Actions Report Fields

Field	Description
ASR	If ASR is used at any point during the call.
TTS	If TTS is used at any point during the call.

Table 63: Per-Call IVR Actions Report Fields (Continued)

Field	Description
LOCALREC	If a local recording was executed during the call.
MSRREC	If a media stream replication recording was executed during the call.
CONF	If conferencing was established during the call.
BRIDGING	If bridging was established during the call.
VIDEO	If a video connection was established during the call.
CODEC	If any transcoding was used for this call (any leg).
VOICEXML	If VoiceXML was used during the call.
NATIVECPA	If native media server CPD/CPA was used during the call.
GATEWAYCPA	If gateway-based CPD/CPA was used during this call.
MSPLAY	If a request for MSML `<play>` was made during the call.
MSCOLLECT	If a request for MSML `<collect>` or `<dtmf>` was made during the call.

Chapter

20 Service Quality Reports

This chapter describes the available reports Service Quality data. It contains the following sections:

Overview

The Service Quality tool measures system performance, based on service quality metrics that impact the caller experience, and includes alarm generation and detailed reporting. The Service Quality reports display statistics of service quality metrics and time measurement for specific system tasks. Service Quality data is for Media Control Platform components only.

There are performance implications to changing the batch size. For more information, see "Configuring the Reporting Server" on page 275.

SQ Call Failures

Failed Calls

A single call can either be a success or a failure from the perspective of Service Quality, and GVP uses this type of failure to calculate SQ % (the percentage of calls that are successful). Consider this type, a failed call.

Quality Failures

Regardless of whether a single call is a failure or a success, multiple non-fatal failures may occur during that single call. Consider these *quality failures*. GVP uses quality failures to track total failures (a number which can exceed the total number of calls).

- Failure time represents the exact time that a failure occurred.

- The order column in an SQ Call Failures report represents the order in which call failures (within a single call) occurred. If a specific call has 10 call failures, then each failure will have an order value somewhere between 1 and 10. Therefore, order is the relationship between multiple failures within a single call.

The SQ Call Failures report (see Figure 34) displays the list of calls that have failed service quality.

The following procedure describes how to generate the SQ Call Failures report.

Procedure: Generating the SQ Call Failures Report with GA

Note: In GAX, see the onscreen help about filtering and configuration.

Purpose: To generate the SQ Call Failures report by using Genesys Administrator.

Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298.
2. For Step 7 in those instructions:
   a. On the `Time Range` tab or in the corresponding start/end fields, select the appropriate `Time Range` (on page 310).
   b. On the `Call Info` tab, if required select the appropriate `Call Failure Type` (on page 312).
   c. On the `IVR Profile` tab, if required specify the `IVR Profiles` (on page 311) to include in the report. To modify the list of IVR Profiles, do any of the following:
      i. Too add an IVR Profile, click `Add`. Select the IVR Profiles from the list.
      ii. To remove an IVR Profile from the list, click `Remove`.
      iii. To remove all IVR Profiles the list. Click `Remove All`.
   d. On the `Components` tab, if required select the `Component Type` (on page 311).To build a list of `Components` (on page 311) for which SQ failures are to be reported on:
      i. To add a component, click `Add`.
      ii. To remove a component, select it and click `Remove`.

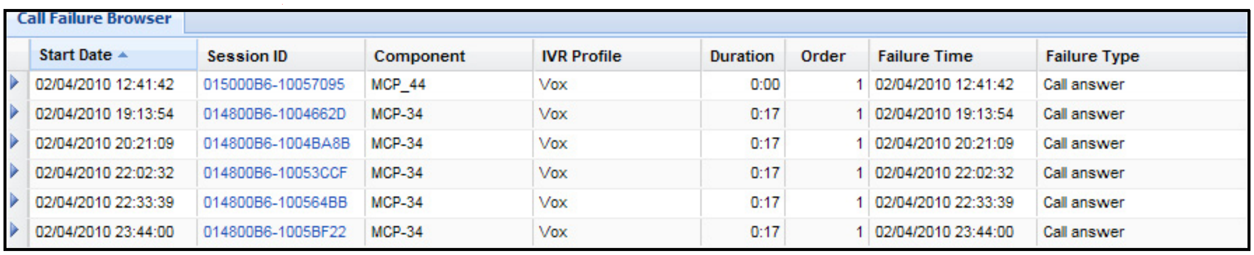
iii. To remove all components, select `Remove All`.

Note: You do not need to select Call Failure Type, IVR Profiles, or Components in order to retrieve data.

3. Click `Generate Report`. Continue from Step 8 on page 302.

   Click the specific `Session ID` link to display the historical call data for that record.

End of procedure



Call Failure Browser

Start Date	Session ID	Component	IVR Profile	Duration	Order	Failure Time	Failure Type
02/04/2010 12:41:42	015000B6-10057095	MCP_44	Vox	0:00	1	02/04/2010 12:41:42	Call answer
02/04/2010 19:13:54	014800B6-1004662D	MCP-34	Vox	0:17	1	02/04/2010 19:13:54	Call answer
02/04/2010 20:21:09	014800B6-1004BA8B	MCP-34	Vox	0:17	1	02/04/2010 20:21:09	Call answer
02/04/2010 22:02:32	014800B6-10053CCF	MCP-34	Vox	0:17	1	02/04/2010 22:02:32	Call answer
02/04/2010 22:33:39	014800B6-100564BB	MCP-34	Vox	0:17	1	02/04/2010 22:33:39	Call answer
02/04/2010 23:44:00	014800B6-1005BF22	MCP-34	Vox	0:17	1	02/04/2010 23:44:00	Call answer

Figure 34: SQ Call Failures Report

Table 64 describes the fields for the SQ Call Failures report.

Table 64: SQ Call Failures Report Fields

Field	Description
Start Date	The start time of the call that failed to meet a defined minimum latency standard (service quality).
Session ID	The Session ID of the call that failed to meet a defined minimum latency standard (service quality).
Component	The Component of the call that failed to meet a defined minimum latency standard (service quality).
IVR Profile	The IVR Profile of the call that failed to meet a defined minimum latency standard (service quality).
Duration	The duration of the call that failed to meet a defined minimum latency standard (service quality).
Order	If there is more than one failure for the given call, this is the chronological order of the failure, starting with the number 1.
Failure Time	The time the call that failed to meet a defined minimum latency standard (service quality).

Table 64: SQ Call Failures Report Fields (Continued)

Field	Description
Failure Type	The reason for the call that failed to meet a defined minimum latency standard (service quality).

SQ Failure Summary

The SQ Failure Summary report (see Figure 35) provides a graphical display of calls that failed to meet a defined minimum latency standard for sessions that have ended in each service quality period.

The following procedure describes how to generate the SQ Failure Summary report.

Procedure: Generating the SQ Failure Summary Report

Purpose: To generate the SQ Failure Summary report by using Genesys Administrator.

Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instructions:
   a. On the `Time Range` tab, select the appropriate `Time Range` (on page 310), and granularity
   b. On the `IVR Profile` tab, specify the `IVR Profiles` (on page 311) to include in the report. To modify the list of IVR Profiles, do any of the following:
      i. Too add an IVR Profile, click `Add`. Select the IVR Profiles from the list.
      ii. To remove an IVR Profile from the list, click `Remove`.
      iii. To remove all IVR Profiles the list. Click `Remove All`.
   c. On the `Components` tab, build a list of `Components` (on page 311) for which service quality failures are to be reported on:
      i. To add a component, click `Add`.
      ii. To remove a component, select it and click `Remove`.
      iii. To remove all components, select `Remove All`.

d. On the `Tenants` tab, build a list of `tenants` (on page 311) for which service quality failures are to be reported on:

   i. To add a component, click `Add`.

   ii. To remove a component, select it and click `Remove`.

   iii. To remove all components, select `Remove All`.

3. Click `Generate Report`. Continue from Step 8 on page 302.

   Select the hyperlinked value in any column to view a detailed histogram of that column.

End of procedure

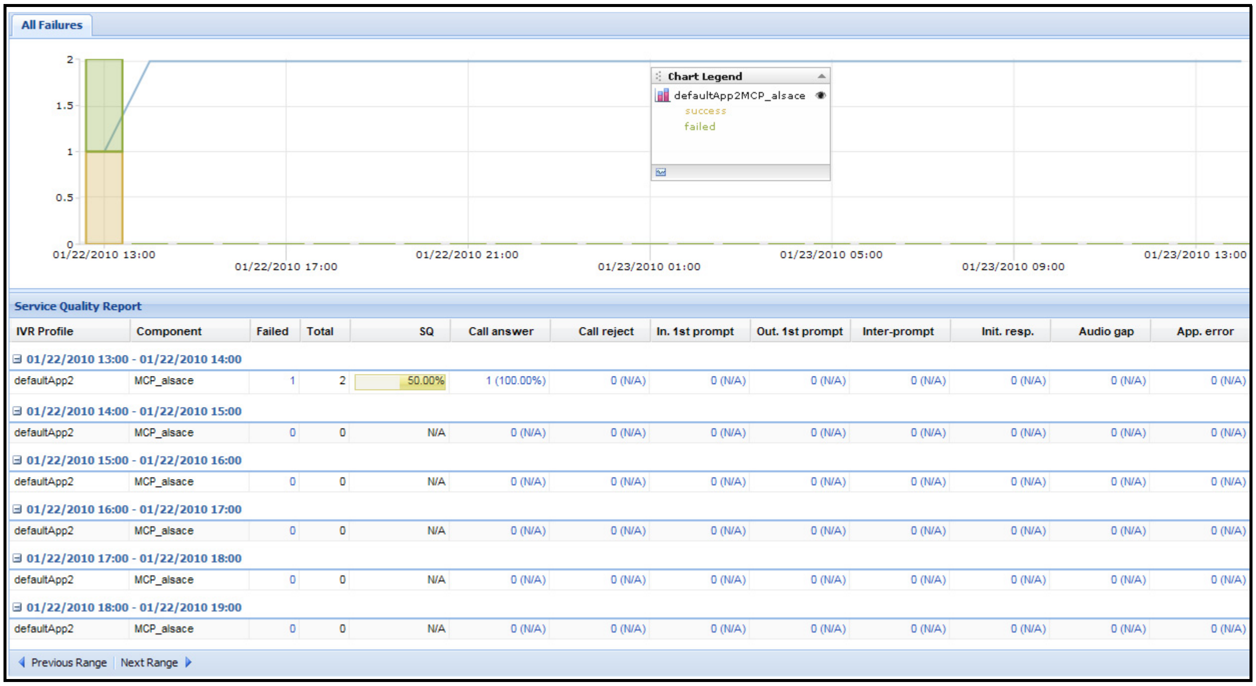


Figure 35: SQ Failure Summary Report

Table 65 describes the fields for the SQ Failure Summary report.

Table 65: SQ Failure Summary Report Fields

Field	Description
IVR Profile	The name of the IVR Profile.
Component	The name of the Component.
Failed	The total number of calls received that failed a threshold.
Total	The total number of calls received.
SQ	The service quality percentage.

Table 65: SQ Failure Summary Report Fields (Continued)

Field	Description
Call answer	The total number of calls answered.
Call reject	The total number of calls rejected.
In. 1st prompt	The total number of calls that failed in the first inbound prompt threshold.
Out. 1st prompt	The total number of calls that failed in the first outbound prompt threshold.
Inter-prompt	The total number of calls that failed in the inter prompt threshold.
Init resp	The total number of calls that failed at the initial response threshold.
Audio gap	The total number of calls that failed in the audio gap threshold.
App.error	The total number of calls that failed because of an application error.
Sys. error	The total number of calls that failed because of a system error.

SQ Latency Summary

The SQ Latency Summary report (see Figure 36) displays the number of calls that fell below a threshold for sessions that have ended in each service quality period. The latency thresholds are configured in the Media Control Platform application under the `ems` section. For more information on these parameters, see “Service Quality Analysis (SQA)” on page 65.

The following procedure describes how to generate the SQ Latency Summary report.

Procedure:
Generating the SQ Latency Summary Report

Purpose: To generate the SQ Latency Summary report by using Genesys Administrator.

Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instructions:
   a. On the `Time Range` tab, select the appropriate `Time Range` (on page 310), and granularity.
   b. On the `Components` tab, select the `Component` (on page 311).

   Note: Time range and component are mandatory parameters.

3. Click `Generate Report`. Continue from Step 8 on page 302.

   Click the double arrow (>>) in each column heading to see the number of latencies per type.

   Select the hyperlinked number to display a histogram of latency threshold for each latency type.

   For information on near-real-time data for SQ Latency statistics, see "SQ Latency Dashboard" on page 323.

End of procedure

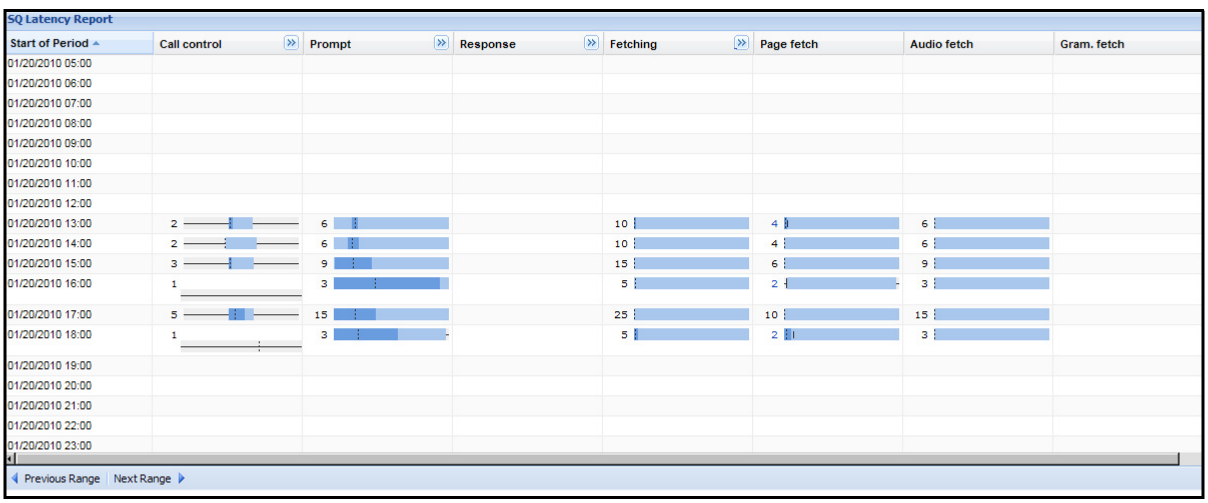


Figure 36: SQ Latency Summary Report

Table 66 describes the fields for the SQ Latency Summary report.

Table 66: SQ Latency Summary Report Fields

Field	Description
Start of Period	The start time of the reporting period.

Table 66: SQ Latency Summary Report Fields (Continued)

Field	Description
Call Control	The number of calls that fell below the Call Control threshold. The Call Control thresholds are: • `CALL_ANSWER` • `CALL_REJECT`
Prompt	The number of calls the fell below the Prompt threshold. The Prompt thresholds are: • `INBOUND_FIRST_PRMOPT` • `OUTBOUND_FIRST_PROMPT` • `INTERPROMPT` • `INITIAL_RESPONSE`
Response	The number of calls that fell below the Response threshold. The Response thresholds are: • `CUMULATIVE_RESPONSE` • `DTMF_INPUT_RESPONSE` • `ASR_INPUT_RESPONSE` • `NOINPUT_RESPONSE` • `RECORDING_RESPONSE` • `TRANSFER_RESPONSE`
Fetching	The number of calls that fell below the Fetching threshold. The Fetching thresholds are: • `PAGE_FETCH` • `AUDIO_FETCH` • `GRAMMAR_FETCH` • `DATA_FETCH` • `JAVA_SCRIPT_FETCH`
Execution	The number of delayed calls because of the following execution reasons: • `PAGE_COMPILE` • `JAVA_SCRIPT_EXECUTION`

Table 66: SQ Latency Summary Report Fields (Continued)

Field	Description
MRCP	The number of delayed calls because of the following MRCP reasons: • MRCP_ASR_SESSION_ESTABLISH • MRCP_TTS_SESSION_ESTABLISH • MRCP_ASR_SET_PARAMS • MRCP_TTS_SET_PARAMS • MRCP_ASR_STOP • MRCP_TTS_STOP • MRCP_DEFINE_GRAMMAR • MRCP_RECOGNIZE • MRCP_SPEAK

- The number next to the chart is the average latency.
- The light blue bar marks the minimum and maximum latency readings.
- The dashed line marks the average.
- The dark blue area around the average marks the standard deviation (1 standard to each side of the dashed line).
- The solid short line marks the nth percentile based on the Reporting Server latency parameters (if an estimate exists).
- Hovering over the chart displays a tooltip that shows the numeric values depicted in the chart.

Genesys

Chapter

21 Voice Application Reports

This chapter describes the Voice Application Reports. It contains the following sections:

Overview

The Voice Application Reports display the usability data for applications that have been divided into logical transactions using the VoiceXML `<log>` tag.

For more information on Genesys Voice Platform specific log extensions, see the *Genesys Voice Platform 8.1 Genesys VoiceXML 2.1 Help* file.

VAR Call Completion Summary

The VAR Call Completion Summary report (see Figure 37) displays the relative frequency with which calls to a given VoiceXML application end in different states.

The following procedure describes how to generate the VAR Call Completion Summary report.

Procedure:
Generating the VAR Call Completion Summary Report with GA

Note: In GAX, see the onscreen help about filtering and configuration.

Purpose: To generate the VAR Call Completion Summary report using Genesys Administrator.

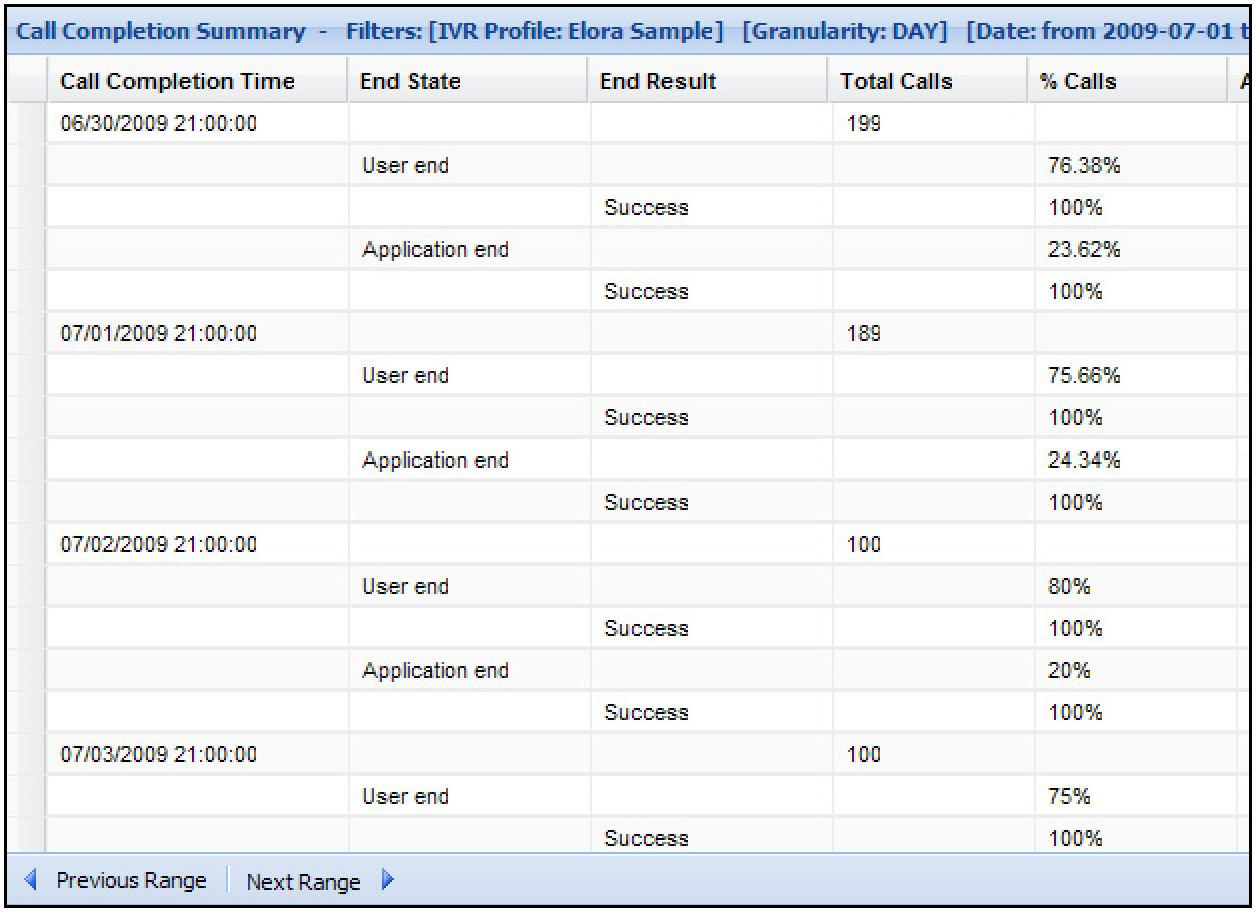
Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instructions:
   a. On the `Time Range` tab, select the appropriate `Time Range` (on page 310) and the `Granularity` (on page 310).
   b. On the `IVR Profile` tab, select the `IVR Profile` (on page 311).
   c. On the `Tenants` tab, select the `Tenant` (on page 311).
3. Click `Generate Report`. Continue from Step 8 on page 302.

End of procedure

To view the matching VAR IVR Action Usage data, select the `VAR IVR Action Usage` link from the `Related Reports` section of the `Tasks` panel.

Note: The VAR Call Completion Summary report focuses on VoiceXML applications using VAR; therefore, it displays MCP data only.



Call Completion Summary - Filters: [IVR Profile: Elora Sample] [Granularity: DAY] [Date: from 2009-07-01 t

Call Completion Time	End State	End Result	Total Calls	% Calls	A
06/30/2009 21:00:00			199		
	User end			76.38%	
		Success		100%	
	Application end			23.62%	
		Success		100%	
07/01/2009 21:00:00			189		
	User end			75.66%	
		Success		100%	
	Application end			24.34%	
		Success		100%	
07/02/2009 21:00:00			100		
	User end			80%	
		Success		100%	
	Application end			20%	
		Success		100%	
07/03/2009 21:00:00			100		
	User end			75%	
		Success		100%	

Previous Range Next Range

Figure 37: VAR Call Completion Summary Report

Table 67 describes the fields for the VAR Call Completion Summary report.

Table 67: Call Completion Summary Report Fields

Field	Description
Call Completion Time	The date and time (in yyyy-mm-dd hh:mm:ss format) when the call finished.
End State	The end state of the call. Valid states are: • Application End—The application hung up. • System Error—The call did not end properly. • Unknown—The MCP did not log an end state. • User End—The caller hung up.

Table 67: Call Completion Summary Report Fields (Continued)

Field	Description
End Result	The end result of the call, as reported by the application. Valid results are: • Success • Failed • Unknown—The call end result is not specified by the application that is using the VAR `<log>` interface. For more information on VoiceXML <log> extensions, see “VoiceXML <log> Extensions” on page 501.
Total Calls	The total number of calls that ended for the time duration (granularity) that is selected.
% Calls	The percentage of calls that ended with the particular combination of result and reason.
Avg. Call Len. (sec)	The average length, in seconds, of the call.

VAR IVR Action Summary

The VAR IVR Action Summary report (see Figure 38) displays statistics on individual IVR Actions that are within the `<log>` tag in a VoiceXML application. For more information on the `<log>` extension, see “VAR Metrics” on page 454.

The following procedure describes how to generate the VAR IVR Action Summary report.

Procedure:
Generating the VAR IVR Action Summary Report

Purpose: To generate the VAR IVR Action Summary report using Genesys Administrator.

Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instructions:
   a. On the `Time Range` tab, select the appropriate `Time Range` (on page 310) and the `Granularity` (on page 310).

b. On the `IVR Profile` tab, specify the `IVR Profiles` (on page 311) to include in the report. To modify the list of IVR Profiles, do any of the following:

   i. Too add an IVR Profile, click `Add`. Select the IVR Profiles from the list.

   ii. To remove an IVR Profile from the list, click `Remove`.

   iii. To remove all IVR Profiles the list. Click `Remove All`.

c. On the `Tenants` tab, select the `Tenant` (on page 311).To build a list of `Tenants` (on page 311) for which active calls are to be reported on:

   i. To add a tenant, click `Add`.

   ii. To remove a tenant, select it and click `Remove`.

   iii. To remove all tenants, select `Remove All`.

3. Click `Generate Report`. Continue from Step 8 on page 302.

End of procedure

To view the matching Call Completion Summary data, select the `VAR Call Completion` link from the `Related Reports` section of the `Tasks` panel

Note: The VAR IVR Action Summary report focuses on VoiceXML applications using VAR; therefore, it displays MCP data only.

IVR Action Summary - Filters: [IVR Profile: Elora Sample] [Granularity: DAY] [Date: from 2009-07-01 to 2

IVR Action	Usage Count	% Actions successful
Elora Sample - Total Session Count: 993		
Call_Agent	354766	100%
Get_Balance	828216	100%
Get_Data	355278	100%
Login	712848	49.88%
Make_Transfer	355022	100%
Select_Type	355278	100%
Transfer_Info	355022	100%

Figure 38: VAR IVR Action Summary Report

Table 68 describes the fields of the VAR IVR Action Summary report.

Table 68: VAR IVR Action Summary Report Fields

Field	Description
IVR Profile	The name of the IVR Profile for which these actions occurred.
IVR Action	The name of the IVR action.
Usage Count	The number of times that the IVR action was used.
% Actions successful	The percentage of actions that were successful.
Calls	The number of calls that used this IVR action at least once.
% Calls	The percentage of total calls that used this IVR action at least once.

VAR Last IVR Action

The VAR Last IVR Action report (see Figure 39) displays the details of the last IVR Actions that were used during the end of a call.

The following procedure describes how to generate the VAR Last IVR Action report.

Procedure: Generating the VAR Last IVR Action Report

Purpose: To generate the VAR Last IVR Action report using Genesys Administrator.

Start of procedure:

1. Follow the instructions to generate a report (see Procedure: Generating a Report Using Genesys Administrator, on page 298).
2. For Step 7 in those instructions:
   a. On the `Time Range` tab, select the appropriate `Time Range` (on page 310) and the `Granularity` (on page 310).
   b. On the `IVR Profile` tab, select the `IVR Profile` (on page 311).
   c. On the `Tenants` tab, select the `Tenant` (on page 311).

3. Click `Generate Report`. Continue from Step 8 on page 302.

End of procedure

Note: The VAR Last IVR Action report displays MCP data only.

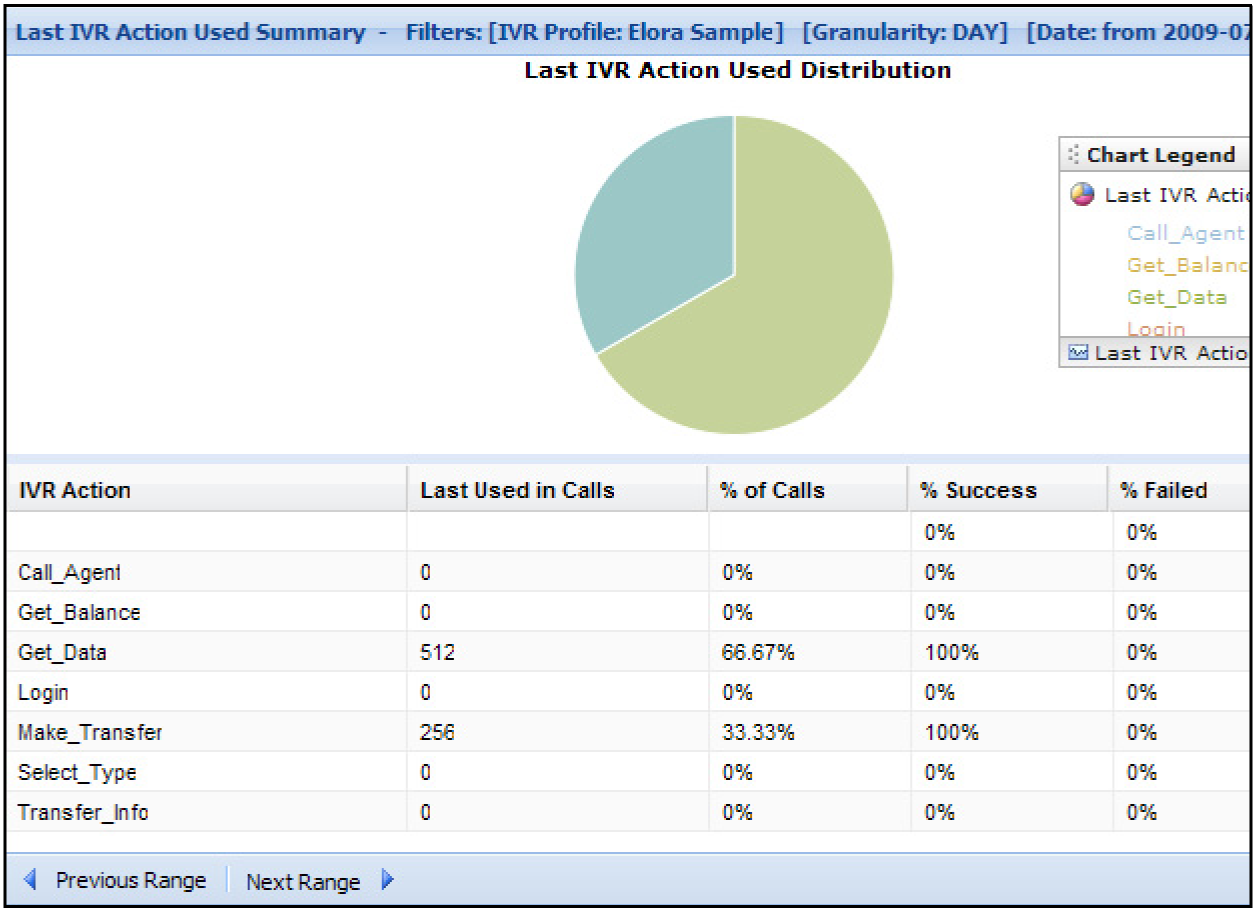


Figure 39: VAR Last IVR Action Report

Table 69 describes the fields for the VAR Last IVR Action report.

Table 69: Last IVR Action Used Report Fields

Field	Description
IVR Action	The name of the IVR Action.
Last Used in Calls	The total number of calls for which the given IVR Action was the last action that was executed.
% of Calls	The percentage of the total number of calls in which this IVR Action was used.

Table 69: Last IVR Action Used Report Fields (Continued)

Field	Description
% Success	The percentage of successful calls in which this IVR Action was used.
% Failed	The percentage of failed calls in which this IVR Action was used.
% Unknown	The percentage of calls that used the Last IVR Action in which the call end result was not specified by the application that is using the VAR `<log>` interface.

Genesys

Part

3 Appendixes

This part of the Guide contains miscellaneous reference information in the following appendixes:

Appendix

A Module and Specifier IDs

This appendix lists various internal Genesys Voice Platform (GVP) identifiers that are required for advanced configuration of EMS Logging and Reporting.

This appendix contains the following sections:

For detailed information about the metrics (application-level logs) that the Media Control Platform (MCP) and the Call Control Platform (CCP) generate, including metric IDs and descriptions, see the *Genesys Voice Platform 8.1 Metrics Reference*.

Media Control Platform

Table 70 lists the Media Control Platform Application Module names and IDs.

For the Next Generation Interpreter (NGI), see “Next Generation Interpreter Module ID and Specifiers” on page 412.

Table 70: Media Control Platform Application Module Names and IDs

Module Name	Description or Comment	Module ID	Specifiers (link)
MTMPC	Media Processing Component (MPC) wrapper	47	MTMPC
LMBase	Base Line Manager	21	LMBase
LMSIP2	SIP Line Manager	40	LMSIP2

Table 70: Media Control Platform Application Module Names and IDs (Continued)

Module Name	Description or Comment	Module ID	Specifiers (link)
SESSMGR	Call Manager API	28	SESSMGR
CALLSESSION		29	None
SMMAIN	Main module in the Media Control Platform	31	SMMAIN
CMUTIL	Media Control Platform utility components	33	CMUTIL
APPMODULE	Base Application Module	34	APPMODULE
REMDIAL	Remote Dial Remdial Application Module	38	REMDIAL
CONFERENCE	Conference Application Module	41	CONFERENCE
SQA	Any and all SQA logs	43	SQA
MEDIAMGR	The Media Manager part of the MPC	176	MEDIAMGR
CONTROL	The control layer part of the MPC	177	CONTROL
MEDIA	The media layer part of the MPC	178	MEDIA
RTP_INTERFACE	The RTP layer of the MPC	179	RTP_INTERFACE
DSP	The DSP components	180	DSP
VGULOGMOD_ MAIN	The main Utility	128	VGULOGMOD_ MAIN
MTINTERNAL	The Internal Media Transport application module	130	MTINTERNAL
RTSPSTACK	The RTSP stack	132	RTSPSTACK
MSML	MSML Implementation	27	MSML
NETANN	NetAnn Implementation	39	NETANN
LMBASE	lmbase	21	LMBASE
VRMMGR		37	None
ADAPTOR	MRCP Client Adaptor	106	ADAPTOR
CLIENT	MRCP V1 Client	97	VRMCLIENT
MRCPV1STACK	MRCP V1 Stack	96	MRCPV1STACK
V2_CLIENT	MRCP V2 Client & VRM Recorder	107	MRCPV2CLIENT

Table 70: Media Control Platform Application Module Names and IDs (Continued)

Module Name	Description or Comment	Module ID	Specifiers (link)
MRCPV2STACK	MRCP V2 Stack	105	MRCPV2STACK
DTMFRECO	DTMF Recognizer	58	DTMFRECO

Table 71 lists the Media Control Platform specifier names and IDs.

Table 71: Media Control Platform Specifier Names and IDs

Specifier ID	Specifier Name
MTMPC	
1001	CMLOGMOD_MTMPC_INITFAILED
2001	CMLOGMOD_MTMPC_CONNERROR
3001	CMLOGMOD_MTMPC_ROUTETOOLONG
LMBase	
1001	CMLOGMOD_LMBASE_IDGENDIRUNACCBLE
1003	CMLOGMOD_LMBASE_SYSIPNOTRETRVABLE
1004	CMLOGMOD_LMBASE_FAILUPDTEOPENCALLIDFILE
1005	CMLOGMOD_LMBASE_NOTPUTSEQNUMTOCALLIDFILE
2001	CMLOGMOD_LMBASE_RESETCALLIDFILECONTNTINVD
3001	CMLOGMOD_LMBASE_NOMEDIASESSPLAYAUDIO
3002	CMLOGMOD_LMBASE_NOMEDIASESSPLAYDTMF
3004	CMLOGMOD_LMBASE_NOMEDIASESSRECRDAUDIO
3005	CMLOGMOD_LMBASE_NOMEDIASESSSTREAMING
LMSIP2	
2001	CMLOGMOD_LMSIP2_RECVUNEXPCTACK
2002	CMLOGMOD_LMSIP2_MEDIAERROR
2003	CMLOGMOD_LMSIP2_ERRSNDINVRESPONSE
2004	CMLOGMOD_LMSIP2_REGISTERTIMEOUT
2005	CMLOGMOD_LMSIP2_REGISTERBADREQUEST

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
2006	CMLOGMOD_LMSIP2_REGISTERFORBIDDEN
2007	CMLOGMOD_LMSIP2_REGISTERNOTFOUND
2008	CMLOGMOD_LMSIP2_REGISTERNOTACCEPTABLE
2009	CMLOGMOD_LMSIP2_REGISTEROTHERERROR
2010	CMLOGMOD_LMSIP2_VGSIPERRORNOTIFY
2011	CMLOGMOD_LMSIP2_ERRPARSESDPCONTENT
2012	CMLOGMOD_LMSIP2_REGISTERALGONOTSUPPORTED
2013	CMLOGMOD_LMSIP2_REGISTERAUTHENTICATIONERROR
2014	CMLOGMOD_LMSIP2_NONMATCHINGSIPINFO
2015	CMLOGMOD_LMSIP2_CUSTOMPARAMERROR
3001	CMLOGMOD_LMSIP2_CANTACCEPTNONINVITECALL
3002	CMLOGMOD_LMSIP2_ERRSNDINVITE
3003	CMLOGMOD_LMSIP2_ERRCREATERTPSESS
3004	CMLOGMOD_LMSIP2_ERRCREATEPSTNSESS
3005	CMLOGMOD_LMSIP2_BADDYNAMICPAYLOAD
3006	CMLOGMOD_LMSIP2_BADDTMFRECV
3007	CMLOGMOD_LMSIP2_ZEROCLOCKRATE
4001	CMLOGMOD_LMSIP2_MESSAGE
4002	CMLOGMOD_LMSIP2_PROCDELAY
SESSMGR	
1001	CMLOGMOD_SESSMGR_IDGENDIRUNACCBLE
1003	CMLOGMOD_SESSMGR_SYSIPNOTRETRVABLE
1004	CMLOGMOD_SESSMGR_FAILUPDTEOPENCALLIDFILE
1005	CMLOGMOD_SESSMGR_NOTPUTSEQNUMTOCALLIDFILE
1007	CMLOGMOD_SESSMGR_VRMINITFAIL

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
1008	CMLOGMOD_SESSMGR_CANTINITLICENSEMGR
2001	CMLOGMOD_SESSMGR_ATTEMPTAUDIOCTRLWBARGEIN
2002	CMLOGMOD_SESSMGR_BADFRMTSCPTAUDIO
2003	CMLOGMOD_SESSMGR_BADFRMTSCPTTTS
2004	CMLOGMOD_SESSMGR_BADFRMTSCPTSTRMNG
2016	CMLOGMOD_SESSMGR_OUTCALLNORESOURCE
2017	CMLOGMOD_SESSMGR_TTSMGRLOST
2018	CMLOGMOD_SESSMGR_TRFRTODESTNOTAUTH
2019	CMLOGMOD_SESSMGR_DESTURINOTSUPP
2020	CMLOGMOD_SESSMGR_DESTURIMALFORMED
2021	CMLOGMOD_SESSMGR_STRMMODUNEXPTEVENT
2022	CMLOGMOD_SESSMGR_LOSTASRMGR
2023	CMLOGMOD_SESSMGR_INITCALLSESSWNOLNMGR
2024	CMLOGMOD_SESSMGR_RESETCALLIDFILECONTNTINVD
2026	CMLOGMOD_SESSMGR_ISDNCAUSECODEERR
3001	CMLOGMOD_SESSMGR_UNEXPECTTTSERROR
3002	CMLOGMOD_SESSMGR_EXPIREASRTTSIGNORED
3003	CMLOGMOD_SESSMGR_UNEXPECTCMCALLBILLEVENT
3005	CMLOGMOD_SESSMGR_FAILMEDSTRMRESLT
3006	CMLOGMOD_SESSMGR_APPMODULENOTFOUND
3007	CMLOGMOD_SESSMGR_UNABLETOSENDLOGTOASR
3008	CMLOGMOD_SESSMGR_INVALIDVRMMESSAGE
4001	CMLOGMOD_SESSMGR_INBOUNDDTMF
4003	CMLOGMOD_SESSMGR_NOINOUTLINES

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
SMMAIN	
1001	CMLOGMOD_SMMAIN_VRMDLLLOADFAIL
1002	CMLOGMOD_SMMAIN_VRMSETLOGFAIL
1003	CMLOGMOD_SMMAIN_MAKEVRMFAIL
1004	CMLOGMOD_SMMAIN_CREATEVRMFAIL
1006	CMLOGMOD_SMMAIN_CALLMGRCFGPARAMERR
1007	CMLOGMOD_SMMAIN_LOADTOOMANYCMGRMOD
1008	CMLOGMOD_SMMAIN_FAILCREATECMGRMOD
1009	CMLOGMOD_SMMAIN_LOADTOOMANYDEVICE
1010	CMLOGMOD_SMMAIN_FAILCREATEDEVICE
1011	CMLOGMOD_SMMAIN_FAILINITDEVICE
1012	CMLOGMOD_SMMAIN_LOADTOOMANYMEDTRPT
1013	CMLOGMOD_SMMAIN_FAILCREATEMEDTRPT
1014	CMLOGMOD_SMMAIN_FAILINITMEDTRPT
1015	CMLOGMOD_SMMAIN_LOADTOOMANYLNMGRS
1016	CMLOGMOD_SMMAIN_FAILCREATELNMGR
1017	CMLOGMOD_SMMAIN_FAILINITLNMGR
1018	CMLOGMOD_SMMAIN_SESSMGRAPPMODCFGERR
1019	CMLOGMOD_SMMAIN_LOADTOOMANYAPPMOD
1020	CMLOGMOD_SMMAIN_SESSMGRMODCFGERR
1021	CMLOGMOD_SMMAIN_LOADTOOMANYSESSMOD
1022	CMLOGMOD_SMMAIN_FAILOPENLICENSE
1023	CMLOGMOD_SMMAIN_FAILPARSELICENSE
1024	CMLOGMOD_SMMAIN_MACVALIDERR
1025	CMLOGMOD_SMMAIN_GENINITLICERR

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
1026	CMLOGMOD_SMMAIN_CANTCREATEVGNETLIB
1027	CMLOGMOD_SMMAIN_CANTINITVGNETLIB
1028	CMLOGMOD_SMMAIN_FAILINITCFGOBJ
1029	CMLOGMOD_SMMAIN_CANTSTARTCMGR
1030	CMLOGMOD_SSMAIN_NOTINBIN
2002	CMLOGMOD_SMMAIN_FAILLOADAPPMODLIB
2003	CMLOGMOD_SMMAIN_FAILINITAPPMOD
2004	CMLOGMOD_SMMAIN_NOVLDAPPMODINLIB
2005	CMLOGMOD_SMMAIN_LIBNODEFMAKEAPPMOD
2006	CMLOGMOD_SMMAIN_VXMLAPPMODNOTLOAD
3001	CMLOGMOD_SMMAIN_FAILSETFDLIMIT
4001	CMLOGMOD_SMMAIN_MCP_STARTED
4002	CMLOGMOD_SMMAIN_MCP_STOPPED
CMUTIL	
2001	CMLOGMOD_CMUTIL_TELNUMLONG
2002	CMLOGMOD_CMUTIL_TELNUMINVCHAR
2003	CMLOGMOD_CMUTIL_POSTDIALLONG
2004	CMLOGMOD_CMUTIL_POSTDIALINVCHAR
2005	CMLOGMOD_CMUTIL_CONFLICTEXT
2006	CMLOGMOD_CMUTIL_HUNTGPINVTRUNK
3001	CMLOGMOD_CMUTIL_HUNTGPNONEXISTTRUNK
3002	CMLOGMOD_CMUTIL_CALLREQNONEXISTHUNTGP
3003	CMLOGMOD_CMUTIL_WAITFORDIAL
3004	CMLOGMOD_CMUTIL_ATTRIBLONG
3005	CMLOGMOD_CMUTIL_VALUELONG

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
APPMODULE	
1001	CMLOGMOD_APPMODULE_FAILSTRTWORKNGTHRD
2001	CMLOGMOD_APPMODULE_FAILREGAPP
2002	CMLOGMOD_APPMODULE_FAILREGAPPMOD
3001	CMLOGMOD_APPMODULE_FAILBINDAPP
REMDIAL	
2001	CMLOGMOD_REMDIAL_FAILREGREMDLMOD
2002	CMLOGMOD_REMDIAL_CANTCREATESERVERSOCK
2003	CMLOGMOD_REMDIAL_SOCKETERROR
3001	CMLOGMOD_REMDIAL_MAXCALLSWARN
3002	CMLOGMOD_REMDIAL_MAXCLIENTS
3003	CMLOGMOD_REMDIAL_NOACTIVESESS
3004	CMLOGMOD_REMDIAL_MAXCALLSREACHED
CONFERENCE	
2001	CMLOGMOD_CONFERENCE_FAILED
2002	CMLOGMOD_CONFERENCE_UNEXPTREASON
4001	CMLOGMOD_CONFERENCE_ESTABLISHED
4002	CMLOGMOD_CONFERENCE_TERMINATED
SQA	
4001	CMLOGMOD_SQA_DTMF
4002	CMLOGMOD_SQA_TRANSFERSTART
4003	CMLOGMOD_SQA_TRANSFEREND
4004	CMLOGMOD_SQA_PROMPTTYPE
4006	CMLOGMOD_SQA_RECOGNITIONSTART
4007	CMLOGMOD_SQA_RECOGNITIONEND

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
4008	CMLOGMOD_SQA_OPENRECORDFILE
4009	CMLOGMOD_SQA_ClOSERECORDFILE
4010	CMLOGMOD_SQA_MEDIAROUTING
4011	CMLOGMOD_SQA_AUDIOGAP
4012	CMLOGMOD_SQA_FIRSTAUDIOPK
4013	CMLOGMOD_SQA_LASTAUDIOPK
427820	CMLOGMOD_SQA_ECMASCRIPT_TIMINGS
427801	CMLOGMOD_SQA_COMPILE_TIME
427802	CMLOGMOD_SQA_FETCH_TIME
MEDIAMGR	
2001	MPCLOGMOD_MEDIAMGR_INVALIDMEDIA
2002	MPCLOGMOD_MEDIAMGR_UNEXPECTEDRTSPDISC
2003	MPCLOGMOD_MEDIAMGR_RTSPREQFAIL
2004	MPCLOGMOD_MEDIAMGR_RTSPREPLYERROR
2005	MPCLOGMOD_MEDIAMGR_RTSPRTPERROR
2006	MPCLOGMOD_MEDIAMGR_UNSUPPORTEDVIDFMT
2008	MPCLOGMOD_MEDIAMGR_UNSUPPORTEDAUDCHNLS
2009	MPCLOGMOD_MEDIAMGR_BADAVICHNKSIZE
2010	MPCLOGMOD_MEDIAMGR_MALFORMEDAVIHDR
2011	MPCLOGMOD_MEDIAMGR_RECBUFFISOTOOSMALL
2012	MPCLOGMOD_MEDIAMGR_UNABLETOALLOCMEM
2013	MPCLOGMOD_MEDIAMGR_NOISOTRAK
2014	MPCLOGMOD_MEDIAMGR_BADISOBOXSIZE
2016	MPCLOGMOD_MEDIAMGR_BRANDINCOMPT3GPP
2017	MPCLOGMOD_MEDIAMGR_BADMAJ3GPPBRAND

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
2018	MPCLOGMOD_MEDIAMGR_ERORISOBOXVALUE
2019	MPCLOGMOD_MEDIAMGR_FAILTOSTARTRECORD
2020	MPCLOGMOD_MEDIAMGR_NOMEDIAINFOOBJECT
3001	MPCLOGMOD_MEDIAMGR_RECFRAMEDISCARD
3002	MPCLOGMOD_MEDIAMGR_UNEXPECTEDRTSPREPLY
3003	MPCLOGMOD_MEDIAMGR_BADISOBOXVALUE
3004	MPCLOGMOD_MEDIAMGR_BADISOBOXTYPE
3005	MPCLOGMOD_MEDIAMGR_MANDISOBOXMISS
3006	MPCLOGMOD_MEDIAMGR_BUFFTOOSMALLTOPARSEISOHDR
3007	MPCLOGMOD_MEDIAMGR_UNSUPPORTEDAUDRATE
3008	MPCLOGMOD_MEDIAMGR_UNSUPPORTEDVIDRATE
CONTROL	
1001	MPCLOGMOD_CONTROL_INITVGMEDIAINFOFAILED
1002	MPCLOGMOD_CONTROL_INITDSPCAPFAILED
2001	MPCLOGMOD_CONTROL_INVALIDHRTIMERRES
2002	MPCLOGMOD_CONTROL_SDPPARSEFAILED
3001	MPCLOGMOD_CONTROL_INVALIDCFG
3002	MPCLOGMOD_CONTROL_CONNINITFAILED
3003	MPCLOGMOD_CONTROL_CONNMODIFYFAILED
3004	MPCLOGMOD_CONTROL_SENDDTMFNOTALLOWED
3005	MPCLOGMOD_CONTROL_INVALIDCONFIGPARAM
3006	MPCLOGMOD_CONTROL_EVENTPOOLTHRESHOLDREACHED
3007	MPCLOGMOD_CONTROL_EVENTPOOLTHRESHOLDLOWERED
4001	MPCLOGMOD_CONTROL_DIRECTBRIDGE

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
MEDIA	
2001	MPCLOGMOD_MEDIA_RECORDOPENFAILED
2002	MPCLOGMOD_MEDIA_PLAYBACKOPENFAILED
2003	MPCLOGMOD_MEDIA_TRANSCODINGSUPPRESSED
3001	MPCLOGMOD_MEDIA_ACCESSFAILED
3002	MPCLOGMOD_MEDIA_SINKBUFFERFULL
3003	MPCLOGMOD_MEDIA_SOURCEBUFFERFULL
3004	MPCLOGMOD_MEDIA_PACKETBUFFERFULL
3005	MPCLOGMOD_MEDIA_RTPPACKETTOOLARGE
3006	MPCLOGMOD_MEDIA_BUFFERTOOSMALL
3007	MPCLOGMOD_MEDIA_BRIDGEOBJECTNOTFOUND
3008	MPCLOGMOD_MEDIA_H263SORTEROUTOFPACKET
3009	MPCLOGMOD_MEDIA_SILENCEFILLDISABLED
3010	MPCLOGMOD_MEDIA_SENDDTMFDISABLED
3011	MPCLOGMOD_MEDIA_NORTPSTREAMSENDDTMF
3012	MPCLOGMOD_MEDIA_NORTPSTREAMMEDIATRANSMIT
3013	MPCLOGMOD_MEDIA_ERRORDECODINGRFC2833
RTP_INTERFACE	
3001	MPCLOGMOD_RTPIF_INCORRECTTIMEINDEX
3002	MPCLOGMOD_RTPIF_OUTOFSEQUENCEINCOMINGRTP
3003	MPCLOGMOD_RTPIF_INCOMINGRTPDELAY
3004	MPCLOGMOD_RTPIF_ERRORDEFRAMINGPACKET
3005	MPCLOGMOD_RTPIF_UNEXPECTEDPAYLOADTYPE
3006	MPCLOGMOD_RTPIF_ERRORCRYPTOSRTP
3007	MPCLOGMOD_RTPIF_TXRTCPAPPPKTFAIL

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
3008	MPCLOGMOD_RTPIF_TXRTCPAPPPKTDELAY
DSP	
3002	MPCLOGMOD_DSP_NOTRANSCODER
3003	MPCLOGMOD_DSP_CODEC_UNSUPPORTED
4001	MPCLOGMOD_DSP_VIDEOTRANSCODE_START
VGULOGMOD_MAIN	
1002	VGLOG_CANT_OPEN_DLL
2003	VGLOG_SOCKET_SEND_FAILED
3003	MPCLOGMOD_DSP_CODEC_UNSUPPORTED
4001	MPCLOGMOD_DSP_VIDEOTRANSCODE_START
7001	VGLOG_TRACE_GENERIC
MTINTERNAL	
2001	VGLOG_MTINTERNAL_MINORMAXPORT
2002	VGLOG_MTINTERNAL_MINLARGERTHANMAX
3001	VGLOG_MTINTERNAL_OPENFILEERROR
3002	VGLOG_MTINTERNAL_SENDDATAERROR
3003	VGLOG_MTINTERNAL_WRITEFILEERROR
3004	VGLOG_MTINTERNAL_DISCARDRTPPACKET
RTSPSTACK	
2001	VGLOG_RTSP_NEW_FAILED
2002	VGLOG_RTSP_INVALID_CONFIG
2003	VGLOG_RTSP_UNINIT
2004	VGLOG_RTSP_CONSTRUCT_BAD_MSG
2005	VGLOG_RTSP_PARSE_BAD_MSG
2006	VGLOG_RTSP_SOCKET_ERROR

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
3001	VGLOG_RTSP_SOCKET_EVENT
5001	VGLOG_RTSP_SOCKET_CLOSE
MSML	
3001	CMLOGMOD_MSML_CONFIGWARNING
NETANN	
2001	CMLOGMOD_NETANN_ERRORPLAYINGMEDIA
3001	CMLOGMOD_NETANN_INVALIDPARAM
3002	CMLOGMOD_NETANN_JOINCONFERENCEFAILED
LMBASE	
1001	CMLOGMOD_LMBASE_IDGENDIRUNACCBLE
1003	CMLOGMOD_LMBASE_SYSIPNOTRETRVABLE
1004	CMLOGMOD_LMBASE_FAILUPDTEOPENCALLIDFILE
1005	CMLOGMOD_LMBASE_NOTPUTSEQNUMTOCALLIDFILE
2001	CMLOGMOD_LMBASE_RESETCALLIDFILECONTNTINVD
3001	CMLOGMOD_LMBASE_NOMEDIASESSPLAYAUDIO
3002	CMLOGMOD_LMBASE_NOMEDIASESSPLAYDTMF
3004	CMLOGMOD_LMBASE_NOMEDIASESSRECRDAUDIO
3005	CMLOGMOD_LMBASE_NOMEDIASESSSTREAMING
VRMMGR	
NULL	
ADAPTOR	
2001	VGLOG_CONFIGURATION_ERROR
2002	VGLOG_CLOSE_SESSION_FAIL
2003	VGLOG_STOP_FAIL
2004	VGLOG_LOG_FAIL

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
2005	VGLOG_LOAD_GRAMMAR_FAIL
2006	VGLOG_ASR_SETPARAMS_FAIL
2007	VGLOG_ASR_RECOGNIZE_FAIL
2008	VGLOG_PROMPTDONE_FAIL
2009	VGLOG_ASR_INTERPRET_FAIL
2010	VGLOG_TTS_GETPARAMS_FAIL
2011	VGLOG_TTS_SETPARAMS_FAIL
2012	VGLOG_TTS_SPEAK_FAIL
2013	VGLOG_TTS_CONTROL_FAIL
2014	VGLOG_TTS_RESUME_FAIL
2015	VGLOG_TTS_PAUSE_FAIL
2016	VGLOG_TTS_BARGEIN_OCCURRED_FAIL
2017	VGLOG_OPEN_SESSION_FAIL
2018	VGLOG_UNKNOWN_MRCPPROTOCOL
2019	VGLOG_PROVISION_HANDLER_FAIL
2020	VGLOG_REDIRECT_FAIL
2021	VGLOG_DTMFINPUT_FAIL
2022	VGLOG_TRAP_ASR_SENDREQUEST_TIMEOUT
2023	VGLOG_TRAP_ASR_SENDREQUEST_FAILURE
2024	VGLOG_TRAP_ASR_RECEIVE_MRCPRESPONSEERR
2015	VGLOG_TRAP_ASR_RECEIVE_MRCPEVENTERR
2026	VGLOG_TRAP_ASR_RECEIVE_SERVERRESPONSEERR
2027	VGLOG_TRAP_TTS_SENDREQUEST_TIMEOUT
2028	VGLOG_TRAP_TTS_SENDREQUEST_FAILURE
2029	VGLOG_TRAP_TTS_RECEIVE_MRCPRESPONSEERR

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
2030	VGLOG_TRAP_TTS_RECEIVE_MRCPEVENTERR
2031	VGLOG_TRAP_TTS_RECEIVE_SERVERRESPONSEERR
VRMCLIENT	
1001	VGLOG_INVALID_ENG_TYPE
1002	VGLOG_INVALID_ENG_URI
1003	VGLOG_INVALID_ENG_ENTRY
1004	VGLOG_INVALID_ENG_IP_PORT
1005	VGLOG_EMPTY_ENG_LIST
1006	VGLOG_ENG_PARSE_ERROR
1007	VGLOG_MISSING_ENG_TYPE_LIST
1008	VGLOG_INVALID_STACK
1009	VGLOG_ENG_TYPE_INIT_ERROR
1010	VGLOG_STACK_INIT_ERROR
1011	VGLOG_REQ_MGR_INIT_ERROR
1012	VGLOG_CONNECTION_MGR_INIT_ERROR
1013	VGLOG_STACK_HDLR_INIT_ERROR
1014	VGLOG_PROVISION_ERROR
1015	VGLOG_INIT_FAILURE
2001	VGLOG_FILE_STAT_ERROR
2002	VGLOG_GRAM_SIZE_ERROR
2003	VGLOG_GRAM_OPEN_ERROR
2004	VGLOG_GRAM_OFFSET_ERROR
2005	VGLOG_MEM_ALLOC_ERROR
2006	VGLOG_GRAM_READ_ERROR
2007	VGLOG_SERVER_CONNECT_ERROR

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
2008	VGLOG_SERVER_INFO_ERROR
2009	VGLOG_INVALID_PARAM
2010	VGLOG_NO_GRAM_BASE
2011	VGLOG_PING_ERROR
2012	VGLOG_NO_RES_ID
2013	VGLOG_SESSION_STORAGE_ERROR
2014	VGLOG_CHANGE_STATE_ERROR
2015	VGLOG_INVALID_TIMER_EVENT
2016	VGLOG_SESSION_REMOVE_ERROR
2018	VGLOG_INVALID_MSG_ID
2019	VGLOG_UNKNOWN_TIMEOUT
2020	VGLOG_REQUEST_TYPE_FAILURE
2021	VGLOG_TIMER_REMOVE_ERROR
2022	VGLOG_RESPONSE_FAILURE
2023	VGLOG_REQUEST_REMOVE_ERROR
2024	VGLOG_INVALID_REQUEST
2025	VGLOG_SOCKET_DISCONNECT
2026	VGLOG_INVALID_AUDIO_CODEC
2027	VGLOG_SEND_REQUEST_ERROR
2028	VGLOG_STACK_SYSTEM_ERROR
2029	VGLOG_UNIMPLEMENTED_METHOD
2031	VGLOG_LOST_CONNECTION
3001	VGLOG_RECO_ERROR
3002	VGLOG_RECONNECT_SUCCESS
3003	VGLOG_INCORRECT_TTS_MSG_ORDER

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
3004	VGLOG_INCORRECT_NLSML_FORMAT
3005	VGLOG_ERROR_DECODE_FAILURE
3006	VGLOG_GRAMMAR_NOT_EXIST
3007	VGLOG_GRAMMAR_READING_ERROR
3008	VGLOG_HOTKEY_GRAMMAR_ERROR
MRCPV2CLIENT	
1001	VGLOG_FAIL_LOADING_MRCP_MODULE
1002	VGLOG_INVALID_ENG_ENTRY
1003	VGLOG_STACK_INIT_ERROR
1004	VGLOG_REQ_GR_INIT_ERROR
1005	VGLOG_RESOURCE_MGR_INIT_ERROR
1006	VGLOG_FAILED_TO_OPENSESSION
2001	VGLOG_CONFIGURATION_ERROR
2002	VGLOG_CLOSE_SESSION_FAIL
2003	VGLOG_STOP_FAIL
2004	VGLOG_LOG_FAIL
2005	VGLOG_LOAD_GRAMMAR_FAIL
2006	VGLOG_ASR_SETPARAMS_FAIL
2007	VGLOG_ASR_RECOGNIZE_FAIL
2008	VGLOG_PROMPTDONE_FAIL
2009	VGLOG_ASR_INTERPRET_FAIL
2010	VGLOG_TTS_GETPARAMS_FAIL
2011	VGLOG_TTS_SETPARAMS_FAIL
2012	VGLOG_TTS_SPEAK_FAIL
2013	VGLOG_TTS_CONTROL_FAIL

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
2014	VGLOG_TTS_RESUME_FAIL
2015	VGLOG_TTS_PAUSE_FAIL
2016	VGLOG_TTS_BARGEIN_OCCURRED_FAIL
2017	VGLOG_OPEN_SESSION_FAIL
2018	VGLOG_UNKNOWN_MRCPPROTOCOL
2019	VGLOG_PROVISION_HANDLER_FAIL
2020	VGLOG_REMOVE_SESSION_FAIL
2021	VGLOG_GET_SESSION_DATA_FAIL
2022	VGLOG_SOCKET_DISCONNECT
2023	VGLOG_STACK_SYSTEM_ERROR
2024	VGLOG_REQUEST_TYPE_FAILURE
2025	GVPLOG_NO_RES_ID
2026	VGLOG_CHANGE_STATE_ERROR
2027	VGLOG_SESSION_STORAGE_ERROR
2028	VGLOG_REQ_MGR_INIT_ERROR
2029	VGLOG_STACK_HDLR_INIT_ERROR
2030	VGLOG_NO_GRAMMAR_BASE
2031	VGLOG_SEND_REQUEST_ERROR
2032	VGLOG_MRCPV2_PARSE_BAD_MSG
2033	VGLOG_INITIALIZATION_FAIL
2034	VGLOG_SIPSEND_REQUEST_ERROR
2035	VGLOG_RECEIVE_RESPONSE_ERROR
2036	VGLOG_RESPONSE_FAILURE
2037	VGLOG_INVALID_REQUEST
2038	VGLOG_TIMER_REMOVE_ERROR

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
2039	VGLOG_OUTSTANDING_CONN_REMOVE_ERROR
2040	VGLOG_INVALID_MSG_ID
2041	VGLOG_REQUEST_REMOVE_ERROR
2042	VGLOG_SERVER_CONNECTION_ERROR
2043	VGLOG_FAIED_TO_GETRESOURCEINFO
2044	VGLOG_INVALID_AUDIO_CODEC
2045	VGLOG_SESSION_INITIATION_ERROR
2046	VGLOG_SESSION_REMOVE_ERROR
2047	VGLOG_INVALID_TIMER_EVENT
2048	VGLOG_UNKNOWN_TIMEOUT
2050	VGLOG_HOTKEY_GRAMMAR_ERROR
2051	VGLOG_FAIL_CREATE_SIP_USER_AGENT
2052	VGLOG_FAIL_TO_INITIATE_SIP_SESSION
2053	VGLOG_SIP_ERROR
3001	VGLOG_INCORRECT_NLSML_FORMAT
3002	VGLOG_INCORRECT_TTS_MSG_ORDER
3003	VGLOG_FAIL_TO_OPEN_FILE
4001	VGLOG_MRCPCLIENT_DEFAULT_ENGINE
4002	VGLOG_GET_LOCALIP_FAILED
4003	VGLOG_VRMCLIENT_TTSREQ
4004	VGLOG_VRMCLIENT_TTSRESP
4005	VGLOG_VRMLIENT_GET_LOCALIP_FAILE
4006	VGLOG_SETPARAM_ERROR
DTMFRECO	
1000	DTMF_OUT_OF_MEMORY

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
1001	FAILED_TO_INIT_DTMF_RECOGNIZER
2000	GRAM_ERROR
2001	INVALID_SESSION_ID
2002	FAILED_TO_ACCESS_GRAMMAR
2003	INVALID_STATE
2004	FAILED_TO_CREATE_SESSION
2005	GRAM_NUMBER_MISMATCH
2006	GRAMMAR_TYPE_ERROR
2007	GRAMMAR_DEFINE_ERROR
2008	EMPTY_GRAM_ID
2009	FAILED_TO_CREATE_JS
2010	GRAM_ERROR_EXCEEDED_MAX_TABLE_SIZE
2011	FAILED_TO_INIT_XML_CONVERTER
2012	SEMANTIC_INTERPRETATION_ERROR
2013	FAILED_TO_CREATE_DTMF_RECOG_THREAD
2014	FAILED_TO_START_DTMF_RECOG
2015	FAILED_TO_FETCH
2016	FAILED_RECOGNITION
2017	GRAM_SYNTAX_ERROR
2018	FAILED_TO_PARSE_GRAM
2019	FAILED_TO_GENERATE_NLSML
3000	GRAM_WARNING
3001	FAILED_TO_CACHE
3002	FAILED_TO_PROCESS_BUFFERED_INPUT
3003	FAILED_TO_CLEAR_BUFFERED_INPUT

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
3004	FAILED_TO_GENERATE_FLAT_PARSE_LIST
3005	DUPLICATED_RULES
3006	FAILED_TO_ACCESS_RULE_DURING_SI
3007	FAILED_TO_PROCESS_DTMF_INPUT
3008	FAILED_TO_STOP_DTMF_RECOG
3009	FAILED_TO_DELETE_DTMF_SESSION
3010	FAILED_TO_GET_BUFFERED_DTMF
3011	FAILED_TO_DELETE_NOINPUT_TIMER
3012	FAILED_TO_PROCESS_EVENT
3013	FAILED_TO_PROCESS_NOINPUT
3014	FAILED_TO_START_NOINPUT_TIMER
MRCPV1STACK	
1001	VGLOG_SOME_MRCPV1_CRITICAL_ALARM
2001	VGLOG_MRCPV1_NEW_FAILED
2002	VGLOG_MRCPV1_INVALID_CONFIG
2003	VGLOG_MRCPV1_UNINIT
2004	VGLOG_MRCPV1_CONSTRUCT_BAD_MSG
2005	VGLOG_MRCPV1_PARSE_BAD_MSG
2006	VGLOG_MRCPV1_BAD_REQUEST
MRCPV2STACK	
2001	VGLOG_MRCPV2_NEW_FAILED
2002	VGLOG_MRCPV2_INVALID_CONFIG
2003	VGLOG_MRCPV2_UNINIT
2004	VGLOG_MRCPV2_CONSTRUCT_BAD_MSG
2005	VGLOG_MRCPV2_PARSE_BAD_MSG

Table 71: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
2006	VGLOG_MRCPV2_BAD_REQUEST
21906	VGLOG_MRCPV2_SOCKET_ERROR
21908	VGLOG_MRCPV2_SOCKET_CLOSE
21909	VGLOG_MRCPV2_SOCKET_EVENT
21910	VGLOG_MRCPV2_INVALID_SESSIONID
21911	VGLOG_MRCPV2_INVALID_METHOD

Next Generation Interpreter Module ID and Specifiers

The Module ID for the NGI application is 192. Table 72 describes the specifiers for the NGI application module.

Table 72: NGI Specifiers

Specifier ID	Specifier Name	Level
3501	NGI_LOG_JS_WARNING	Warning
3502	NGI_LOG_JS_INFO	Info
3503	NGI_LOG_NET_CONNECT_FAILURE	Warning
3504	NGI_LOG_NET_CONNECT_FAILURE_INFO	Info
3505	NGI_LOG_INITIALIZE_ERROR	Warning
3506	NGI_LOG_CREATE_DIALOG_FAILURE	Info
3507	NGI_LOG_CONFIGURATION	Info
3508	NGI_LOG_INVALID_PROPERTY	Info
3509	NGI_LOG_INVALID_SYNTAX	Warning
3510	NGI_LOG_UNEXPECTED_WARNING	Warning
1000	NGI_LOG_CONVERSION	Info
1001	NGI_LOG_CONVERSION_WARNING	Warning
1002	NGI_LOG_APPLICATION_ERROR	Info (vxml application)
1003	NGI_LOG_APPLICATION_WARNING	Warning (vxml application)

Table 72: NGI Specifiers (Continued)

Specifier ID	Specifier Name	Level
1004	NGI_LOG_SOMETHING_UNEXPECTED	Error
1005	NGI_LOG_UNEXPECTED_WARNING	Warning
1006	NGI_LOG_INCALL_SETUP_FAILURE	Error
1007	NGI_LOG_CREATE_CALL_FAILURE	Error
1008	NGI_LOG_NGI_INITIALIZATION_FAILURE	Error
1009	NGI_LOG_CONFIGURATION_WARNING	Warning
1010	NGI_LOG_RECORDED_FILE_TOO_SMALL	Warning
1011	NGI_LOG_FETCH_FAILURE	Info
1012	NGI_LOG_FETCH_FAILURE_WARNING	Warning
1013	NGI_LOG_GRAMMAR_ERROR	Info
1014	NGI_LOG_PROMPT_FETCH_TIMEOUT	Error
1015	NGI_LOG_PROMPT_FETCH_ERROR	Error
1016	NGI_LOG_FETCH_RESOURCE_TIMEOUT	Error
1017	NGI_LOG_FETCH_RESOURCE_ERROR	Error
1018	NGI_LOG_PARSE_ERROR	Error

Genesys Voice Platform Interpreter Module ID and Specifiers

The Module ID for the GVPi application is 193. Table 73 describes the specifiers for the GVPi application module.

Table 73: GVPi Specifiers

Specifier ID	Specifier Name
103	POP_CTRL_TIMING_INFO
150	CONV_CTRL_GET_PAGE
151	CONV_CTRL_NEW_URL_EXCEPTION
309	XML_CREATEACTION_ENTRYTRACE
310	XML_CREATEACTION_EXITTRACE

Table 73: GVPi Specifiers (Continued)

Specifier ID	Specifier Name
311	XML_PRINTTOKEN_TRACE
312	XML_STACKSIZE_TRACE
320	XML_ACTION_EXEC_TRACE
321	XML_ACTION_EXCEPTION_TRACE
325	XMLPAGE_INDEXFILE
330	XMLPAGE_CREATE_DELETE_TRACE
340	CALL_PERFORMANCE
375	XMLPAGE_CACHE_ACTIONEXEC
380	XML_PAGE_CACHE
381	XML_PAGE_STAT
410	VXML_ACTIONSTATEMAP
440	VXML_CALL_TRACE
441	VXML_GRAMMAR_MATCH
442	VXML_PROMPT_QUEUE
443	VXML_EXEC_CONTEXT
444	VXML_CRDATA_OBJ
445	VXML_TRANS_REC
446	VXML_USER_UTTERANCE
447	VXML_UPDATE_SESS_VARS
448	VXML_GR_REGEXPR
449	VXML_GR_MIN_MAX_TONES
450	VXML_JS_DOM
500	PC_GENERAL
501	PC_DOWNLOAD_TIME
502	PC_UPLOAD_TIME

Table 73: GVPi Specifiers (Continued)

Specifier ID	Specifier Name
503	PC_STREAM_TIME
504	PC_PREFETCH_IGNORE
505	PC_PREFETCH_ABANDON
506	PC_CLEANUP
507	PC_THRDDATA_CLEANUP
508	PC_SESSION_CLEANUP
509	PC_INTRA_HOST
510	PC_NOTIFICATION
511	PC_HTTP_THRD
517	PC_HTTP_GET
518	PC_HTTP_PUT
520	PC_CONN
521	PC_SESSION
522	PC_COOKIE

Call Control Platform

Table 74 lists the Call Control Platform Application Module names and IDs.

Table 74: Call Control Platform Application Module Names and IDs

Module	Module ID
Main Call Control Platform (CCP) application module	151
CCXML Interpreter	152
Media Controller	153

Connection, Dialog, or Conference Events

Table 75 describes the specifiers for the main Call Control Platform module (Module ID = 151). These events are related to a connection, dialog box, or conference.

Table 75: CCP Connection, Dialog, or Conference Events

Module ID	Specifier ID	Description
151	**Critical Events**	
	1	Failed to initialize software.
	Error Events	
	256	Failed to initialize software.
	257	Inbound connection failure.
	258	Media Controller reported error.
	Warning Events	
	514	Inbound connection rejected while in suspended state.
	515	Application did not specify an event name for `<send>`.
	516	History Info header is malformed.
	517	Invalid hints passed.
	Info Events	
	1025	Connection created.
	1026	Connection terminated.
	1027	Sending `180 Ringing` automatically as configured.
	1044	Conference created.
	1045	Conference terminated.
	1046	Dialog created.
	1047	Dialog terminated.
	1048	Dialog transfer request rejected per configured.
	1049	Buffering up join request until ready.
	1050	Issuing buffered join requests.

Media Controller Events

Table 76 describes the Media Controller events.

Table 76: CCP Media Controller Events

Module ID	Specifier ID	Description
153	**Critical Events**	
	1	Failed to initialize software.
	Error Events	
	256	Failed to initialize software.
	257	Device profile entry empty.
	258	Inbound call leg offer was rejected by application.
	259	Bridging server encountered error.
	260	Failure to initialize the Session Factory.
	Warning Events	
	514	Uninitialization encountered problems.
	515	Operation issued on the leg failed.
	516	SDP generation/processing encountered problems.
	517	SIP `491 Glare` occurred.
	518	Maximum number of retries reached.
	519	Maximum number of updates reached.
	520	Operation execution failed.
	521	`NULL` Operation added to Transaction.
	522	CallTerminate received and state is either `DISCONNECTED` or `ERROR`.
	523	The Default SIP Reject Code is `< 300`
	Info Events	
	1024	Connection timeout.
	1025	Dialog Unsupported MIME Type.
	1026	Conference created.

Table 76: CCP Media Controller Events (Continued)

Module ID	Specifier ID	Description
153 (continued)	1027	Conference terminated.
	1028	Standard Conference creation.
	1029	Implicit Conference creation.
	1030	Media established.
	1031	Media modified.
	1032	Media terminated.
	1033	SIP-CallID: `[<SIP Call-ID>]; SendInvite() Failed[<returncode>].`
	1034	SIP-CallID: `[<SIP Call-ID>]; SendResponse(<SIP code>)` for `<SIP method> Failed[<returncode>].`
	1035	SIP-CallID: `[<SIP Call-ID>]; SendCancel() Failed[<returncode>].`
	1036	SIP-CallID: `[<SIP Call-ID>]; SendRequest(<SIP method>) Failed[<returncode>].`
	1037	SIP-CallID: `[<SIP Call-ID>]; SendInfo() Failed[<returncode>].`
	1038	SIP-CallID: `[<SIP Call-ID>]; SendBye() Failed[<returncode>].`
	1039	SIP-CallID: `[<SIP Call-ID>]; SendAck() Failed[<returncode>].`
	1040	Device profile selected.
	1041	List of operations in the transaction: `<OP1, OP2, ...>`.
	1042	SIP `UserCall` not connected.
	1043	SIP `UserCall` error state.
	1044	SIP `UserCall` received a failure response.
	1045	SIP `UserCall` failed sending a message.

Log_4 (INFO) Events

Table 77 describes the CCXML interpreter events at the INFO level.

Table 77: CCXMLI Log_4 INFO Events

Module ID	Specifier ID	Description
152	256	INTR initialization failed.
	352	CCXMLI initialization failed.
	353	Set property value failed.
	1024	INTR initialized.
	1025	INTR uninitialized.
	1026	A new CCXML session created.
	1027	A CCXML session terminated.
	1028	Failed to fetch document.
	1029	Failed to parse document.
	1030	Failed to compile document.
	1031	Document initialization failed.
	1032	Event not caught by application.
	1033	Application log (by <log> tag).
	1034	Error event generated by application.
	1035	Exceed maximum session limit.
	1036	Interpreter already shutting down.
	1037	Session is still alive.
	1120	CCXMLI initialized.
	1122	A new CCXML session created.
	1123	A CCXML session terminated.
	1124	Failed to fetch document.
	1127	Document initialization failed.

Table 77: CCXMLI Log_4 INFO Events (Continued)

Module ID	Specifier ID	Description
	1128	Event not caught by application.
	1129	Application log.
	1130	Error event generated by application.
	1132	Interpreter already shutting down.

Resource Manager

The Module ID for the Resource Manager application is 148.

Table 78 describes the specifiers for the Resource Manager application module.

Table 78: Resource Manager Specifiers

Specifier ID	Specifier Name
257	GVPLOG_RM_UNRECOVERABLEERR
513	GVPLOG_RM_CONFIGERR
514	GVPLOG_RM_CCPSS7ERR
515	GVPLOG_RM_SOCKETERR
516	GVPLOG_RM_RESOURCEALLOCERR
517	GVPLOG_RM_CDRINITERR
518	GVPLOG_RM_CDRUNINITERR
519	GVPLOG_RM_CDRRECORDCREATEERR
520	GVPLOG_RM_CDRRECORDDELETEERR
521	GVPLOG_RM_DIALINGRANGEEXCEED
522	GVPLOG_RM_DIALINGTYPEINVALID
523	GVPLOG_RM_DIALINGEXPRINVALID
524	GVPLOG_RM_DNISNOTEXIST
525	GVPLOG_RM_DEFAULTTENTANTNOTFOUND

Table 78: Resource Manager Specifiers (Continued)

Specifier ID	Specifier Name
526	GVPLOG_RM_REQUESTURITRANSLATIONFAIL
527	GVPLOG_RM_CALLCREATEFAIL
528	GVPLOG_RM_APPPROFILENOTFOUND
529	GVPLOG_RM_TENANTNOTFOUND
530	GVPLOG_RM_DEFAULTIVRPROFILENOTFOUND
531	GVPLOG_RM_DEFAULTSERVICETYPENOTFOUND
532	GVPLOG_RM_MANDATORYURIPARAMNOTFOUND
533	GVPLOG_RM_INVALIDURIPARAM
534	GVPLOG_RM_SERVICEPREREQNOTFOUND
535	GVPLOG_RM_NOMATCHINGSERVICETYPE
536	GVPLOG_RM_NOMATCHINGGWPREFERENCE
537	GVPLOG_RM_CCILIBINVALIDPARAM
538	GVPLOG_RM_CCILIBCONFIGOBJERR
539	GVPLOG_RM_CCILIBRMOBJERR
540	GVPLOG_RM_CCILIBRESOBJNOTFOUND
541	GVPLOG_RM_CCILIBLOGICALRESCREATEFAIL
542	GVPLOG_RM_CCILIBPHYSICALRESCREATEFAIL
543	GVPLOG_RM_CCILIBTENANTNOTFOUND
544	GVPLOG_RM_CCILIBTENANTCREATEFAIL
545	GVPLOG_RM_CCILIBAPPIDNOTFOUND
546	GVPLOG_RM_CCILIBLINKEDRESNOTFOUND
547	GVPLOG_RM_CCILIBPARENTNOTFOUND
548	GVPLOG_RM_CCILIBLOGICALRESGROUPNOTFOUND
549	GVPLOG_RM_CCILIBTENANTCONVERTERROR
550	GVPLOG_RM_CCILIBCAPADDERROR

Table 78: Resource Manager Specifiers (Continued)

Specifier ID	Specifier Name
551	GVPLOG_RM_CCILIBAPPCONVERTERROR
552	GVPLOG_RM_CCILIBINVALIDINPUTARG
553	GVPLOG_RM_RESSESSIONCREATEFAIL
554	GVPLOG_RM_CCILIBAPPCREATEFAIL
555	GVPLOG_RM_CCILIBUPDATEINVALIDCFGOBJ
556	GVPLOG_RM_CCILIBUPDATETENANTNOTFOUND
557	GVPLOG_RM_CCILIBUPDATETENANTPOPULATEFAIL
558	GVPLOG_RM_CCILIBUPDATEAPPNOTFOUND
559	GVPLOG_RM_CCILIBUPDATEAPPPOPULATEFAIL
560	GVPLOG_RM_CCILIBUPDATELOGICALRESNOTFOUND
561	GVPLOG_RM_CCILIBUPDATELOGICALRESADDERR
562	GVPLOG_RM_CCILIBUPDATERESOBJNOTFOUND
563	GVPLOG_RM_CCILIBUPDATEPHYRESCREATEFAIL
564	GVPLOG_RM_CCILIBUPDATEINVALIDOBJ
565	GVPLOG_RM_CCILIBUPDATEINVALIDOBJTYPE
566	GVPLOG_RM_CCILIBUPDATETENANTADDFAIL
567	GVPLOG_RM_CCILIBUPDATEAPPADDFAIL
568	GVPLOG_RM_CCILIBUPDATETENANTREMOVEFAIL
569	GVPLOG_RM_CCILIBUPDATEAPPREMOVEFAIL
570	GVPLOG_RM_CCILIBUPDATETENANTUPDATEFAIL
571	GVPLOG_RM_CCILIBUPDATEAPPLICATIONUPDATEFAIL
572	GVPLOG_RM_REGISTERERROR
574	GVPLOG_RM_POLICYVIOLATIONERROR
575	GVPLOG_RM_GENERIC_ERROR
576	GVPLOG_RM_SUBSCRIPTION_ERROR

Table 78: Resource Manager Specifiers (Continued)

Specifier ID	Specifier Name
577	GVPLOG_RM_POLICYENFORCEMENTVIOLATIONERROR
769	GVPLOG_RM_INVALIDMSG
770	GVPLOG_RM_INVALIDCONFIG
771	GVPLOG_RM_CCPSS7SUBSERFAIL
772	GVPLOG_RM_NETWORKPROBLEM
773	GVPLOG_RM_REQUESTURIPARSEFAIL
774	GVPLOG_RM_OPTIONUSERINFOEXIST
775	GVPLOG_RM_TOHEADERPARSEFAIL
776	GVPLOG_RM_RMSERVICEAGENTBADMSGFORMAT
777	GVPLOG_RM_RMSUSPEND
778	GVPLOG_RM_SIPSERVICESAMEPRECEDENCE
779	GVPLOG_RM_INVALIDCALLTENANTID
780	GVPLOG_RM_FAILEDTOFINDLINKEDTENANT
781	GVPLOG_RM_FAILEDTOFINDLINKEDRESOURCE
782	GVPLOG_RM_LOGICALRESINFONOTFOUND
783	GVPLOG_RM_LOGICALRESPOPULATEFAIL
784	GVPLOG_RM_LOGICALRESSECTIONNOTFOUND
785	GVPLOG_RM_PHYSRESPOPULATEFAIL
786	GVPLOG_RM_TENANTPOPULATEINCOMPLETE
787	GVPLOG_RM_APPINFONOTFOUND
788	GVPLOG_RM_APPPOPULATEINCOMPLETE
789	GVPLOG_RM_DNISEXTRACTFAIL
790	GVPLOG_RM_SETTINGLOGICALRESPROPERTIES
791	GVPLOG_RM_AORNOTFOUND
792	GVPLOG_RM_CAPACITYNOTFOUND

Table 78: Resource Manager Specifiers (Continued)

Specifier ID	Specifier Name
793	GVPLOG_RM_CAPACITYNONUNSIGNED
794	GVPLOG_RM_SETTINGPHYRESPROPERTIES
795	GVPLOG_RM_UPDATELOGICALRESGROUPNOTFOUND
796	GVPLOG_RM_UPDATEPOPULATEPHYRESFAIL
797	GVPLOG_RM_UPDATEFAILGETPHYRES
798	GVPLOG_RM_UPDATEPOPULATELOGICALRESFAIL
799	GVPLOG_RM_UPDATELOGICALRESNOTFOUND
800	GVPLOG_RM_UPDATEPHYRESREMOVED
801	GVPLOG_RM_UPDATETENANTADDED
802	GVPLOG_RM_UPDATEAPPADDED
803	GVPLOG_RM_UPDATELINKEDTENANTREMOVED
804	GVPLOG_RM_UPDATEAPPREMOVED
805	GVPLOG_RM_UDPATELINKEDTENANTUPDATED
806	GVPLOG_RM_UDPATEAPPDATAUPDATED
807	GVPLOG_RM_UPDATEIGNORED
808	GVPLOG_RM_WARNING_BAD_REGEX
809	GVPLOG_RM_SNMP_DISABLED
810	GVPLOG_RM_BURSTAPPBEGIN
811	GVPLOG_RM_BURSTAPPEND
812	GVPLOG_RM_BURSTTENANTBEGIN
813	GVPLOG_RM_BURSTTENANTEND
814	GVPLOG_RM_NETWORKRECOVERY
815	GVPLOG_RM_REDUNDANCYUNDEFINED
816	GVPLOG_RM_UNSUBSCRIBE
817	GVPLOG_RM_SINGLETENANT

Table 78: Resource Manager Specifiers (Continued)

Specifier ID	Specifier Name
818	GVPLOG_RM_HAMODE
1025	GVPLOG_RM_CCPSS7STATE
1026	GVPLOG_RM_COMMNOTICE
1027	GVPLOG_RM_CLUSTERNOTICE
1028	GVPLOG_RM_CCPPROXYSTATE
1029	GVPLOG_RM_STARTUP
1030	GVPLOG_RM_SHUTDOWN
1031	GVPLOG_RM_RMSERVICEAGENTSTATUS
1032	GVPLOG_RM_STATUSLOG
1033	GVPLOG_RM_ACTIVEMODE
1034	GVPLOG_RM_STANDBYMODE
1035	GVPLOG_RM_CONFIGINFO
1281	GVPLOG_RM_PROVCHANGE
1282	GVPLOG_RM_CCPSS7NOTIFY
1283	GVPLOG_RM_MODULECONNECTIVITY
1284	GVPLOG_RM_MODULECONFIGMODIF
1285	GVPLOG_RM_CLUSTERINFO
1286	GVPLOG_RM_NEWCALL
1287	GVPLOG_RM_REGISTERINFO
2305	GVPLOG_RM_GENERIC_TRACE

CTI Connector

Table 79 lists the CTI Connector Application Module names and IDs.

Table 79: CTI Connector Application Module Names and IDs

Module	Module ID
CTI Adaptor	171
CTI Client	172

CTI Adaptor

Table 80 describes the specifiers for the CTI Adaptor module.

Table 80: CTI Adaptor Specifiers

Specifier ID	Specifier Name
1501	CTICA_INVALID_SESSION_ERROR
1502	CTICA_MEMORY_ERROR
1503	CTICA_INTERNAL_ERROR
1504	CTICA_UNSUPPORTED_SIP_EVENT_ERROR
1505	CTICA_UNSUPPORTED_MSG_BODY_ERROR
1506	CTICA_INITIALIZATION_ERROR
1507	CTICA_CCLIB_ERROR
1508	CTICA_MC_ERROR
1509	CTICA_SIPCALLERROR_ERROR
1510	CTICA_ENCODE_DATA_ERROR
1511	CTICA_DECODE_DATA_ERROR
1512	CTICA_SIP_STAT_ERROR
1513	CTICA_CALLOBJECT_STAT_ERROR
1514	CTICA_PARSING_ERROR
1515	CTICA_SNMPLIB_ERROR
1516	CTICA_NOTFOUND_ERROR
1517	CTICA_UNKNOWN_ERROR
1518	CTICA_CTICLIENT_ERROR

Table 80: CTI Adaptor Specifiers (Continued)

Specifier ID	Specifier Name
1519	CTICA_RM_DOWN
1520	CTICA_CONFIG_ERROR
1550	CTICA_CALLFLOW
1551	CTICA_CALLFLOW_QUERYSTRING
1552	CTICA_CALLFLOW_SIPSTAT
1553	CTICA_CALLFLOW_OBJSTAT
1554	CTICA_CALLFLOW_SIP
1555	CTICA_AUTOMATION
1575	CTICA_SNMPDATA

CTI Client

Table 81 describes the specifiers for the CTI Client module.

Table 81: CTI Client Specifiers

Specifier ID	Specifier Name
1401	CTICC_INVALID_SESSION_ERROR
1402	CTICC_INTERNEL_ERROR
1403	CTICC_UNSUPPORTED_URS_TREATMENT_ERROR
1404	CTICC_SEND_XML_MSG_ERROR
1405	CTICC_UNSUPPORTED_IVR_TREATMENT_ERROR
1406	CTICC_INITIALIZATION_ERROR
1407	CTICC_CCIBLIB_ERROR
1408	CTICC_IVR_SERVER_CONNECTION_ERROR
1409	CTICC_IVR_DATAMSG_ERROR
1410	CTICC_IVR_LOGIN_ERROR
1411	CTICC_INVALID_QUERY_STRING_PARAMETER_ERROR
1412	CTICC_RECV_CALLSTATUS_ERROR
1413	CTICC_GETCALLINFOFORRESP_ERROR
1414	CTICC_INVALID_QUERY_STRING_ERROR
1415	CTICC_CALLERROR_ERROR
1416	CTICC_ENCODE_DATA_ERROR
1417	CTICC_DECODE_DATA_ERROR
1418	CTICC_INVALID_CALL_ERROR
1419	CTICC_IVR_SERVER_TIMEOUT_ERROR
1420	CTICC_SNMPLIB_ERROR
1440	CTICC_IVR_SHUTDOWN_ERROR
1450	CTICC_CALLFLOW
1451	CTICC_CALLFLOW_QUERYSTRING

Table 81: CTI Client Specifiers (Continued)

Specifier ID	Specifier Name
1452	CTICC_CALLFLOW_XML
1453	CTICC_CALLFLOW_INFO
1475	CTICC_SNMPDATA
1480	CTICC_CTIC_STARTED
1481	CTICC_CTIC_STOPPED
1482	CTICC_CTIC_STOPPED_GRACEFULLY
1483	CTICC_IVR_SERVER_CONNECTION_UP

Supplementary Services Gateway

The Module ID for the Supplementary Services Gateway is 174.

Table 82 describes the specifiers for the Supplementary Services Gateway application module.

Table 82: Supplementary Services Gateway Specifiers

ID	Specifier Name
20101	SSG_INITIALIZATION_ERROR
20102	SSG_INVALID_PTR_ERROR
20103	SSG_INTERNAL_ERROR
20104	SSG_CCILIB_ERROR
20105	SSG_HTTP_ERROR
20106	SSG_SNMPLIB_ERROR
20107	SSG_CONFIG_OBJECT_ERROR
20111	SSG_DB_CONNECTION_DOWN
20112	SSG_DB_CONNECT_ERROR
20113	SSG_DB_ERROR
20114	SSG_DB_PROCEDURE_FAILED_ERROR

Table 82: Supplementary Services Gateway Specifiers (Continued)

ID	Specifier Name
20121	SSG_HTTP_REQ_STORAGE_ERROR
20122	SSG_REQUEST_PROCESS_ERROR
20123	SSG_SIP_PROCESSING_ERROR
20124	SSG_QUERYSTRING_PARSE_ERROR
20125	SSG_CUSTOM_OBJECT_ERROR
20126	SSG_NOTIFICATION_URL_GET_FAILED
20131	SSG_STARTED
20132	SSG_STOPPED
20133	SSG_SHUTDOWN
20141	SSG_HTTP_QUERYSTRING
20142	SSG_SNMPDATA
20151	SSG_TLIB_INIT_ERROR
20152	SSG_TEVENTS_ERROR
20153	SSG_TLIB_CONN_RECOVERY_ERROR
20154	SSG_TLIB_GENERIC_ERROR
20161	SSG_SIPSERVER_CONTACT_FAILED
20162	SSG_REQUEST_REJECTION_SIPSERVER_NOT_CONNECTED
20163	SSG_SIPSERVER_APPLICATION_NOT_FOUND
20171	SSG_RESOURCE_DN_NOT_REGISTERED
20172	SSG_REQUEST_REJECTION_RESOURCE_DN_NOT_REGISTERED
20173	SSG_TENANT_RESOURCE_DN_NOT_AVAILABLE

PSTN Connector

Table 83 lists the PSTN Connector Application Module names and IDs.

Table 83: PSTN Connector Application Module Names and IDs

Module	Module ID
Dialogic Manager	138
Gateway Manager	139
Media Manager	140
PSTN Connector	141

Dialogic Manager

Table 84 describes the specifiers for the Dialogic Manager module.

Table 84: Dialogic Manager Specifiers

Specifier ID	Specifier Name
1001	DLGC_MGR_INIT_ERROR
2001	DLGC_MGR_D_CHAN_STATUS_DOWN
2002	DLGC_MGR_D_CHAN_STATUS_UP
2003	DLGC_MGR_LINK_ERROR
2004	DLGC_MGR_LINK_OK
3001	DLGC_MGR_B_CHAN_STATUS_DOWN
4001	DLGC_MGR_B_CHAN_STATUS_UP
4002	DLGC_MGR_B_CHAN_STATUS_CHANGED

Gateway Manager

Table 85 describes the specifiers for the Gateway Manager module.

Table 85: Gateway Manager Specifiers

Specifier ID	Specifier Name
1001	GW_MGR_INITIALIZATION_ERROR
2001	GW_MGR_QUERY_PARSE_ERROR
2002	GW_MGR_CALLOBJ_NOT_FOUND
2003	GW_MGR_TDM_HANGUP_ERROR
2004	GW_MGR_UNSUPPORTED_MEDIA
2005	GW_MGR_CODEC_MATCH_ERROR
2006	GW_MGR_INVALID_CALL_STATE
2007	GW_MGR_OTHER_CALLOBJ_NOT_FOUND
2008	GW_MGR_RETRIEVE_MSG_ERROR
2009	GW_MGR_ACTIVATE_MEDIA_ERROR
2010	GW_MGR_DEACTIVATE_MEDIA_ERROR
2011	GW_MGR_ANSWER_CALL_ERROR
2012	GW_MGR_NO_FREE_PORTS_ERROR
2013	GW_MGR_INVALID_DIAL_NUM_ERROR
2014	GW_MGR_DIAL_ERROR
2015	GW_MGR_CREATE_MEDIA_ERROR
2016	GW_MGR_CALLOBJ_CREATE_ERROR
2017	GW_MGR_DNIS_MISSING_ERROR
2018	GW_MGR_ACCEPT_CALL_ERROR
2019	GW_MGR_REFER_NUM_MISSING_ERROR
2020	GW_MGR_UNSUPPORTED_XFER_TYPE_ERROR
2021	GW_MGR_DLGC_BLIND_XFER_UNSUPPORTED_ERROR
2022	GW_MGR_ONE_CHANNEL_XFER_ERROR

Table 85: Gateway Manager Specifiers (Continued)

Specifier ID	Specifier Name
2023	GW_MGR_REPLACES_CALL_ID_MISSING_ERROR
2024	GW_MGR_TWO_CHANNEL_XFER_ERROR
2025	GW_MGR_DESTROY_MEDIA_ERROR
2026	GW_MGR_BRIDGE_ERROR
2027	GW_MGR_SIP_STACK_INIT_ERROR
2027	GW_CREATE_MSG_ERROR
3001	GW_MGR_STOP_RINGBACK_WARN
3002	GW_MGR_UNEXPECTED_BYE_WARN
3003	GW_MGR_UNEXPECTED_CANCEL_WARN
3004	GW_MGR_CALLER_HUNGUP_WARN
3005	GW_MGR_AGENT_HUNGUP_WARN
3006	GW_MGR_START_RINGBACK_WARN
3007	GW_MGR_MAX_GLARE_RETRIES_DONE_WARN
3008	GW_MGR_MEDIA_STOP_TIMEOUT_WARN
3009	GW_MGR_BLIND_XFER_TIMEOUT_WARN
3010	GW_MGR_ATT_XFER_TIMEOUT_WARN
3011	GW_MGR_TWO_CH_XFER_TIMEOUT_WARN
3012	GW_MGR_UNSUPPORTED_MIB_ATTRIB_WARN
4001	GW_MGR_GLARE_OCCURRED_INFO
4002	GW_MGR_CALLOBJ_DELETED_INFO

Media Manager

Table 86 describes the specifiers for the Media Manager module.

Table 86: Media Manager Specifiers

Specifier ID	Specifier Name
1001	MEDIA_MGR_INIT_ERROR
2001	MEDIA_MGR_MEDIA_SESSION_CREATE_ERROR
2002	MEDIA_MGR_MEDIA_SESSION_ACTIVATE_ERROR
2003	MEDIA_MGR_MEDIA_SESSION_DEACTIVATE_ERROR
2004	MEDIA_MGR_MEDIA_SESSION_DESTROY_ERROR
2005	MEDIA_MGR_MEDIA_SESSION_NOT_FOUND
2006	MEDIA_MGR_RTP_SESSION_CREATE_ERROR
2007	MEDIA_MGR_RFC2833_HANDLER_CREATE_ERROR
2008	MEDIA_MGR_BUFFER_QUEUE_CREATE_ERROR
2007	MEDIA_MGR_TDM_INIT_MEDIA_ERROR
2010	MEDIA_MGR_TDM_START_MEDIA_ERROR
2011	MEDIA_MGR_DTMF_ENCODE_ERROR
2012	MEDIA_MGR_DTMF_SEND_ERROR
2013	MEDIA_MGR_DTMF_DECODE_ERROR
2014	MEDIA_MGR_TDM_STOP_MEDIA_ERROR
2015	MEDIA_MGR_DTMF_SEND_TO_TDM_ERROR
2016	MEDIA_MGR_RTP_NETWORK_CREATE_ERROR
2017	MEDIA_MGR_RTP_BRIDGE_CREATE_ERROR
2018	MEDIA_MGR_RTP_NETWORK_DESTROY_ERROR
2019	MEDIA_MGR_RTP_BRIDGE_DESTROY_ERROR
2020	MEDIA_MGR_RTP_MODIFY_NETWORK_ERROR
2021	MEDIA_MGR_RTP_JOIN_ERROR
2022	MEDIA_MGR_RTP_PACKET_SEND_ERROR

PSTN Connector

Table 87 describes the specifiers for the PSTN Connector module.

Table 87: PSTN Connector Specifiers

Specifier ID	Specifier Name
1001	PSTNC_CRITICAL_INITIALIZATION
4001	PSTNC_PROC_STARTED
4002	PSTNC_PROC_STOPPED

Fetching Module

The Module ID for the Fetching Module Application is 80.

Table 88 describes the specifiers for the Fetching Module application module.

Table 88: Fetching Module Specifiers

Specifier ID	Specifier Name	Description
Level: Critical		
40000	FMLOG_MEM_ALLOC_FAIL	Memory allocation failed for %s.
40001	FMLOG_FM_INIT_FAIL	Fetching Module initialization failed.
Level: Error		
20020	FMLOG_SESS_OPEN_FAIL	Open Session to Fetching Server failed.
20021	FMLOG_CONN_FAIL	Connect to Fetching Server failed.
20022	FMLOG_SEND_FAIL	Send to Fetching Server failed.
20023	FMLOG_BAD_SESS_ID	Invalid session ID.
20006	FMLOG_EMSLOG_INIT_FAIL	EMS logging service initialization failed.
20007	FMLOG_BAD_SHMEM_PARAM	Invalid shared memory parameter.
20008	FMLOG_SHMEM_NAME_EMPTY	Empty shared memory name.
20009	FMLOG_SHSEM_NAME_FAIL	Shared semaphore name generation failed.
20010	FMLOG_SHSEM_CREATE_FAIL	Shared semaphore creation failed.
20011	FMLOG_SHSEM_LOCK_FAIL	Shared semaphore lock failed.

Table 88: Fetching Module Specifiers (Continued)

Specifier ID	Specifier Name	Description
20012	FMLOG_SHMEM_MAP_FAIL	Shared memory map failed for file %s.
20013	FMLOG_SHMEM_ATTACH_FAIL	Shared memory attach failed for ID %d.
20014	FMLOG_SHMEM_NAME_FAIL	Shared memory name generation failed.
20015	FMLOG_SHMEM_CREATE_FAIL	Shared memory creation failed for size %d.
20016	FMLOG_SHMEM_READ_FAIL	Unable to read shared-memory.
20017	FMLOG_SHMEM_WRITE_FAIL	Unable to write shared-memory.
20018	FMLOG_GET_PIPE_FAIL	Failed to get pipe name.
20019	FMLOG_OPEN_PIPE_FAIL	Failed to open pipe.
Level: Warning		
30000	FMLOG_SESS_CLS_FAIL	Close Session to Fetching Server failed.
30003	FMLOG_CLS_PIPE_FAIL	Failed to close pipe.

Appendix

B Media Control Platform Reference Information

This appendix provides miscellaneous reference information about the Media Control Platform.

It contains the following sections:

Audio and Video File Formats

This section provides information about the supported file formats for playing and recording audio and video media:

- Audio-Only Formats—Play
- Video-Only Formats—Play (see page 442)
- Combined Audio and Video Formats—Play (see page 442)
- Audio-Only Formats—Record (see page 443)
- Video-Only Formats—Record (see page 445)
- Combined Audio and Video Formats—Record (see page 445)

Audio-Only Formats—Play

Table 89 lists the supported audio-only file formats for playing prompts.

Table 89: Supported Audio File Formats—Play

Expected File Extension	MIME-type	File Format	Sample Size	Encoding
.vox	audio/x-vox audio/vox	Raw audio	8-bit mono	G.711 ulaw, G.711 alaw (depends on platform configuration)
.au	audio/au audio/x-au	Audio with `.au` header	8-bit mono	G.711 ulaw, G.711 alaw, PCM, G.726, G.722 (depends on file header information)
.awb	audio/amr-wb	Raw audio		AMR-WB for encoding
.ulaw	audio/basic audio/PCMU audio/mulaw	Raw audio	8-bit mono	G.711 ulaw
.alaw	audio/x-alaw-basic audio/PCMA audio/alaw	Raw audio	8-bit mono	G.711 alaw
g.722	audio/g722	Raw audio		G.722
.g729	audio/g729	Raw audio		G.729
.adpcm24	audio/x-g726-24 audio/g726-24	Raw audio	24 kb/sec	ADPCM (G.726)
.adpcm	audio/x-g726 audio/x-g726-32 audio/g726 auduio/g726-32 audio/x-adpcm audio/adpcm audio/x-adpcm8	Raw audio	32 kb/sec	ADPCM (G.726)
.adpcm40	audio/x-g726-40 audio/g726-40	Raw audio	40 kb/sec	ADPCM (G.726)

Note: The sample rate is always 8000 Hz except for AMR-WB and G722 which are 16000 Hz. If Media Control Platform detects a non-8000 Hz audio file, it issues a warning message, and plays the prompt as if the sampling rate is 8000 Hz. A configurable Media Control Platform parameter, `mpc.mediamgr.strictsamplingrate`, enables you to prevent the playing of non-8000 Hz audio files (except for AMR-WB).

Table 89: Supported Audio File Formats—Play (Continued)

Expected File Extension	MIME-type	File Format	Sample Size	Encoding
.pcm8	audio/L8	Raw audio	8-bit unsigned mono	Linear PCM
.pcm16	audio/L16	Raw audio	16-bit signed mono	Linear PCM
.wav	audio/wav audio/x-wav	Audio with .wav header		G.711 ulaw, G.711 alaw, G.722, G.726, G.729, PCM, PCM16, GSM 6.10, MS-GSM, AMR, AMR-WB (G.722.2), ADPCM (depends on file header information)
.avi	audio/avi audio/x-avi	Audio stored in AVI container		G.711 ulaw, G.711 alaw, G. 722, G.726, G.729, PCM, PCM16, GSM, AMR, AMR-WB (G.722.2), ADPCM (depends on file header information)
.nist	audio/wav audio/x-wav	Audio with `NIST` header	8-bit mono	G.711 ulaw, G.711 alaw (depends on file header information)
.gsm	audio/x-gsm audio/gsm	Raw audio		GSM 6.10
.msgsm	audio/x-ms-gsm audio/ms-gsm	Raw Audio		MS-GSM
.amr	audio/amr	Raw audio		AMR
.awb	audio/amr-wb	Raw audio		AMR-WB (G.722.2)
.3gp(p)	audio/3gpp	Audio stored in 3GP container		AMR, AMR-WB (G.722.2)

Note: The sample rate is always 8000 Hz except for AMR-WB and G722 which are 16000 Hz. If Media Control Platform detects a non-8000 Hz audio file, it issues a warning message, and plays the prompt as if the sampling rate is 8000 Hz. A configurable Media Control Platform parameter, `mpc.mediamgr.strictsamplingrate`, enables you to prevent the playing of non-8000 Hz audio files (except for AMR-WB).

Note: The Media Control Platform (MCP) supports V7.00 (2006-06) of 3GPP TS 26.244 for 3GP and ISO/IEC 14496-12 ISO; therefore, the MCP may not play latter versions of 3GP files correctly.

Video-Only Formats—Play

Table 90 lists the supported video-only file formats for playing prompts.

Table 90: Supported Video File Formats—Play

Expected File Extension	MIME-type	Sample Rate	File Format	Encoding
.263	video/h263 video/x-h263	30 fps (recommended)	Raw video	h263
.263	video/h263-1998	30 fps (recommended)	Raw video	h263-1998
.264	video/3gp(p) video/x-h264	30 fps (recommended)	Video stored in 3GP container	h264

Note: The .avi recordings produced by GVP are supported with media players that can play avi files only without index tables. A list of known supported media players include Super and ffdshow. A list of known unsupported media players include QuickTime and Nokia Media Converter Pro.

Combined Audio and Video Formats—Play

Table 91 lists the supported audio/video file formats for playing prompts.

Table 91: Supported Audio/Video File Formats—Play

Expected File Extension	MIME-type	Sample Rate	File Format	Encoding
.avi	video/avi video/x-avi	• Audio: 8000 Hz • Video: 30 fps (recommended)	Audio/video stored in AVI container	• Audio: G.711 ulaw, G.711 alaw, PCM, ADPCM (depends on file header information) • Video: h263, h263-1998, h264, vp8 (depends on file header information)
.3gp	video/3gp(p)	• Audio: 8000 Hz • Video: 30 fps (recommended for h263) • Video: 29.97 fps (recommended for h264)	Audio/video stored in 3GP container	• Audio: AMR • Video: h263, h263-1998, h264 (depends on file header information)

Audio-Only Formats—Record

Table 92 lists the supported audio-only file formats for recording.

Table 92: Supported Audio File Formats—Record

MIME-type	Recorded File Format	Sample Size	Encoding	File Extension
audio/amr-wb (g722.2)	Raw audio		AMR-WB	.awb
audio/x-vox audio/vox	Raw audio	8-bit mono	G.711 ulaw, G.711 alaw (depends on platform configuration)	.vox
audio/basic audio/PCMU audio/mulaw	Raw audio	8-bit mono	G.711 ulaw	.ulaw
Notes: • Genesys Voice Platform (GVP) 8.1 supports the 8000 Hz and 16000 Hz audio sampling rates. • Genesys Voice Platform (GVP) 8.1 does not support `.au` and `.nist` file recording.				

Table 92: Supported Audio File Formats—Record (Continued)

MIME-type	Recorded File Format	Sample Size	Encoding	File Extension
audio/x-alaw-basic audio/PMCA audio/alaw	Raw audio	8-bit mono	alaw	.alaw
audio/g722	Raw audio		G.722	.g722
audio/g729	Raw audio		G.729	.g729
audio/x-g726-24 audio/g726-24	Raw audio	24 kb/sec	ADPCM (G.726)	.adpcm24
audio/x-g726 audio/x-adpcm audio/adpcm audio/x-adpcm8 audio/x-g726-32 audio/g726-32 audio/g726	Raw audio	32 kb/sec	ADPCM (G.726)	.adpcm
audio/x-g726-40 audio/g726-40	Raw audio	40 kb/sec	ADPCM (G.726)	.adpcm40
audio/L8	Raw audio	8-bit unsigned mono	Linear PCM	.pcm8
audio/L16	Raw audio	16-bit signed mono	Linear PCM	.pcm16
audio/x-wav;codec=<audio_codec>;rate=<g726_encoding_rate> audio/wav;codec=<audio_codec>;rate=<g726_encoding_rate>	Audio with `.wav` header		audio_codec: ulaw, alaw, pcm, pcm16, g722, g726, gsm. Default: ulaw or alaw (depends on platform configuration). g726_encoding_rate: 16 kb, 24 kb, 32 kb, or 40 kb. **Default:** 32 kb.	.wav

Notes:

- Genesys Voice Platform (GVP) 8.1 supports the 8000 Hz and 16000 Hz audio sampling rates.
- Genesys Voice Platform (GVP) 8.1 does not support `.au` and `.nist` file recording.

Table 92: Supported Audio File Formats—Record (Continued)

MIME-type	Recorded File Format	Sample Size	Encoding	File Extension
audio/x-gsm audio/gsm	Raw audio		gsm 6.10	.gsm
audio/amr	Raw audio		AMR	.amr
audio/3gpp	Audio stored in 3GP(P) container		AMR, AMR-WB(G.722.2)	.3gp(p)
audio/amr-wb	Raw audio	16000 hz	AMR-WB	.awb
Notes: • Genesys Voice Platform (GVP) 8.1 supports the 8000 Hz and 16000 Hz audio sampling rates. • Genesys Voice Platform (GVP) 8.1 does not support `.au` and `.nist` file recording.				

Video-Only Formats—Record

Table 93 lists the supported video-only file formats for recording.

Table 93: Supported Video File Formats—Record

MIME-type	Recorded File Format	Encoding	File Extension
video/h263	Raw video	h263	.263
video/h263-1998	Raw video	h263-1998	.263
video/h264	Raw video	h264	.264

Combined Audio and Video Formats—Record

Table 94 lists the supported audio/video file formats for recording.

Table 94: Supported Audio/Video File Formats—Record

MIME-type	Recorded File Format	Encoding	File Extension
video/avi;codec=<audio_ codec>;rate=<g726_ encoding_rate> ;videocodec=<video_codec> video/x-avi;codec=<audio_ codec>;rate=<g726_ encoding_rate> ;videocodec=<video_codec>	Audio/video stored in AVI container	audio_codec: ulaw, alaw, pcm16, pcm8, adpcm, gsm, g722, g726, g.729, amr, amr-wb (g722.2), none. **Default:** ulaw or alaw (depends on platform configuration). video_codec: h263. **Default:** h263 g726_encoding_rate: 16 kbps, 24 kbps, 32 kbps, or 40 kbps. **Default:** 32 kbps.	.avi
video/3gpp;codec=<audio_ codec> ;videocodec=<video_codec>	Audio/video stored in 3GP(P) container	audio_codec: amr, amr-wb (g.722.2), none. **Default:** amr video_codec: h263, h264. **Default:** h263	.3gp(p)
Note: Genesys Voice Platform (GVP) 8.1 supports the 8000 Hz and 16000 Hz audio sampling rates.			

RTSP Server Support

The Media Control Platform supports the following RTSP servers:

- RTP transport for MP3 audio format using RTSP.
- Darwin Streaming Server v 6.0.3-2
- Helix Server (Columbia) (13.0.0.479)

Note: Because of a streaming hint format compatibility issue, MCP recorded 3GP files cannot be played by the Darwin Streaming Server.

Dynamic Media Control Platform Parameters

This section lists the configuration and service parameters whose values can be set dynamically for a call session.

The dynamic value is obtained from the `gvp.config.<parameter name>=<parameter value>` parameter in the `Request-URI` of the establishing SIP `INVITE`.

The following configuration options can be set dynamically:

asr Section

load_once_per_call[a]

delay_for_dtmf

log_metrics_to_asr

sessmgr Section

maxincalltime

ECS_Fallback

join_fallback

record.start.beep.filename

alert_before_fetch

mediaswitch_on_alert

acceptcalltimeout

sip Section

warningheaders

sendalert

sendrecvevents

logmsg

xfercopy.headers

mpc Section

codec

codecpref

fcr.defaultdtmfhandling

transmitmultiplecodec

appendrejcodec

rtp.dtmf.receive

rtp.dtmf.send

record.defaultdtmfhandling

rtp.tos

rtcp.tos

maxrecordfilesize

disabledcodecs

disabledtranscoders

rtp.request_iframe

a. The value can also be overridden by the value of the gvp.policy.mcp-asr-usage-mode parameter of the IVR Profile, which the Resource Manager passes to the Media Control Platform as a Request-URI parameter.

CPA Configuration Options That Can be Overwritten

Call Progress Analysis (CPA) configuration options can be overwritten by the gvp.service parameters in the IVR Profile. The IVR Profile service parameter must be prefixed by voicexml.gvp.config for VoiceXML services, and by msml.gvp.config for MSML services.

For more information about call progress detection and analysis, see the *Genesys Voice Platform 8.1 Deployment Guide*

Table 95: CPA Options That Can Be Overwritten

msml.cpd.beeptimeout	mpc.cpa.maxsil_before_beep
msml.cpd.postconnectimeout	mpc.cpa.preconn_tones_det_mode
msml.cpd.preconnectimeout	mpc.cpa.ontime_ringback_match_percent
mpc.cpa.enable_log_param	mpc.cpa.ontime_preconn_match_percent
mpc.cpa.enable_log_result	mpc.cpa.silencefilltimeout
mpc.cpa.maxpreconntime	mpc.cpa.ringback
mpc.cpa.maxpostconntime	mpc.cpa.busy
mpc.cpa.maxbeepdettime	mpc.cpa.fastbusy
mpc.cpa.detectstatenolimitbuffdur	mpc.cpa.sit_nocircuit
mpc.cpa.keptdur_before_statechange	mpc.cpa.sit_vacantcircuit
mpc.cpa.priority_normal_machinegreetingdur	mpc.cpa.sit_operatorintercept
mpc.cpa.priority_normal_voicepausedur	mpc.cpa.sit_reorder
mpc.cpa.priority_normal_maxvoicesigdur	mpc.cpa.fax
mpc.cpa.priority_voice_machinegreetingdur	mpc.cpa.custom1
mpc.cpa.priority_voice_voicepausedur	mpc.cpa.custom2
mpc.cpa.priority_voice_maxvoicesigdur	mpc.cpa.custom3
mpc.cpa.priority_machine_machinegreetingdur	mpc.cpa.custom4
mpc.cpa.priority_machine_voicepausedur	mpc.cpa.tone1(..10).segment1(..3).f1(..2).min
mpc.cpa.priority_machine_maxvoicesigdur	mpc.cpa.tone1(..10).segment1(..3).f1(..2).max
mpc.cpa.faxdur	mpc.cpa.tone1(..10).segment1(..3).ontime.min
mpc.cpa.voice_range_db	mpc.cpa.tone1(..10).segment1(..3).ontime.max
mpc.cpa.voice_level_db	mpc.cpa.tone1(..10).segment1(..3).offtime.min
mpc.cpa.maxrings	mpc.cpa.tone1(..10).segment1(..3).offtime.max

SIP Headers

Table 96 lists the SIP headers that the Media Control Platform recognizes and uses. You can use values from many of these headers to send and receive data to and from the VoiceXML or CCXML application in SIP `INFO` messages.

Do not use the header names in Table 96 for any custom headers, or they will be ignored.

Table 96: SIP Headers Known to GVP

SIP Header Name	Standard/Specification (Section)	Description
Accept	RFC 3261 (20.1)	When responding to a SIP `OPTIONS` request, lists all the content types accepted by the component.
Allow	RFC 3261 (20.5)	When responding to a SIP `OPTIONS` request, lists all the methods supported by the component.
Call-ID	RFC 3261 (20.8)	Standard support.
Contact	RFC 3261 (20.10)	Forms the remote request URI in a dialog.
Content-Length	RFC 3261 (20.14)	Standard support.
Content-Type	RFC 3261 (20.15)	Supported content types: • `application/dtmf-relay` • `application/sdp` • `application/text` • `application/www-form-urlencoded` • `message/sipfrag;version=2.0` • `telephone/event`
CSeq	RFC 3261 (20.16)	Standard support.
Diversion	draft-levy-sip-diversion (08)	Exposed to the application as a read-only redirection variable if the `History-Info` header is not available.
Event	RFC 33515	Supported event package: • `refer`
From	RFC 3261 (20.20)	Contains the calling party information (ANI). Maps to the VoiceXML session variable `session.connection.remote.uri`.
History-Info	RFC 4244	The list of header values that are exposed at the application layer as the redirection variable. Maps to the VoiceXML session variable `session.connection.redirect`. • Original Called Number (OCN) is treated as the first entry in the `History-Info` header. • Redirection Reason is treated as a list of all reasons in the `History-Info` header values.

Table 96: SIP Headers Known to GVP (Continued)

SIP Header Name	Standard/Specification (Section)	Description
Min-Expires	RFC 4028 (5)	Minimum session timer.
Max-Forwards	RFC 3261 (20.22)	Standard support.
P-Alcatel-CSBU		Sets the header of the `200OK` response to the initial incoming `INVITE`. This is used for ESS deployments.
P-Asserted-Identity	RFC 3325	Provides the calling party information (ANI) if the `From` header is anonymous. If this header exists, its value overrides the `From` header as the ANI.
P-Called-Party_ID		Provides the DNIS if the `To` header is anonymous. If this header exists, its value overrides the `To` header as the DNIS.
Privacy	RFC 3323	Sets the Presentation Indicator of the VoiceXML session variable `session.connection.redirect`.
Reason	RFC 3326	If the `Reason` header is in the `BYE` message, the reason text will be available as a read-only variable in the application.
Record-Route	RFC 3261 (20.30, 16.12.1)	Specifies the routeset when sending requests within the dialog.
Refer-To	RFC 3515 (2.1)	Sets the destination of the transfer request.
Replaces	RFC 3891	Sets the dialog to replace for whisper transfer.
Require	RFC 3261 (20.32)	Supported option tags: • `100rel` (`PRACK` not supported) • `timer` If the Media Control Platform receives tags that it does not understand, it rejects the request with `420 Bad Extension`.
Route	RFC 3261 (20.34)	Sets the next hop address when sending a request. You can set the value with the application or by configuration. If the `INVITE` request contains `Record-Route` headers, `Record-Route` values override the configured ruttiest for all requests within the dialog.

Table 96: SIP Headers Known to GVP (Continued)

SIP Header Name	Standard/Specification (Section)	Description
RSeq	RFC 3262 (7.1)	Sent by the User Agent Server (UAS) on a reliable response.
Rack	RFC 3262 (7.2)	Sent by the User Agent Client (UAC) to acknowledge (ACK) a reliable response.
PRACK	RFC 3262	Supports reliable provisional responses for a SIP INVITE.
Session-Expires	RFC 4028 (4)	Sets the session expiry time and the refresher role.
Subscription State	RFC 3515	Supported by the REFER method only.
Supported	RFC 3261 (20.37)	Supported option tags: • 100rel (PRACK not supported) • timer
To	RFC 3261 (20.39)	Contains the called party information (DNIS). Maps to the VoiceXML session variable session.connection.local.uri.
Unsupported	RFC 3261 (20.40)	Contains the list of option tags not supported by the User Agent (UA) when rejecting a call.
Via	RFC 3261 (20.42)	Standard support.
Warning	RFC 3261 (20.43)	Returned by a UAS when it failed to negotiate a media session or the request contained a malformed NETANN request. The following warning codes are used in the following situations: • 300 - incompatible network protocol • 301 - incompatible network address • 302 - incompatible transport protocol • 303 - incompatible bandwidth • 304 - unsupported media type • 305 - unsupported media format • 306 - unknown attribute not supported • 307 - unknown parameter was presented • 399 - malformed request URI (malformed NETANN request)

Table 96: SIP Headers Known to GVP (Continued)

SIP Header Name	Standard/Specification (Section)	Description
X-Genesys-CallUUID		Genesys UUID, generated by SIP Server (or T-Server).
X-Genesys-GVP-Session-ID		GVP Session Identifier, generated by the Resource Manager (for new inbound sessions) or the Media Control Platform or the Call Control Platform (for new outbound sessions).
X-Genesys-RM-Application-dbid		The DBID of the IVR Profile (in other words, the VoiceXML or CCXML application).
X-Genesys-GSW-Predictive-Call		The indicator that the SIP Server has received a request to make an outbound call.
X-Genesys-GSW-IVR-Profile-id		The ID of the IVR Profile.
X-Genesys-OutboundData		The indicator that there is user-data attached to the Outbound call.
X-Genesys-GSW-Session-DBID		The user-data parameter for CTI Connector integration with Supplementary Services Gateway.

Handling Error Responses for Outbound Calls

Table 97 summarizes how the Media Control Platform interprets SIP error responses that it receives in response to outgoing `INVITE` requests.

For information about SIP response codes that the Media Control Platform generates, see Appendix D, “SIP Response Codes,” on page 469.

Table 97: Error Response Handling—Outbound Calls

SIP Response		Call End Reason (for Metrics)	Disconnect Reason (to Determine Call/ Transfer Result)	Action in VoiceXML Application
Code	Phrase			
301	Moved Permanently	baddest	CM_DISCREASON_ BADDEST	`error.connection.baddestination` event during `<transfer>`
404	Not Found			
410	Gone			
484	Address Incomplete			
502	Bad Gateway			
604	Does Not Exist Anywhere			
401	Unauthorized	noautho	CM_DISCREASON_ OUT_NOAUTH	`error.connection.noroute` event during `<transfer>` A `<transfer>` form value of `unknown` is assigned.
402	Payment Required			
403	Forbidden			
407	Proxy Authentication Required			
408	Request Timeout	noanswer	CM_DISCREASON_ OUT_NOANSWER	`noanswer` in the `<transfer>` result
480	Temporarily Unavailable	busy	CM_DISCREASON_ OUT_USERBUSY	`busy` in the `<transfer>` result
486	Busy Here			
600	Busy Everywhere			
603	Decline			

Table 97: Error Response Handling—Outbound Calls (Continued)

SIP Response		Call End Reason (for Metrics)	Disconnect Reason (to Determine Call/ Transfer Result)	Action in VoiceXML Application
Code	**Phrase**			
405	Method Not Allowed	unsupported	CM_DISCREASON_ UNSUPPORTED	`error.unsupported.transfer.blind/ consultation/bridge` event during `<transfer>`
488	Not Acceptable Here			
501	Not Implemented			
606	Not Acceptable			
503	Service Unavailable	resourcelimit	CM_DISCREASON_ OUT_NORESRC	`error.connection.noresource` event during `<transfer>`
504	Gateway Timeout	busy	CM_DISCREASON_ OUT_NWBUSY	`network_busy` in the `<transfer>` result
No response				
All other errors		error	CM_DISCREASON_ GENERROR	`error.connection.noroute` event during `<transfer>` A `<transfer>` form value of `unknown` is assigned.

VAR Metrics

Table 98 summarizes the metrics that the Media Control Platform generates when the Next Generation Interpreter (NGI) executes a VAR-specific `<log>` tag. The metrics include the `PCDATA` specified in the `<log>` element.

Table 98 includes information about the valid syntax and values for the VAR-specific `<log>` tag. If the format of the `PCDATA` for the element does not conform to the valid syntax, the VAR metric will not be logged.

For more information about using the VAR `<log>` tag labels (or *extensions*) in VoiceXML applications, see the *Genesys Voice Platform 8.1 Genesys VoiceXML 2.1 Reference Help.*

Formatting Note

Contrary to type conventions in the remainder of this guide, italic text in the <log> tag syntax indicates placeholders for user-specified values. The angle brackets are a required part of the VoiceXML syntax.

Table 98: VAR <log> Tags and Metrics

Metric	<log> Tag Label Syntax and Valid Values
call_result	<log label="com.genesyslab.var.CallResult">*result*[|*reason*]</log> where: • *result* is SUCCESS|FAILED|UNKNOWN. (The default is UNKNOWN.) • *reason* is an optional string of up to 256 characters that provides a textual reason for the call result. **Notes** • The *result* and *reason* values are not case-sensitive. • If the developer specifies a call result other than SUCCESS or FAILED, UNKNOWN is assumed. • Preceding and trailing white space in the *result* is ignored. • The system will truncate *reason* content beyond 256 characters.
call_notes	<log label="label=com.genesyslab.var.CallNotes">*notes*</log> where *notes* are up to 4 KB (4096 bytes) of free-form notes associated with the call. **Notes** • The *notes* collection cannot be empty. • The system will truncate content beyond 4 KB.
ivr_action_start	<log label="com.genesyslab.var.ActionStart">*actionID*[|parentID=*PID*]</log> where: • The *actionID* is the ID of the VoiceXML application action being started. • *PID* is the ID of the parent action, if this action is nested inside some other active action. **Notes** • The *actionID* and *PID* IDs are any valid UTF8 string, to a maximum of 64 characters, that does not contain spaces or pipes. • Action IDs are case-sensitive. • White space is ignored. • An active action is one that has started and not yet ended. If a specified *PID* is not the ID of an active action, the reporting infrastructure will ignore the ivr_action_start metric.

Table 98: VAR <log> Tags and Metrics (Continued)

Metric	<log> Tag Label Syntax and Valid Values
ivr_action_end	`<log label="com.genesyslab.var.ActionEnd">`*actionID*`[|`*result*`[|`*reason*`]]</log>` where: • The *actionID* is the ID of the VoiceXML application action being ended. • The *result* value is one of `SUCCESS|FAILED|UNKNOWN`, indicating the result of the action. The default is `UNKNOWN`. • The *reason* value is an optional string of up to 256 characters that provides a textual reason for the action result. If `ActionEnd` is not explicitly specified, the Reporting Server implicitly ends actions under the following circumstances: • A sibling action (in other words, an action with the same parent) is started. • The call ends. If the Reporting Server implicitly ends an action, the value of *result* is `UNKNOWN`, and the value of *reason* is `NULL`. **Notes** • The *actionID* is any valid UTF8 string, to a maximum of 64 characters, that does not contain spaces or pipes. • The *actionID* is case-sensitive. • The *result* and *reason* values are not case-sensitive. • White space in the metric is ignored. • If the specified *actionID* is not the ID of an active action, the reporting infrastructure will ignore the `ivr_action_end` metric. • If the developer specifies an action result other than `SUCCESS`, `FAILED`, or `UNKNOWN`, the reporting infrastructure will ignore the `ivr_action_end` metric.
ivr_action_notes	`<log label="com.genesyslab.var.ActionNotes">`*actionID*`|`*notes*`</log>` where: • The *actionID* is the ID of the VoiceXML application action. • The *notes* variable is a collection of up to 4 KB (4096 bytes) of free-form notes associated with the action. **Notes** • The a*ctionID* is any valid UTF8 string, to a maximum of 64 characters, that does not contain spaces or pipes. Preceding and trailing white space is ignored. • The *notes* collection cannot be empty. • Any *notes* content beyond 4 KB will be truncated. • VoiceXML action notes (`ivr_action_notes`) may be logged during the specified action or after it has ended.

Table 98: VAR <log> Tags and Metrics (Continued)

Metric	<log> Tag Label Syntax and Valid Values
custom_var	`<log label="com.genesyslab.var.CustomVar">`*name*|*value*`</log>` where: • The *name* is the name of the custom variable. • The *value* is the value of the custom variable. **Notes** • The *name* variable is any valid UTF8 string, to a maximum of 64 characters, that does not contain spaces or pipes. Preceding and trailing white space is ignored. • The *value* variable is any valid UTF8 string, to a maximum of 256 characters. White space is significant. • Custom variables may be specified at any point in a VoiceXML application. • The reporting infrastructure will allow a maximum of eight (8) custom variables to be specified for a given call. Any variables logged beyond the maximum will be ignored.

Appendix

C Tuning Call Progress Detection

This appendix provides information about how to finely tune the behavior of Call Progress Detection (CPD) on the Genesys Voice Platform (GVP) to facilitate diagnostics of unsuccessful calls and better manage contact center campaigns.

It contains the following sections:

Call Progress Detection

Call Progress Detection enables detection and identification of telephone network call progress tones. These tones identify conditions, such as network congestion, busy conditions, and ringback (alerting). Tone detection can provide improved diagnostics of the conditions when a caller cannot be reached successfully. It also provides improved management of call campaigns, in which calls that receive certain SIT results can be removed from the campaign, rather than retried.

CPD matches pre-defined tones against the audio that is received from the network. CPD is typically applied in the pre-connect state, before the call is answered. In addition, in certain cases, it is useful to continue monitoring for CPD tones after the connection is made.

The following sections contain information about the supported CPD tones and how to use GVP configuration options to tune CPD on your platform:

- "Supported North American SIT Tones" on page 460
- "Tone Definition" on page 460
- "Answering Machine Detection" on page 464
- "Beep Detection" on page 466

Supported North American SIT Tones

The Media Control Platform supports North American CPD tones (by default), which include:

- Fax
- Busy
- Fast Busy/Congestion
- Ringback
- Special Information Tones (SIT), which include:
  - No circuit
  - Reorder
  - Operator Intercept
  - Vacant circuit
- Ten additional custom tones

Tone Definition

Tones are either predefined, or custom defined in the platform configuration. Each tone definition includes up to three segments, in which each segment can contain one or two audio frequency bands that must be present for detection to occur. The duration of each segment and the pause between segments can also be configured as part of the tone definition.

For example, the parameters in Table 99 are required to define the SIT `No Circuit` tri-tone:

Table 99: SIT 'No Circuit' Tone Definition

Segment	Frequency 1 Min-Max (Hz)	Frequency 2 Min-Max (Hz)	Tone 'On' time Min-Max (ms)	Tone 'Off' time Min-Max (ms)
1	950-1020	0-0	320-440	0-60
2	1400-1450	0-0	320-440	0-60
3	1740-1850	0-0	320-440	0-100

By assigning these values and ranges to the segments of the tone definition, you are enabling the detection of the SIT `No Circuit` tri-tone during CPD.

The accuracy of frequency detection is +/- 10 Hz for a signal level that is equal to the nominal level for the North American Numbering Plan. (See *Supplement 2* to *ITU.T Recommendation E.180*.)

The detection result is included in the MSML fragment that is executing on the Media Control Platform and is passed to the application in a SIP `INFO` message.

Pattern Types

CPD behavior is controlled by a list of pattern types. Each pattern type (see "Supported North American SIT Tones" on page 460) has an associated configuration option, in which a list of one or more tone definitions (that are mapped to the pattern type) is configured. If the configuration option for a pattern type is not set, detection of that pattern type is disabled.

The list of available tone definition names consists of a set of built-in standard tones and a number of configurable tones. The built-in tones currently include North American definitions for ringback, busy, fast busy, fax, and SIT. These tone definitions are named `na_ringback`, `na_busy`, `na_sit`, and `standard_fax`.

Preconnect, Postconnect and Custom Tones

You can also configure ten additional custom tone definitions, named `tone1` through `tone10`. These custom tone definitions are the vehicle for localization. You can localize the custom tones by configuring them to match the local CPD tone requirements, and then assigning them to tone patterns.

Preconnect Tones

Preconnect tones (busy, fast busy, ringback, SIT, and custom defined tones), can be detected in both preconnect and postconnect states. You can enable the detection mode by configuring the `mpc.cpa.preconn_tones_det_mode` option with a value of `1`. By default, this mode is disabled and preconnect tones are detected in the preconnect state only.

The transition to the connected state (where postconnect detection occurs) can be triggered by using out-of-band signaling, or in the event that ringback is used, when ringback stops (assuming the maximum number of rings is not exceeded).

Fax tone is a postconnect tone and has a treatment different from that of preconnect tones. As a result, assigning a fax tone to a preconnect tone pattern or assigning a preconnect tone to the fax pattern (for example, assigning a fax tone to a busy pattern) is not recommended. If a custom fax tone is required, any one of the ten custom tones can be used for this purpose.

Postconnect Tones

In the postconnect state, the returned result can be `fax`, `human`, or `answering machine`. Optionally, the Media Control Platform can retain a configurable amount of media (received before the connection is available) for postconnect processing. This enables you to control (through the configuration) scenarios in which the audio path to the caller might be open before a `connect` signal is received. Use the `mpc.cpa.keptdur_before_statechange` configuration option, in which a time interval (duration) is defined in milliseconds to control this behavior.

Custom Tones

Custom tones are defined by using a set of configuration options for each tone. The configuration of each of the three segments in a tone includes the frequency range for one of two frequencies—the `on` time range for the segment, and the `off` time range, which marks the end of the segment.

In the following list, <m> is the custom tone identifier (1 through 10), <n> is the segment identifier within the tone (1 through 3), the frequencies are identified as f1 and f2, and the min and max configuration options define a range:

- mpc.cpa.tone<m>.segment<n>.f1min
- mpc.cpa.tone<m>.segment<n>.f1max
- mpc.cpa.tone<m>.segment<n>.f2min
- mpc.cpa.tone<m>.segment<n>.f2max
- mpc.cpa.tone<m>.segment<n>.ontimemin
- mpc.cpa.tone<m>.segment<n>.ontimemax
- mpc.cpa.tone<m>.segment<n>.offtimemin
- mpc.cpa.tone<m>.segment<n>.offtimemax

When f2 values are specified, a second frequency for the segment is implicitly enabled. When segment2 values are specified, a second segment is implicitly enabled, and when segment3 values are specified, a third segment is enabled. These configuration options match the data in the tone definition examples in Table 99 on page 460.

Monitoring Ringbacks

When the Media Control Platform monitors ringbacks to detect the connection, the mpc.cpa.maxrings configuration option is used to define an upper limit for the number of ringbacks. If the number of ringbacks exceeds the value of this option, a timeout result is returned, and it is assumed that the call was not answered. The timeout feature is disabled when this option is configured with a value of 0.

Time Limits

The Media Control Platform supports the configuration of time limits for CPA detection. A timeout result is returned if the timeout interval expires. Three timeout intervals are supported:

1. Timeout interval for the call to advance from the preconnect to the connected state (mpc.cpa.maxpreconntime).
2. Timeout interval for the call fax, human or answering machine detection occurring after the start of the connected state (mpc.cpa.maxpostconntime).
3. Timeout interval for answering machine beep after answering machine result-type detection (mpc.cpa.maxbeepdettime).

If any of these options are configured with a value of 0, the timeout is disabled.

Configuration

You can provision CPD in one of two ways:

- Configuring the Media Control Platform—The values that are set in the [mpc].cpa configuration options enable CPD detection for all calls that land on this particular Media Control Platform.

- Configuring the IVR Profiles—The values that are specified in the IVR Profile service parameters enable IVR application-level CPD tuning and override the Media Control Platform system-wide configuration options. Calls that use these particular IVR Profiles will have customized CPD-related configurations.

For example, an option is configured in Configuration Manager as follows:
`[mpc].cpa.voice_range_db=25`

However, the IVR Profile's `gvp.service-parameter` section has the name/value pair configured as follows:

Name: `voicexml.gvp.config.mpc.cpa.voice_range_db`

Value: `fixed, 20`

When Resource Manager routes a VoiceXML application to Media Control Platform with this particular IVR Profile, it invokes the VoiceXML application, which includes the `mpc.cpa.voice_range_db=20` option as the session parameter setting in the URI. The Media Control Platform honors the session parameter setting first, so the human voice detection is based on a voice range of 20 decibels.

For MSML services, you can set this service parameter in the `gvp.service-parameter` section of the default IVR Profile. For example:

Name:`cpd.gvp.config.mpc.cpa.voice_range_db`

Value: `fixed, 20`

This configuration will then apply to human voice detection in all MSML services that are included in the CPD request.

For a complete list of CPA options that can be overwritten, see Table 95, "CPA Options That Can Be Overwritten," on page 448.

Tuning

This section contains useful information that can be used to tune CPD performance.

CPD performance is controlled by tone patterns and tone definitions. To diagnose CPD issues:

1. Carefully examine the configuration to ensure that the tone patterns include appropriate tone definitions that are accurately configured for your locale and telephony network.
2. If the configuration is correct, collect CPD recordings and results, and analyze them to determine if there are any systemic issues. For example, is the CPD audio in the recorded files valid?

3. Ensure that the audio cut-through is taking place prior to connect. If not, it is likely that the only available CPD results are being delivered by the SIP network itself. For example, if a media gateway is being used to connect to the TDM network, and audio cut-through does not occur, the only available results are from the gateway itself.

Logging

You can enable or disable CPD-related information logging by using the following two configuration options:

- `[mpc].cpa.enable_log_param` (Value: `true` or `false`)
- `[mpc].cpa.enable_log_result` (Value: `true` or `false`)

When the `[mpc].cpa.enable_log_param` is set to `true`, the Media Control Platform logs all of the configuration option values that are used for CPD detection, whenever CPD detection is initiated by an MSML request, or in a VoiceXML application.

Metrics Log Format

The log is in metrics log format, which includes the Global Call ID, timestamp, and configuration information. See the following example of the metrics log format:

```
max_preconnect_time=30000|max_postconnect_time=20000|max_beep_det_time=
30000|no_limit_timeout=30000|chunks_not_flush_on_state_chg=90000|machin
e_greet_dur=1800|voice_pause_dur=1000|max_voice_signal_dur=800|fax_dura
tion=160|voice_range_db=25|voice_level_db=17.5|max_ring_cnt=9|sil_befor
e_beep=4500|preconnect_tone_det_mode=0|notime_ringback_match_percent=50
|ontime_preconnect_match_percent=60
```

Tone Setting Information

In addition, the MCP logs the tone-setting information in the metrics log format. See the following example of the tone setting information:

```
ringbak=tone1|segment=1,f1min=0,f1max=0,f2min=0,f2max=0,ontimemin=20,on
timemax=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,f
1min=0,f1max=0,f2min=0,f2max=0,ontimemin=20,ontimemax=20,offtimemin=0,o
fftimemax=0
```

When the `[mpc].cpa.enable_log_result` option value is set to `true`, the Media Control Platform logs all of the CPA results that are reported by the Media Control Platform. The CPA result log is in the metrics log format, which includes the Global Call ID, timestamp, and CPA result. See the following example of a CPA result:

```
cpa_result Answering machine detected
```

Answering Machine Detection

When the call moves to the postconnect state, Answering Machine Detection (AMD) begins. An answering machine typically delivers a greeting that is different from that of an actual person, or one that is used by a business. An answering machine or service often plays a beep tone to prompt the caller to leave a message, which is then recorded. As a result, a number of parameters and internal heuristics control assessment of the postconnect media, and drive a categorization of the entity that is answering the call.

Configuration

The Media Control Platform enables the configuration of three categories to control AMD behavior:

- Human pause time
- Maximum human voice time
- Answering machine greeting time

Profile Types

In addition, the Media Control Platform supports three AMD profiles, which provide different weightings, based on the expected demographic that is receiving the calls. For example, when detection is ambiguous:

- Normal profile—Does not favor either human or answering machine detection.
- Answering machine profile—Favors answering machine detection.
- Human profile—Favors human detection.

These profiles, each defining separate values, can be selected on a per-call basis by using MSML.

Use the following options to configure the profile types:

- `mpc.cpa.priority_normal_machinegreetingdur`
- `mpc.cpa.priority_normal_voicepausedur`
- `mpc.cpa.priority_normal_maxvoicesigdur`
- `mpc.cpa.priority_voice_machinegreetingdur`
- `mpc.cpa.priority_voice_voicepausedur`
- `mpc.cpa.priority_voice_maxvoicesigdur`
- `mpc.cpa.priority_machine_machinegreetingdur`
- `mpc.cpa.priority_machine_voicepausedur`
- `mpc.cpa.priority_machine_maxvoicesigdur`

Signal-to-Noise Ratio

The Media Control Platform supports two configuration options related to the signal-to-noise ratio, and the level of speech that is detected in the input signal.

- `mpc.cpa.voice_range_db`—Specifies the minimum dynamic range (ratio of maximum-to-minimum energy level) for which each section of the received media is considered to contain an active signal (in decibels).
- `mpc.cpa.voice_level_db`—Specifies the active voice signal level (in decibels) relative to the maximum.

Tuning

Use the following configuration options to tune AMD:

- `voice_range_db`—The signal-to-noise ratio, which indicates voice traffic.
- `machinegreetingdur`—The duration of time after connection, which indicates a machine greeting. If the voice signal is longer than this duration, the input is likely to be considered an answering machine. If the voice signal is shorter than this, and longer than `maxvoicesignal` duration, the result is weighted, based on the profile that is in use.
- `maxvoicesigdur`—The duration of time after connection, which indicates a voice signal. If the voice signal is shorter than this duration, the input is likely to be considered human. If the voice signal is longer than this, and shorter than `machinegreetingdur` duration, the result is weighted, based on the profile that is in use.
- `voicepausedur`—The amount of silence that indicates the end of AMD.

If you observe:	Make the following adjustments:
• Longer human greetings that are being identified as a machine.	Increase the value of the `maxvoicesigdur`, and `machinegreetingdur` configuration options.
• Noisy greetings not identified as human.	Reduce the value of the `voice_level_db` configuration option.
• Takes too long to connect after human greeting.	Reduce the value of the `voicepausedur` configuration option.

Beep Detection

The Media Control Platform supports optional answering machine beep detection. The request for beep detection is passed as part of the MSML fragment within the request for CPD/AMD. Beep detection takes place as part of AMD and enables identification of the beep tone that usually follows an answering machine greeting. This phase of detection, which is indicated by a period of silence, then a beep, followed by another period of silence, has the following characteristics:

- A transition from low energy to a period of strong energy in the signal.
- Detection of a beep that has one or two strong frequencies present in the signal.
- The presence of this signal for a minimum amount of time followed by a minimum period of silence.

Note: This feature detects any single or dual frequency response in the signal.

Configuration

The `mpc.cpa.maxbeepdettime` configuration option controls how long the Media Control Platform waits for a beep detection result after an answering machine has been identified. When this option value is set to `0,` the timeout feature is disabled.

The `mpc.cpa.maxsil_before_beep` configuration option controls the maximum amount of silence allowable before a beep. If this value is exceeded, beep detection is abandoned and a silence timeout result is returned.

The minimum `on` time, and the minimum `off` time are currently not configurable.

Tuning

The following configuration option is the only one that impacts beep detection:

`maxsil_before_beep`—The maximum amount of silence allowable prior to the detection of a beep.

If you observe:	Then adjust:
Delays before connection to the application, or beeps not being detected because the application has already started.	Increase or decrease the beep detection timeout.

Continuous Tone Detection

Beginning with release 8.1.6, you can configure CPA to detect a continuous tone, to satisfy specific—but not universal—conditions. This configuration provides a mechanism in the Media Server CPD to handle continuous tone detection, by making the definition of *continuous* configurable.

The UK, Ireland, UAE and at least 7 other countries use a continuous tone for the states `confirmation`, `howler`, `paytone`, or `number unattainable`.

In these environments, where outbound calling applications need to handle the state `number unattainable` (which is a continuous tone), set the value of `ontimemax` to `0` (zero), to disable checking the ontime against a maximum value. As a result, any tone lasting longer than the ontimemin value is considered to be “continuous.”

Genesys

Appendix

D SIP Response Codes

This appendix lists the Session Initiation Protocol (SIP) responses that Genesys Voice Platform (GVP) components send or receive in response to error conditions and other events.

It contains the following section:

For information about how the Media Control Platform handles error responses that it receives for outbound call requests, see Table 97 on page 453.

SIP Responses to Inbound Calls

Table 100 summarizes the SIP response codes, other than the normal `200 OK` responses, that the Resource Manager (RM), Media Control Platform (MCP), and Call Control Platform (CCP) signal in response to error conditions and other events during incoming and outbound call setup and processing. Table 100 also lists the configuration options to customize those responses, where applicable.

Resource Manager handles SIP responses from other components in accordance with rules described in the *Genesys Voice Platform 8.1 Deployment Guide*.

Table 100: SIP Response Codes

SIP Response		Sent By	Situations	Configurable Options and Notes
Code	Phrase			
100	Trying	MCP CCP RM PSTN Connector	The immediate response to a valid `INVITE` request.	
180 Ringing		MCP	The default intermediate response to an `INVITE` request.	`sip.sendalert` (see page 185)
		CCP	Intermediate response is sent for all incoming calls, on `<accept>` (no media bridge configured). Depending on configuration: • Response is sent when `<send>` is called. • Response is sent immediately after sending `100 Trying`.	`ccpccxml.sip.send_progressing`
		PSTN Connector	The default intermediate response to an `INVITE` request.	
183	Session Progress	MCP	The non-default intermediate response, which includes SDP information.	`sip.sendalert` (see page 185)
		MCP	An incoming call is being offered to the Next Generation Interpreter (NGI) for debugging. The NGI passes the debugger IP address and port information to the calling party in the following SIP headers: • `X-GVP-NGI-DEBUG-IP` • `X-GVP-NGI-DEBUG-PORT` **Note:** The information can also be sent in `INVITE` messages.	Sent if debugging is enabled on the MCP: `vxmli.debug.enabled`

Table 100: SIP Response Codes (Continued)

SIP Response		Sent By	Situations	Configurable Options and Notes
Code	**Phrase**			
183	Session Progress (continued)	CCP	Intermediate response sent for incoming calls when a media bridge has been configured between this bridge and any other endpoint. The response includes the appropriate SDP content.	If the media bridge requires changes, they are implemented through subsequent SDP updates in re-`INVITE, 200 OK,` and `183` messages. **Note:** When a `BYE` is received on a SIP dialog that is associated with an endpoint while a transition involving that endpoint is being executed, any new bridge involving the endpoint will fail. The `error.connection.join` event is thrown, with an empty `Reason` property.
202 Accepted		MCP	A `REFER` request to initiate an outbound call outside of a SIP dialog is accepted by the VoiceXML application.	
		PSTN Connector	A REFER request to initiate outbound call for Dialogic Blind Transfer or AT&T Out-of-band Transfers.	
3xx	[Various]	MCP	The MCP, acting as a User Agent Server (UAS), failed to negotiate a media session or the NETANN request was malformed.	See `Warning` header information in Table 96 on page 449.
302	Moved Temporarily	CCP	The platform is redirecting a call in the `ALERTING` state (`<redirect>` tag). If the CCXML application specifies a `<reason>` attribute, the reason is included in the text portion of the `Reason` header.	To customize the SIP response code for specific situations, use the `<hints>` attribute of the `<redirect>` tag—the `responseCode` property of the `hints` object specifies the response code that is to be used.

Table 100: SIP Response Codes (Continued)

SIP Response		Sent By	Situations	Configurable Options and Notes
Code	Phrase			
400 Bad Request		RM	Malformed `Request-URI` in a `REGISTER` message.	
		MCP	Malformed `Request-URI` in an `INVITE` message.	
		CCP	The initial CCXML page URI is malformed.	
		PSTN Connector	Malformed protocol. PSTN Connector failed to extract the SDP message from the INVITE., or the custom header for the CPA is incorrect.	
403	Forbidden	RM	The domain name in a `REGISTER` request does not match the configured domain name.	
		RM	Call is rejected because of an IVR Profile policy (dialing rule).	IVR Profile: `gvp.policy.dialing-rule-forbidden-respcode` **Note:** An equivalent configuration option also enables an alarm to be set.

		RM	Service request is rejected because the IVR Profile policy does not allow the service in the session.	IVR Profile:
				`gvp.policy.conference-forbidden-respcode`
				`gvp.policy.external-sip-forbidden-respcode`
				`gvp.policy.outbound-call-forbidden-respcode`
				`gvp.policy.transfer-forbidden-respcode`
				`gvp.policy.voicexml-dialog-forbidden-respcode`
				Note: Equivalent configuration options also enable an alarm to be set.

Table 100: SIP Response Codes (Continued)

SIP Response		Sent By	Situations	Configurable Options and Notes
Code	Phrase			
404	Not Found	RM	The Resource Manager could not match the incoming request to an IVR Profile.	
		RM	The Resource Manager could not match the incoming request to a service.	
		MCP	The Request-URI has `conf` as the user part, but does not have a `conf-id` parameter.	
		PSTN Connector	The PSTN Connector received an unauthorized number or invalid digits from the TDM network.	
405	Method Not Allowed	RM	`REFER`, `OPTIONS`, `SUBSCRIBE`, or `INFO` message was sent outside of an existing SIP dialog.	
408	Request Timeout	RM	The Resource Manager does not receive a response from the resource.	
		RM	The Resource Manager has not received a response from any resource.	
		RM	An `INVITE` request specifies a host for which the Resource Manager is not responsible. (By default, no responsible domains are specified, so that all requests are accepted.)	
		PSTN Connector	The PSTN Connector did not receive a response from the TDM network for an outbound call.	
415	Unsupported Media Type	PSTN Connector	The PSTN Connector received an unsupported media type.	
420	Bad Extension	MCP	The MCP rejects an incoming call that requires `100rel` (SIP Provisional Message Reliability).	

Table 100: SIP Response Codes (Continued)

SIP Response		Sent By	Situations	Configurable Options and Notes
Code	Phrase			
423	Interval Too Brief	RM	The Resource Manager received a `REGISTER` request for a registration period that fell below the configured minimum expiry time.	
480	Temporarily Unavailable	RM	A conference call fails because of insufficient resource port capacity or because the conference has reached maximum size.	`rm.conference-sip-error-respcode`
		RM	The Resource Manager is not able to select a resource to forward the request to, because a suitable resource is not available.	`rm.resource-unavailable-respcode`
		RM	The Resource Manager is not able to select a resource to which to forward the request, because the deployment does not include the required resource.	`rm.resource-no-match-respcode`
		RM	Call is rejected because usage limits, as specified in the IVR Profile policy, have been exceeded.	IVR Profile: `gvp.policy.usage-limit-exceeded-respcode` **Note:** An equivalent configuration option also enables an alarm to be set.
		CCP	The platform is rejecting an incoming connection in the `ALERTING` state (`<reject>` tag). If the CCXML application specifies a `<reason>` attribute, the reason is included in the text portion of the `Reason` header.	`ccpccxml.defaultrejectcode` To further customize the SIP response code for specific situations, use the `<hints>` attribute of the `<reject>` tag—the `responseCode` property of the `hints` object specifies the response code that is to be used.
		CCP	The platform is not in `READY` state, and is therefore rejecting all `INVITE` and `OPTIONS` requests.	`ccpccxml.defaultrejectcode`

Table 100: SIP Response Codes (Continued)

SIP Response		Sent By	Situations	Configurable Options and Notes
Code	Phrase			
		PSTN Connector	There are no ports available for an outbound call, or there is no answer from the TDM network.	
481	Call Does Not Exist	MCP PSTN Connector	The platform received a request that does not match with any existing dialog.	
484	Address Incomplete	PSTN Connector	The DNIS is not present in the TO header of the INVITE message.	
487	Request Terminated	MCP	A `CANCEL` or `BYE` is received before the final response to the `INVITE` is sent.	
488	Not Acceptable Here	CCP	An endpoint's SDP capabilities cannot be obtained.	The response includes a `Warning` header with warning code `399,` and one of the following warning text messages: • `Unable to generate an offer`—The `INVITE` contained no SDP. • `Unable to generate an answer`—All other situations.

Table 100: SIP Response Codes (Continued)

SIP Response		Sent By	Situations	Configurable Options and Notes
Code	Phrase			
500	Server Internal Error	RM	The Resource Manager received a `REGISTER` request from a resource about which it had no information from Management Framework.	
		RM	The Resource Manager does not receive a 2xx response from any resource (see the *Genesys Voice Platform 8.1 Deployment Guide*).	
		MCP	Unable to create a media session to handle the call.	The response includes an explanatory `Warning` header (see `Warning` header information in Table 96 on page 449).
		MCP	The MCP is unable to fetch or parse the VoiceXML document.	
		PSTN Connector	The PSTN Connector failed to make an outbound call.	
503	Service Unavailable	RM	The Resource Manager is in suspend mode when a new session request arrives.	`rm.suspend-mode-respcode`
		CCP	The CCP is unable to create a CCXML session to handle the call.	

Table 100: SIP Response Codes (Continued)

SIP Response		Sent By	Situations	Configurable Options and Notes
Code	Phrase			
503	Service Unavailable (continued)	RM	Call is rejected because usage limits for the service, as specified in the IVR Profile policy, have been exceeded.	IVR Profile: `gvp.policy.ccxml-usage-limit-exceeded-respcode` `gvp.policy.conference-usage-limit-exceeded-respcode` `gvp.policy.voicexml-usage-limit-exceeded-respcode` **Note:** An equivalent configuration option also enables an alarm to be set.
		MCP	A conference cannot be created because of a resource problem (for example, failed to join to conference because of the conference limit).	
		MCP	The Media Server is not accepting new calls for a reason other than those that are covered by the `500` response.	All error responses to an `INVITE` request, if they do not involve SDP negotiation, will contain a `Warning` header with a value of `399` and a human-readable description of the error.
		MCP	The VoiceXML application could not be fetched or parsed.	
		CCP	The platform is not in `READY` state, and is therefore rejecting all HTTP requests to start a new CCXML session.	`ccpccxml.defaultrejectcode`
		PSTN Connector	The PSTN Connected received a “Service Not Available: from the TDM network.	
603	Declined	PSTN Connector	The PSTN Connector failed to send a Redirection FACILITY message in an AT&T Conference Transfer.	

Table 100: SIP Response Codes (Continued)

SIP Response		Sent By	Situations	Configurable Options and Notes
Code	Phrase			
BYE message		CCP	Bridge failure resulting from the failure of endpoints to negotiate SDP might cause the CCP to send SIP BYE messages to the components involved.	The Reason header value is set to Application Disconnect.
[Configurable]		RM	The Resource Manager returns a response to a SIP OPTIONS message.	rm.options_response_contenttype rm.options_response_msg_body
[Configurable]		SIP Server	When the Resource Manager receives any of the configured SIP responses from SIP Server for a request to a gateway resource, it will retry the request on other gateway resources in the logical resource group.	Resource Manager: <gateway resource group>.noresource-response-code

Appendix

E Device Profiles

This appendix provides details for configuring device profile and summarizes the settings for the default device profiles that are provisioned for the Call Control Platform and CTI Connector. It contains the following sections:

Device Profile Usage

The Call Control Platform and the CTI Connector use device profiles to determine the behavior of the devices with which it interacts, in order to produce the most appropriate SIP messages. Each call handled by Call Control Platform or CTI Connector is associated with a device profile. The profile is used to customize Session Description Protocol (SDP) information sent to the device.

This section describes the device profile parameters and how each parameter affects the SDP formation.

Sending SDP

This section describes the Call Control Platform and the CTI Connector behavior based on device profile configuration when constructing the SDP.

Inbound Usage Examples

This section describes some usage examples for inbound calls.

Offer-less Initial INVITE Without Bridging

Example 1 Figure 40 shows that an SDP offer is generated in a183 response or a 200OK response with the following settings:

- unjoined-initial-answer-pref equals none or connectionless-sdp
- connectionless-sdp-type equals hold

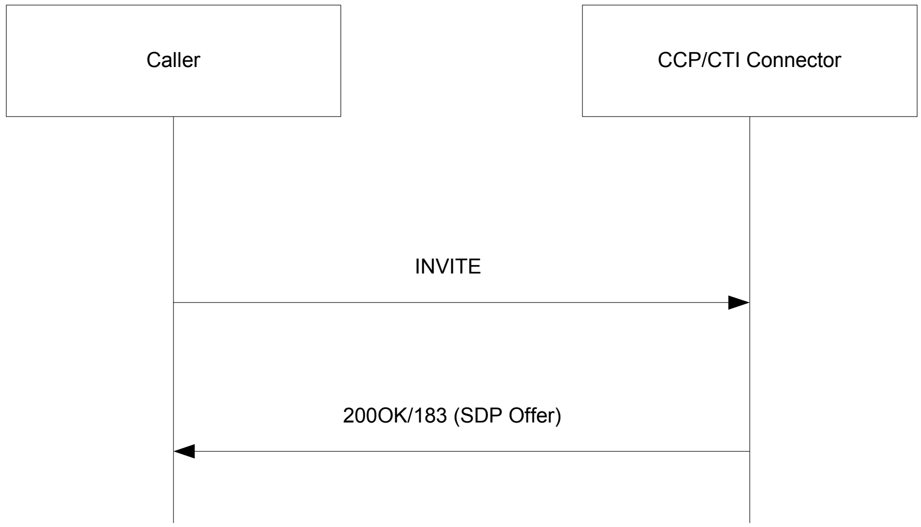


Figure 40: Offer-less INVITE for Inbound Call Example 1

This SDP offer contains a connection line with 0.0.0.0 as the IP address.

Example 2 Figure 41 shows an SDP offer that is generated in a 183 response or a 200OK response with the following settings:

- unjoined-initial-answer-pref equals none or connectionless-sdp
- connectionless-sdp-type set to non-routable

Figure 41: Offer-less INVITE for Inbound Call Example 2

This SDP offer contains a connection line with 1.1.1.1 as the IP address.

Example 3 Figure 42 shows an SDP offer that is generated in a `183` response or a `200OK` response with the following settings:

- `offer-answer-support` equals `true`
- `connectionless-sdp-type` equals `none`
- `nomedia-sdp-support` equals `true`

or

- `unjoined-initial-answer-pref` equals `nomedia-sdp`
- `offer-answer-support` equals `true`
- `nomedia-sdp-support` equals `true`

Figure 42: Offer-less INVITE for Inbound Call Example 3

This SDP offer does not contain any media lines.

Example 4 Figure 43 on page 482 shows that CCP/CTI Connector is unable to decide which method to use for generating an SDP offer with the following settings configured:

- `offer-answer-support` equals `true`
- `connectionless-sdp-type` equals `none`
- `nomedia-sdp-support` equals `false`

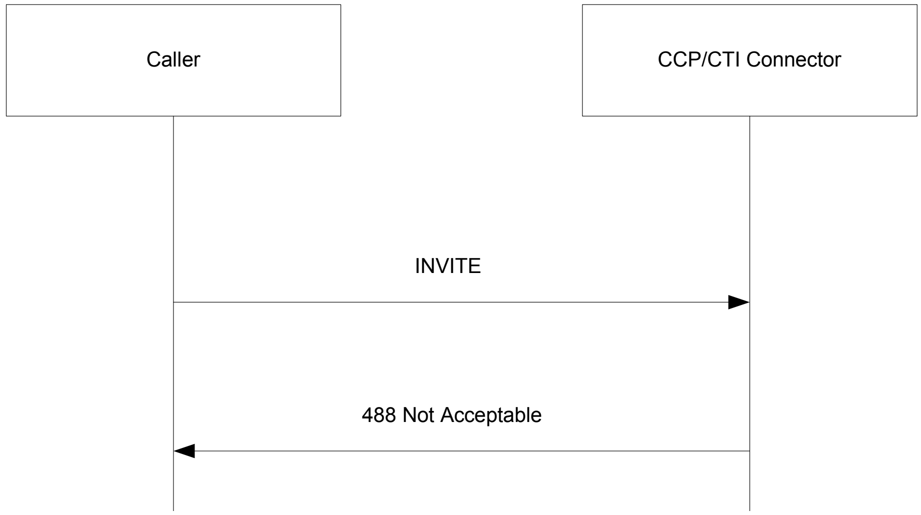


Figure 43: Offer-less INVITE for Inbound Call Example 4

A SIP 488 response is generated to terminate the call.

Initial Inbound Offer Without Bridging

Example 1 Figure 44 shows that an SDP answer is generated in a `183` response or a `200OK` response with the following settings:

- `unjoined-initial-answer-pref` equals `none` or `connectionless-sdp`
- `connectionless-sdp-type` equals `hold`

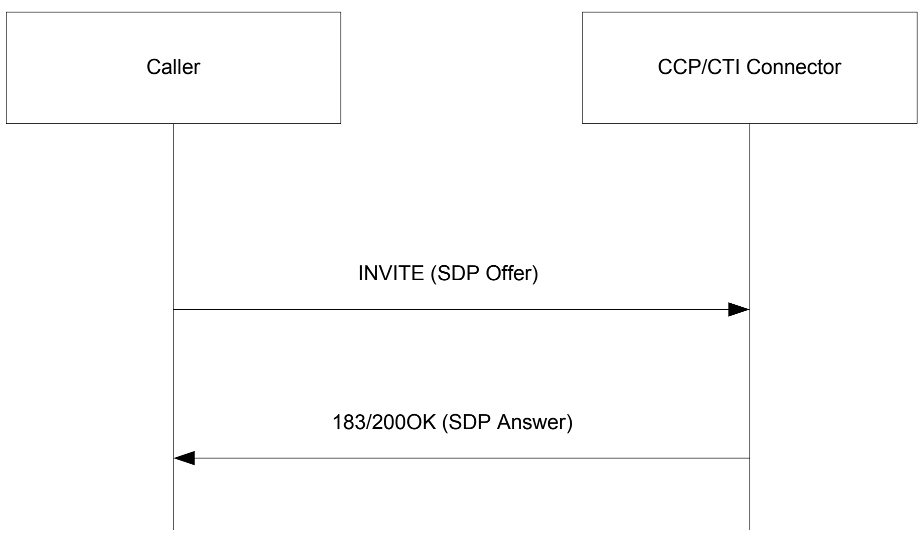


Figure 44: Initial Inbound Offer Without Bridging Example 1

This SDP contains a connection line with 0.0.0.0 as the IP address.

Example 2 Figure 45 shows that an SDP answer is generated in a `183` response or a `200OK` response with the following settings:

- `offer-answer-support` equals `true`
- `unjoined-initial-answer-pref` equals `reject-media`

or

- `unjoined-initial-answer-pref` equals `none`
- `offer-answer-support` equals `true`
- `connectionless-sdp-type` equals `none`

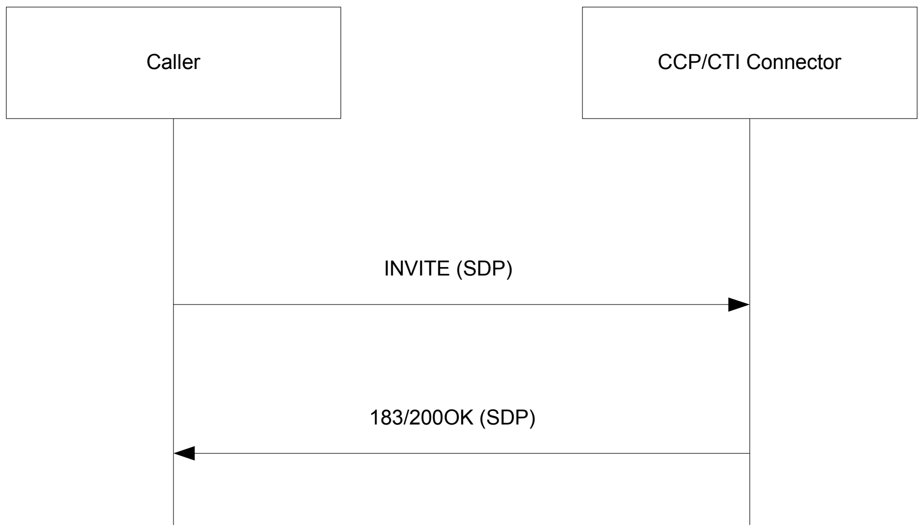


Figure 45: Initial Inbound Offer Without Bridging Example 2

This SDP response has all media lines in the offer with ports set to 0.

Example 3 Figure 46 shows that an SDP answer is generated in a `183` response or a `200OK` response with the following settings.

- `offer-answer-support` equals `false`
- `unjoined-initial-answer-pref` equals `reject-media`
- `nomedia-SDP-support` equals `true`

or

- `offer-answer-support` equals `false`
- `unjoined-initial-answer-pref` equals `nomedia-SDP`
- `nomedia-SDP-support` equals `true`

or

- `offer-answer-support` equals `false`
- `unjoined-initial-answer-pref` equals `none`
- `connectionless-sdp-type` equals `none`

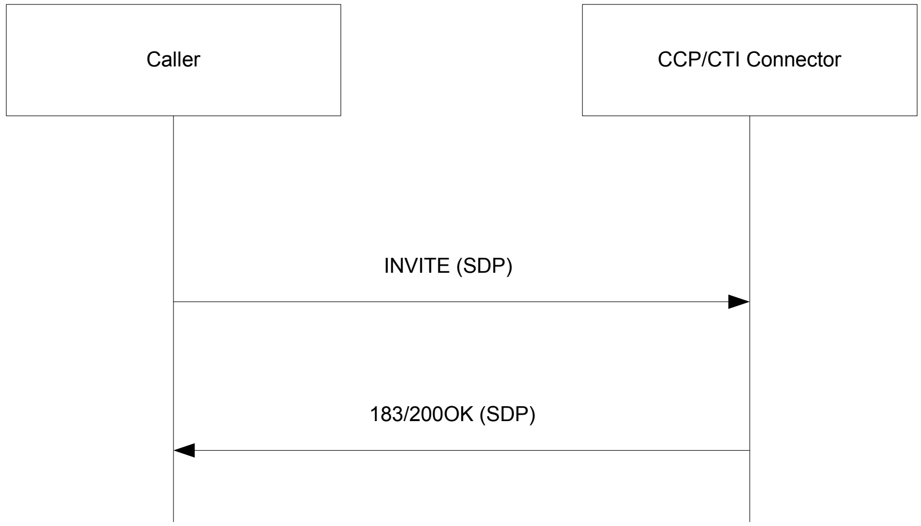


Figure 46: Initial Inbound Offer Without Bridging Example 3

This SDP response does not contain any media lines.

Example 4 Figure 47 shows that the platform could not determine a valid way to reject the media based on the device profile information with the following settings:

- `offer-answer-support` equals `false`
- `unjoined-initial-answer-pref` equals `reject-media`
- `nomedia-SDP-support` equals `false`

or

- `unjoined-initial-answer-pref` equals `none`
- `connectionless-sdp-type` equals `none`
- `nomedia-SDP-support` equals `false`
- `offer-answer-support` equals `false`

Figure 47: Initial Inbound Offer Without Bridging Example 4

The incoming call is rejected.

Outbound Usage Examples

This section describes some usage examples for outbound calls.

Offer-less Outbound INVITE

Example 1 Figure 48 shows an SDP answer in an ACK response that contains a connection line with 0.0.0.0 IP address. The configuration settings are as follows:

- `offer-less-invite-support` equals `true`
- `unjoined-initial-offer-pref` equals `offer-less`
- `unjoined-initial-answer-pref` equals `connectionless-SDP`
- `connectionless-sdp-type` equals `hold`

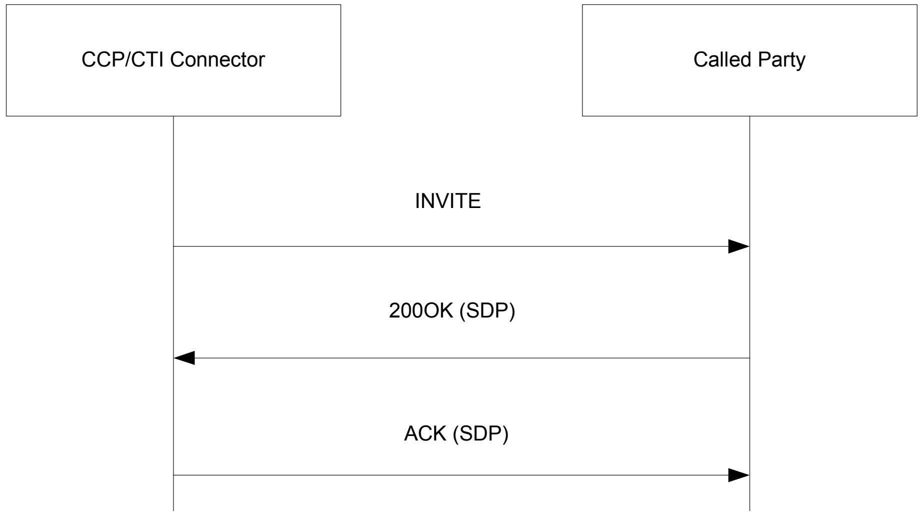


Figure 48: Offer-less Outbound INVITE Example1

Example 2 Figure 49 shows an SDP offer that is generated in INVITE with the following settings:

- `connectionless-sdp-type` equals `non-routable`
- `offer-less-invite-support` equals `false`
- `unjoined-initial-offer-pref` equals `none`

or

- `connectionless-sdp-type` equals `non-routable`
- `unjoined-initial-offer-pref` equals `connectionless-SDP`

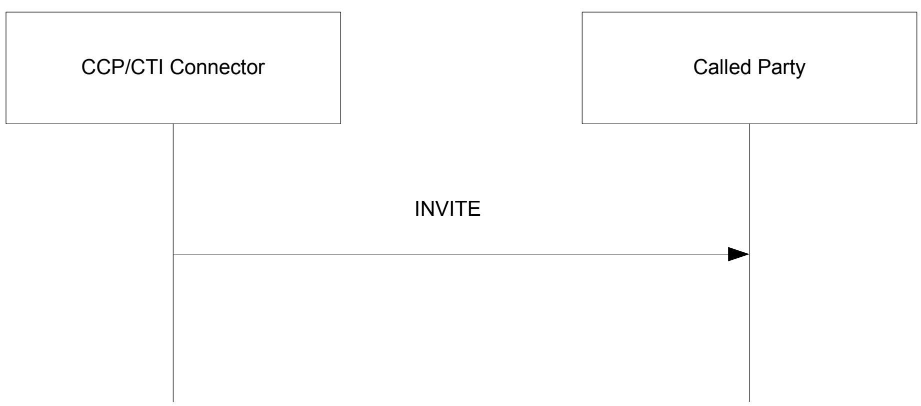


Figure 49: Offer-less Outbound INVITE Example 2

This SDP offer contains a connection line with 1.1.1.1 as the IP address.

Example 3 The platform cannot generate an outbound INVITE based on the device profile with the following settings:

- `offer-less-invite-support` equals `false`
- `connectionless-sdp-type` equals `none`

Example 4 Creating a dialog to Media Server is an exceptional case. The platform will always set the ports in each media line to 0. This puts the call on hold if the device supports offer and answer.

Offer-less Early Join

Figure 50 shows an early join of an inbound call to an outbound call. The outbound initial INVITE is always offer-less regardless of its device profile setting.

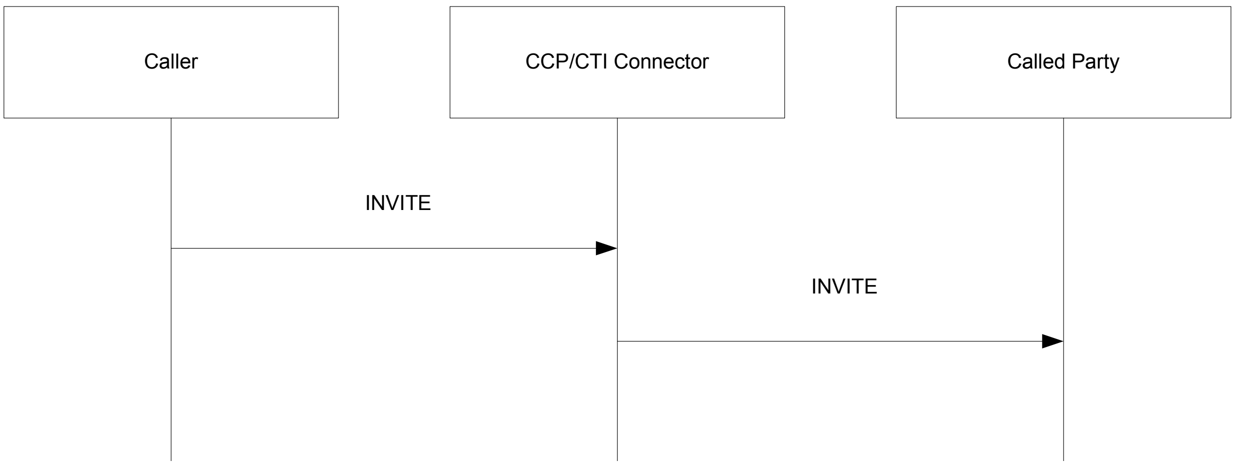


Figure 50: Offer-less Early Join Example

Receiving SDP

CCP and CTI Connector perform the following verifications when receiving an SDP offer:

1. If the `offer-answer-support` parameter equals `true`, the number of media lines in the offer can not be less than previously negotiated, or the call will be terminated.
2. If the `offer-answer-support` equals `false`, and the new SDP contains more media lines than previously received, the new media lines are ignored. They will not be passed to any other calls that are joined.

CCP and CTI Connector perform the following verification when receiving an SDP answer:

- If the `offer-answer-support` equals `true`, the number of media lines in the SDP answer must be the same as the offer sent, or the call will be terminated.

Configuring Device Profiles

The Call Control Platform and CTI Connector use device profiles to determine the behavior of the devices with which they interacts, in order to produce the most appropriate SIP messages.

Device Profile Configuration File

Device profiles are defined in the `<Call Control Platform Installation Directory>\config\ccpccxml_provision.dat` and `<CTI Connector Installation Directory>\config\CTIC_Provision.dat` files, which are installed when you install the Call Control Platform, and the CTI Connector.

For the format and syntax of device profile entries in the configuration file, see Procedure: Provisioning Device Profiles, on page 494.

Device Classes

Table 101 describes the properties that define the device classes of the device profiles.

Table 101: Device Profile Class Properties

Property	Description	Valid Values
connectionless-sdp-type	Support for Connectionless SDP. Indicates the mechanism that should be used to indicate `null SDP` (in other words, no media streams) to the device.	• `hold`—Use `hold SDP` (for example, `c=IN IP4 0.0.0.0` or `c=IN IP6 0.0.0.0.0.0.0.0`). • `non-routable`—Use SDP with a non-routable connection (for example, `c=IN IP4 1.1.1.1` or `c=IN IP6 1.1.1.1.1.1.1.1`).
distinct-send-recv-support	Support for `send-recv` media lines. Indicates if the offer-answer user agent supports a `sendonly` media line and a `recvonly` media line on separate connections.	• `true`—The Device supports a `sendonly` media line and a `recvonly` media line on separate connections. • `false`—The Device does not support a `sendonly` media line and a `recvonly` media line on separate connections.
multiple-recvonly-support	Support for multiple `recvonly` media lines. Indicates if the offer-answer user agent supports receiving multiple `recvonly` media lines.	• `true`—The Device supports receiving the `recvonly` attribute in multiple SDP media lines. • `false`—The Device does not support receiving the `recvonly` attribute in multiple SDP media lines.

Table 101: Device Profile Class Properties (Continued)

Property	Description	Valid Values
nomedia-SDP-support	Support for receiving SDP containing no media lines. Indicates if the offer-answer user agent supports receiving an SDP containing no media lines. The device answers with an SDP with no media lines and remains in a state with no media connections until a `re-INVITE`.	• `true`—The Device supports receiving SDP without media lines. • `false`—The Device does not support receiving SDP without media lines.
offer-answer-support	Support for the Offer-Answer model. Indicates if the device supports the offer-answer model described in RFC 3264 (in other words, whether the device will respond to SDP offers with an answer according to the rules defined in the RFC).	• `true`—The Device supports the offer-answer model. • `false`—The Device does not support the offer-answer model.
offer-less-invite-support	Support for `INVITE` requests that do not contain an SDP offer. Indicates if the device supports an `INVITE` request that does not contain an SDP offer. The device responds with an SDP offer in its response to the `INVITE`.	• `true`—The Device supports receiving `INVITE` requests that do not contain an SDP offer. • `false`—The Device does not support receiving `INVITE` requests that do not contain an SDP offer.
recvonly-support	Support for `recvonly` media lines. Indicates if the offer-answer user agent supports the `a=recvonly` media line attribute in SDP messages that it receives.	• `true`—The Device supports receiving the `recvonly` attribute in the SDP media line. • `false`—The Device does not support receiving the `recvonly` attribute in the SDP media line.

Table 101: Device Profile Class Properties (Continued)

Property	Description	Valid Values
restricts-media-source	Support for receiving media only from the User Agent (UA) to which media is being sent. Indicates if the offer-answer user agent is able to receive media from a UA other than the UA to which it is sending media.	• `true`—The Device supports receiving media from the UA to which media is being sent, and only from that UA. • `false`—The Device supports receiving media from the UA to which media is being sent, as well as from other UAs.
sendonly-support	Support for `sendonly` media lines. Indicates if the offer-answer user agent supports the `a=sendonly` media line attribute in SDP messages that it receives.	• `true`—The Device supports receiving the `sendonly` attribute in the SDP media line. • `false`—The Device does not support receiving the `sendonly` attribute in the SDP media line.
unjoined-initial-answer-pref	Indicates the preferred method for performing an answer during an initial `INVITE` when no bridges have been established.	• `connectionless-SDP`—If the value of `connectionless-sdp-type` is anything other than `none`, a response with the specified connectionless-SDP type will be sent to the endpoint. If the value of `connectionless-sdp-type` is `none`, there is no preferred method (this parameter is treated as if the preference was `none`). • `reject-media`—If the value of `offer-answer-support` is `true`, a response with SDP that rejects all media lines (`0` port) will be sent to the endpoint. Otherwise, there is no preferred method (this parameter is treated as if the preference was `none`).

Table 101: Device Profile Class Properties (Continued)

Property	Description	Valid Values
unjoined-initial-answer-pref (continued)		• nomedia-SDP—If the value of nomedia-SDP-support is true and the value of offer-answer-support is false, a response with no-media SDP will be sent to the endpoint. Otherwise, there is no preferred method (this parameter is treated as if the preference was none). • none—No preferred method. Other device profile parameters determine the method.
unjoined-initial-offer-pref	Indicates the preferred method for performing an initial INVITE without establishing bridges.	• offer-less—If the value of offer-less-invite-support is true, an INVITE without an offer will be sent to the endpoint. Otherwise, there is no preferred method (this parameter is treated as if the preference was none). • connectionless-SDP—If the value of connectionless-sdp-type is anything other than none, an INVITE with the specified connectionless-SDP type will be sent to the endpoint. If the value of connectionless-sdp-type is none, there is no preferred method (this parameter is treated as if the preference was none). • none—No preferred method. Other device profile parameters determine the method.

Unjoined-initial-offer-pref

The unjoined-initial-offer-pref parameter controls the behavior when generating the initial INVITE for an outbound call if the call is not explicitly joined to any other calls.

If this parameter is set to none or does not exist in the device profile, the following methods are used in the order given if enabled in the device profile:

1. offer-less-invite-support
2. connectionless-sdp

If neither method is enabled, the call will fail.

If multiple methods are supported by the device, the `unjoined-initial-offer-pref` parameter specifies the method to use.

During software initialization, the parameter has the following verifications, and if it detects an error, the value of `unjoined-initial-offer-pref` will default to `none`.

- `unjoined-initial-offer-pref` equals `offer-less`

  Required—`offer-less-invite-support` equals `true`

  Otherwise—`unjoined-initial-offer-pref` equals `none`
- `unjoined-initial-offer-pref` equals `connectionless-sdp`

  Required—`connectionless-SDP-type` does not equal `none`

  Otherwise—`unjoined-initial-offer-pref` equals `none`
- `unjoined-initial-offer-pref` equals `none`

  `unjoined-initial-offer-pref` equals `offer-less` if `offer-less-invite-support` equals `true`

  Otherwise—`unjoined-initial-offer-pref` equals `connectionless-sdp` if `connectionless-SDP-type` does not equal `none`

unjoined-initial-answer-pref Configuration

The `unjoined-initial-answer-pref` parameter controls the behavior when generating the initial SIP response for an inbound call if the call is not already joined to another call.

This parameter also applies to an outbound call if the `offer-less` INVITE method is used to generate the INVITE. In this case, the received response will contain an offer, and the `unjoined-initial-answer-pref` parameter controls how the SDP answer is generated in the ACK message.

If this parameter is set to `none` or does not exist in the device profile, the following methods are used in the order given for answering an SDP offer:

Note: This applies when receiving an `offer-less` INVITE for an inbound call.

1. `connectionless-sdp-type`
2. `reject-media` (if `offer-answer-support` equals `true`)
3. `nomedia-sdp-support` (if `offer-answer-support` equals `false`)
4. Call Rejected with 488

If the `unjoined-initial-answer-pref` parameter is set to `none` or does not exist in the device profile, the following methods are used in the order given for generating an SDP offer:

1. `connectionless-sdp-type`

2. `nomedia-sdp-support`
3. Call Rejected with 488

If multiple methods are supported by the device, the `unjoined-initial-answer-pref` specifies the method to use.

During software initialization, this parameter has the following verifications, and if it detects an error, the value of `unjoined-initial-answer-pref` will default to `none`

- `unjoined-initial-answer-pref` equals `connectionless-sdp`

  Required—`connectionless-SDP-type` does not equal `none`

  Otherwise—`unjoined-initial-answer-pref` equals `none`

- `unjoined-initial-answer-pref` equals `reject-media`

  Required—`offer-answer-support` equals `true`

  Otherwise—`unjoined-initial-answer-pref` equals `none`

- `unjoined-initial-answer-pref` equals `nomedia-sdp`

  Required—`nomedia-sdp-support` equals `true` and `offer-answer-support` equals `false`

  Otherwise—`unjoined-initial-answer-pref` equals `none`

Customizing Device Profiles

Call Control Platform and CTI Connector are preprovisioned with a number of default device profiles, which reflect Genesys' knowledge of the behavior of some commonly used SIP devices, including the GVP Media Control Platform.

Your deployment may require the Call Control Platform or the CTI Connector interface with a SIP device that is not currently defined in the default device profile provisioning file. If the SIP device attributes do not match any of the preprovisioned device profiles, you must create a new device profile, or else modify an existing one, to match the actual attributes supported by the SIP device.

- If the SIP request from the unknown device includes the `User-Agent` header, or another header that the Call Control Platform or the CTI Connector can use to identify the device, Genesys recommends that you create a new device profile.
- If the SIP request does not include headers that can be used for identification purposes, calls from the unknown device will use one of the default device profiles (Default Inbound, Default Outbound, Default Dialog, or Default Conference). In this case, if you wish to support the unknown device, you must modify parameters in the default device profile(s).

  For example, if an unknown SIP device that does not support Offer-Answer makes an inbound call, the call will fail unless you change the `offer-answer-support` parameter for the Default Inbound device profile from `true` to `false`.

> **Tip:** To verify which device profile was used for a failed call, use the log files at debug level: Search for *SelectProfile,* and match the incoming `INVITE` to the device profile selection. Then review the parameter values for that profile to identify the parameters you need to change.

The following procedure describes how to modify the device profile provisioning file.

Procedure: Provisioning Device Profiles

Purpose: To modify the `ccpccxml_provision.dat` or the `CTI_Provision.dat` file to enable the Call Control Platform or the CTI Connector to interact with non-default SIP devices.

Prerequisites

- The Call Control Platform or the CTI Connector has been installed in a directory for which you have write access permissions.
- You have identified the required attributes for the device profile(s) you want to create or modify.

Start of procedure

1. Back up the existing `provision` files, in case you later want to restore the original settings.
2. Open the `<Call Control Platform Installation Directory>\config\ccpccxml_provision.dat or the <CTI Connector Installation Directory>\config\CTIC_Provision.dat` file in a text editor.
3. Add or modify device profile entries as required for your deployment.

   The format for each device profile entry is the following:

   ```
   <entry id="<Entry ID>" type="401" name="CCXML Device Profile">
   <Precedence>
   <Profile Name>
   <Device Profile Class Name>
   <# of properties>
   <Property Name 1> <Property Value 1>
   ...
   <Property Name m> <Property Value m>
   <SIP Header Name> <Regex>
   </entry>
   ```

   Where:

- `<Entry ID>` is an unsigned integer that uniquely identifies the entry.
- `<Precedence>` is an unsigned integer that indicates the order of priority in which the Call Control Platform or the CTI Connector will attempt to assign the device profile. The larger the value, the lower the precedence (`1` is the highest). A value of `0` (zero) indicates default. Except for `0`, precedence values must be unique.
- `<Profile Name>` is a unique alphanumeric string that identifies the device profile. Spaces are allowed.
- `<Device Profile Class Name>` is the class of device profiles to which this device profile belongs.
- `<# of properties>` is the number of properties that are defined in the entry.
- `<Property Name x>` is a non-empty alphanumeric string that must be unique within the device profile. Spaces are not allowed.
- `<Property Value x>` is a non-empty alphanumeric string that specifies the value of the `<Property Name x>` property. Spaces are not allowed.
- `<SIP Header Name>` is an optional parameter that, if defined, specifies the SIP header from the incoming SIP `INVITE` that the Call Control Platform or the CTI Connector will attempt to match, to assign the device profile for inbound connections. Spaces are not allowed.
- `<Regex>` is the expression that the Call Control Platform or the CTI Connector will attempt to match in the specified SIP header from the incoming SIP `INVITE`. If `<SIP Header Name>` is empty, `<Regex>` is also empty.

Note: The angle brackets in the first and last lines of each device profile entry are required characters in the syntax.

4. Save the file.
5. Restart the Call Control Platform or the CTI Connector.

End of procedure

Default Device Profiles

Tables 102 and 103 summarize the default device settings, by profile.

The format of each entry in the profile definition file is:

```
<entry id="Entry ID" type="401" name="CCXML Device Profile">
Precedence
Profile Name
SIP Device
Number of properties
PropertyA ValueA
PropertyB ValueB
...
User-Agent User-Agent
</entry>
```

Where:

- The angle brackets are a necessary part of the syntax.
- Italic text indicates placeholders for items that are listed in Tables 102 and 103.
- `SIP Device` is the Device Profile Class Name.
- `User-Agent` is the SIP Header Name.

You may optionally add a short description in each property line.

Table 102: Default Device Profile Settings

Item	Value for						
	Cisco Gateway	Default In-bound	Default Out-bound	Default Con-ference	Default Dialog	Audio-codes Gate-way	Con-vedia Media Server
Entry ID	1	2	3	4	5	6	7
Precedence	1	0	0	0	0	2	3
Profile Name	Cisco Gateway	Default Inbound	Default Outbound	Default Con-ference	Default Dialog	Audio-codes Gate-way	Conv-edia Media Server
Number of properties	12	10	10	10	10	12	10

Table 102: Default Device Profile Settings (Continued)

Item		Value for						
		Cisco Gateway	Default In-bound	Default Out-bound	Default Con-ference	Default Dialog	Audio-codes Gate-way	Con-vedia Media Server
Properties	sendonly-support	false	true	true	true	true	true	true
	recvonly-support	false	true	true	true	true	true	true
	distinct-send-recv-support	false	true	true	true	true	false	false
	multiple-recvonly-support	false	true	true	true	true	false	false
	restricts-media-source	false	true	true	true	true	false	true
	connectionless-sdp-type	hold	hold	non-routable	non-routable	hold	non-routable	hold
	offer-answer-support	false	true	false	true	true	false	false
	nomedia-SDP-support	false	true	false	false	false	false	true
	offer-less-invite-support	true	true	true	true	true	true	true
User-Agent		Cisco	Inbound	Outbound	Con-ference	Dialog	Audio-codes	Con-vedia

Table 103: Default Device Profile Settings

Item							
	X-Lite	Brook-trout Snow-shore	GVP MCP	Audio-codes MP104	Eye Beam	Kapanga	Dialogic Media Gateway
Entry ID	8	9	10	11	12	13	15
Precedence	4	5	6	7	8	9	15
Profile Name	X-Lite	Brooktrout Snowshore	GVP MCP	Audio codes MP104	Eye Beam	Kapanga	Dialogic Media Gateway

Table 103: Default Device Profile Settings (Continued)

Item		X-Lite	Brook-trout Snow-shore	GVP MCP	Audio-codes MP104	Eye Beam	Kapanga	Dialogic Media Gateway
Number of properties		12	10	12	10	12	12	12
Properties	sendonly-support	true	false	true	true	true	true	true
	recvonly-support	true	false	true	true	true	true	true
	distinct-send-recv-support	true	false	true	false	true	false	false
	multiple-recvonly-support	true	false	true	true	true	false	true
	restricts-media-source	true	true	true	false	true	true	false
	connectionless-sdp-type	non-rout-able	hold	hold	hold	non-routable	non-routable	non-routable
	offer-answer-support	false	true	true	true	false	false	false
	nomedia-SDP-support	false	false	true	false	false	false	false
	offer-less-invite-support	true	true	true	false	true	true	true
User-Agent		X-Lite	Brooktrout	GVP MCP	Audio codes MP104	Eye Beam	Kapanga	Dialogic Media Gateway

Appendix

F VAR API

This appendix describes the Voice Application Reporter (VAR) application programming interface (API). It contains the following sections:

Overview

In GVP 8.1, Voice Application Reporter is provided by the GVP Reporting Server. The Reporting Server provides access to a web service (VAR Reporting Service) that generates VAR reports. VAR statistics are computed by the Reporting Server based on the series of events the Media Control Platform produces while it is executing VoiceXML applications. The platform generates some of these events when it encounters VAR-specific `<log>` tag extensions.

VAR Records

The following section describes the information that is contained in the three main types of VAR records that are stored by the Reporting Server.

VAR Call Detail Records

VAR Call Detail Records (CDRs) are used to record information about individual MCP sessions. They contain the following VAR specific information:

- MCP Session ID
- MCP Component ID
- VAR Call Result

- Call End State
    - User End—Indicates that the MCP session ended because the caller hung up first.
    - Application End—Indicates that the MCP session ended because the VoiceXML application hung up first.
    - System Error—Indicates that the MCP session ended because of a system error.
    - Unknown—Indicates that the MCP did not log an end state for the session.
    - VAR Call Notes logged by the VXML Application.
    - Access to the corresponding MCP CDR (for additional information).

VAR Call Detail Records are used to generate VAR CDR reports.

VAR Call Summary Records

VAR Call Summary records provide aggregated statistics (specifically call count and cumulative call length) for MCP sessions with details broken down by VoiceXML Application ID, Call State, VAR Call Result, and VAR Call Reason.

The records are used to derive Call Summary statistics which are useful for showing the number of calls in each call result category. For example, if you want to know the number of calls that failed for Application ID - 101, or the different reasons that are associated with these failed calls, these are the statistics that you retrieve. For more information on viewing Call Summary records, see "VAR Call Completion Summary" on page 381.

VAR IVR Action Summary Records

GVP 8.1 uses the concept of an IVR Action to refer to a sequence of VoiceXML executions that has been identified by the application developer.

The Reporting Server processes information about the IVR Actions that are executed by the IVR Application, and then aggregates the data into summary records. These records provide information about the IVR Actions that were processed during a given time period. Theses records are broken down by time period, application ID, IVR Action ID, IVR Action result, and IVR Action reason. The total IVR Action count, the total unique call, the total last IVR Action count (for each result type), the total count (for each result type), and end state are also stored in these records.

Action Summary records are used to generate IVR Action reports. For more information on viewing IVR Action Summary records, see "VAR IVR Action Summary" on page 384.

VoiceXML <log> Extensions

The Media Control Platform and the Reporting Server support extensions to the VoiceXML `<log>` tag in order to generate Voice Application Reporter statistics. These extensions allow application developers to add specific VAR details to their VoiceXML pages. A given VoiceXML call can have IVR actions, a call result, notes that are associated with a specific action, and custom name/value pairs (variables). The extensions are described in the following sections.

com.genesyslab.var.CallResult

The platform provides an extension to the `<log>` tag, using `label=com.genesyslab.var.CallResult`. This allows application developers to specify a result for a call using the following format:

```
<log label="com.genesyslab.var.CallResult">result[|reason]</log>
```

The following code snippet is an example that uses the `com.genesyslab.var.CallResult` element.

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
 <form id="weather_info">
  <block>
   Welcome to the weather information service.
  </block>
  <field name="state">
   <prompt>What state?</prompt>
   <grammar src="state.grxml" type="application/srgs+xml"/>
   <catch event="help">
    Please speak the state for which you want the weather.
   </catch>
   <catch event="exit">
   <log label="com.genesyslab.var.CallResult"> FAILED |exit</log>
    <exit/>
   </catch>
  </field>
  <field name="city">
   <prompt>What city?</prompt>
   <grammar src="city.grxml" type="application/srgs+xml"/>
   <catch event="help">
    Please speak the city for which you want the weather.
   </catch>
   <catch event="exit">
   <log label="com.genesyslab.var.CallResult"> FAILED |exit</log>
    <exit/>
   </catch>
  </field>
```

```
  <block>
     <log
     label="com.genesyslab.var.CallResult">SUCCESS|reported</log>
       <submit next="/servlet/weather" namelist="city state"/>
</block>
 </form>
</vxml>
```

The value of the result must either SUCCESS, FAILED or UNKNOWN (default). The result is not case-sensitive, and preceding or trailing spaces are ignored. If the VoiceXML application specifies a call result other than those that are mentioned, or if no call result is specified, the call result is set to UNKNOWN. A CallResult <log> tag can be used more than once in an application, but only the one that is processed last will be recorded by the Reporting Server.

The maximum length of the reason is 256 bytes, and any text beyond the limit will be truncated.

com.genesyslab.var.ActionStart and com.genesyslab.var.ActionEnd

An IVR Action can be used to define key transaction points within a VoiceXML application, and associate those transactions with a given action ID. An Action starts when a <log> tag is executed with the label attribute set to com.genesyslab.var.ActionStart. The Action ends when a <log> tag is executed with the label attribute set to com.genesyslab.var.ActionEnd.

The following code snippet shows the syntax to start an IVR Action:

```
<log
label="com.genesyslab.var.ActionStart">actionID[|parentID=<PID>]</log>
```

Application developers must be aware of the following:

- The actionID is the ID of the action that is being started. If this action is nested inside of an active action, the ID of the parent action (PID) must also be included.
- The actionID and PID can be any valid UTF8 string that does not contain spaces or pipes, and it is restricted to a maximum of 64 bytes.
- Spaces are ignored.

The following code snippet shows the syntax to end an IVR Action:

```
<log
label="com.genesyslab.var.ActionEnd">actionID[|result[|reason]]</log>
```

Application developers must be aware of the following:

- The actionID is the ID of the action that is being started.
- The ID can be any valid UTF8 string that does not contain spaces or pipes, and it is restricted to a maximum of 64 bytes.
- Spaces are ignored.

- The result must be either `SUCCESS`, `FAILED` or `UNKNOWN` (default). The result is not case sensitive, and proceeding and trailing spaces are ignored.
- The reason is optional. The maximum length of the reason is 256 bytes, and any text beyond that limit is truncated.

Note: The Reporting Server will implicitly end actions in certain cases (see “Implicit End” on page 504.

The following code snippet is an example using the `com.genesyslab.var.ActionStart` and the `com.genesyslab.var.ActionEnd` elements.

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
 <form id="weather_info">
  <block>
   Welcome to the weather information service.
   <log label="com.genesyslab.var.ActionStart">action_1</log>
  </block>
  <field name="state">
   <prompt>What state?</prompt>
   <grammar src="state.grxml" type="application/srgs+xml"/>
   <catch event="help">
    Please speak the state for which you want the weather.
   </catch>
   <catch event="exit">
    <log label="com.genesyslab.var.ActionEnd">action_1|FAILED|user
exited</log>
    <log label="com.genesyslab.var.ActionNotes">action_1|ended in
state dialog</log>
    <exit/>
   </catch>
  </field>
  <field name="city">
   <prompt>What city?</prompt>
   <grammar src="city.grxml" type="application/srgs+xml"/>
   <catch event="help">
    Please speak the city for which you want the weather.
   </catch>
   <catch event="exit">
    <log label="com.genesyslab.var.ActionEnd">action_1|FAILED|user
exited</log>
    <log label="com.genesyslab.var.ActionNotes">action_1|ended in
city dialog</log>
    <exit/>
   </catch>
  </field>
```

```
  <block>
   <log label="com.genesyslab.var.CustomVar">state|<value
expr="state"/></log>
   <log label="com.genesyslab.var.CustomVar">city|<value
expr="city"/></log>
   <log
label="com.genesyslab.var.ActionEnd">action_1|SUCCESS|Weather
Accessed</log>
   <submit next="/servlet/weather" namelist="city state"/>
  </block>
 </form>
 <catch event=".">
  <log
label="com.genesyslab.var.ActionEnd">action_1|FAILED|unexpected
event</log>
  <exit/>
 </catch>
</vxml>
```

In some cases an `ActionEnd` and `ActionStart` labels are ignored by the Reporting Server if:

- The specified `parentID` is not the ID of an active Action (an Action that has started, but that has not yet ended).
- Its `actionID` is not the ID of an active IVR Action.
- Its result is not one of `SUCCESS`, `FAILED`, or `UNKNOWN`.

Implicit End

An IVR Action starts when a `<log>` tag with an `ActionStart` label is executed. However, if an IVR Action is currently active, starting a new Action will automatically cause the previously active Action to end (implicit end), unless the new Action designates the previously active Action as its parent.

Ending an IVR Action will cause the Reporting Server to end all of its child Actions implicitly. In addition, when a call ends, all active IVR Actions will be ended implicitly.

Last IVR Action

The last IVR Action represents the last IVR Action that was executed for a given call. VAR IVR Action summary records allow the Reporting Server to keep track of the total number of times that a specific IVR Action was considered the last action that was executed.

If IVR Actions are still active at the end of a call, the most-nested IVR Action that was still in progress at call end is designated as the last action. For example, in the following code snippet, `action_3` will be designated the last IVR action because the application exits while the Action is still in progress.

```
<block>
   Testing Last IVR Action
```

```
   <log label="com.genesyslab.var.ActionStart">action_1</log>
   <log label="com.genesyslab.var.ActionStart">action_2|
parentID=action_1</log>
   <log label="com.genesyslab.var.ActionStart">action_3|
parentID=action_2</log>
   <exit/>
   <log
label="com.genesyslab.var.ActionEnd">action_3|SUCCESS|test1</log>
   <log
label="com.genesyslab.var.ActionEnd">action_2|SUCCESS|test2</log>
   <log
label="com.genesyslab.var.ActionEnd">action_1|SUCCESS|test3</log>
  </block>
```

In the following code snippet, `action_1` will be designated the last IVR Action because the application exits after this Action ends.

```
  <block>
   Testing Last IVR Action
   <log label="com.genesyslab.var.ActionStart">action_1</log>
   <log label="com.genesyslab.var.ActionStart">action_2|
parentID=action_1</log>
   <log label="com.genesyslab.var.ActionStart">action_3|
parentID=action_2</log>
   <log
label="com.genesyslab.var.ActionEnd">action_3|SUCCESS|test1</log>
   <log
label="com.genesyslab.var.ActionEnd">action_2|SUCCESS|test2</log>
   <log
label="com.genesyslab.var.ActionEnd">action_1|SUCCESS|test3</log>
   <exit/>
  </block>
```

com.genesyslab.var.ActionNotes

The platform provides an extension to the `<log>` tag that allows application developers to associate free-form notes with an IVR Action.

The following code snippet shows the syntax for action notes:

```
<log label="com.genesyslab.var.ActionNotes">actionID|notes</log>
```

Application developers must be aware of the following:

- The `actionID` is the ID of the action.
- The ID can be any valid UTF8 string that does not contain spaces or pipes, and it is restricted to a maximum of 64 bytes.
- Spaces are ignored.
- Notes are limited to 4096 bytes, and cannot be empty. Any content beyond the limit is truncated.

IVR Action Notes can be logged during or after the specified action is ended.

com.genesyslab.var.CallNotes

The platform provides an extension to the `<log>` tag that allows application developers to associate free-form notes with a call. Call notes are limited to 4096 bytes, and cannot be empty. Any content beyond the limit will be truncated.

the following code snippet shows the syntax for call notes:

```
<log label="com.genesyslab.var.CallNotes"> notes</log>
```

com.genesyslab.var.CustomVar

The platform provides an extension to the `<log>` tag that allows application developers to associate custom name/value pairs with a call.

The following code snippet shows the syntax for custom variables:

```
<log label="com.genesyslab.var.CustomVar">name|value</log>
```

Application developers must be aware of the following:

- The name is any valid UTF8 string that does not contain spaces or pipes, and it is restricted to a maximum of 64 bytes. Spaces are ignored.
- The value is any valid UTF8 string, to a maximum of 256 bytes. Spaces are significant.
- If it is not formatted properly, the custom variable data is logged as a simple message, and will not impact VAR statistics.

Custom variables can be specified at any point in a VoiceXML application. You can have a maximum of eight configured custom variables for any given call by setting the `[ems]dc.default.max.custom_vars` parameter in the Media Control Platform Application object in Genesys Administrator. Any custom variables that are specified beyond the maximum are discarded by the system.

Appendix

G Video Support

This appendix describes the Genesys Voice Platform (GVP) supported video formats.

It contains the following section:

Overview

GVP includes support for the following video applications:

- Video voicemail
- Video conferencing and conferencing management
- Entertainment applications

Because video support is not defined as part of VoiceXML 2.1, the VoiceXML tags, `<audio>` and `<record>`, are enhanced to allow development of video play and video record applications—for example, video voicemail.

Supported Protocols and Specifications

GVP supports the following video formats:

- AVI container files with H.263 encoded video and G.711 encoded audio (8kHz).
- 3GPP container files with H.263 encoded video and AMR/AMR-WB encoded audio (8kHz).
- 3GPP container files with H.264 encoded video and AMR/AMR-WB encoded audio (8KHZ/16KHZ).

Table 104 lists the supported MIME types.

Table 104: MIME Types

<table>
<tr><th>Format</th><th>File Extension</th><th>MIME Type</th></tr>
<tr><td>AVI</td><td>.avi</td><td><code>video/avi;codec or video/x-avi=<audio codec>;videocodec=<video codec>;</code></td></tr>
<tr><td>3GP</td><td>.3gp</td><td><code>video/3gpp;codec=<audio codec>;videocodec=<video codec></code></td></tr>
<tr><td rowspan="2">RAW</td><td>.263</td><td><code>video/H263</code> or <code>video/H263 - 1998</code> or <code>video/x-h263</code></td></tr>
<tr><td>.264</td><td><code>video/H264</code> or <code>video/x-h264</code></td></tr>
<tr><td colspan="3">
• audio_codec for AVI can be ulaw (g.711 mulaw), alaw (g.711 alaw), g729[b]/g729a[b], AMR-NB, AMR-WB, adpcm, pcm 16 (singed linear PCM 16 bit, or pcm (unsigned linear PCM 8 bit)
• audio_codecs for 3GP can be amr (AMR-NB or AMR-WB)
• video_codec for AVI can be h263 (h.263) or h263-1998 (h.263+) or VP8
• video_codec for 3GP can be h263 (h.263) or h263-1998 (h.263+) or h264
</td></tr>
</table>

Notes: H.263 media transport over RTP conforms to Mode A transmission as defined in RFC2190.

H.263+ media transport over RTP conforms to RFC2429.

H.264 media transport over RTP conforms to RFC3984 bis.

The VP8 transcoder is not required if the file being played contains VP8 video,and/or the caller has requested VP8 video.

Video Features

GVP supports many features for video recording and playback.

VoiceXML Features

GVP video supports the following VoiceXML features:

- Video file playback (including embedded audio)
- Video file record (including embedded audio)
- Video text overlay
- DTMF recognition
- Speech recognition
- Speech and DTMF barge-in
- Prompt queuing

- Caching

Video Playback

The following snippet of code provides an example of video playback:

```
<?xml version="1.0"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
<meta name="application" content="Video Playback
Example"/>
   <form id="Welcome">
   <block name="Hello">
   <audio src="builtin:prompts/sting.vox"/>
Which trailer would you like to watch, Science Fiction or
Drama?
   </block>
   <field name="movie">
   <option> Science Fiction </option>
   <option> Drama </option>
   <filled>
         <if cond="movie=='Science Fiction'">
            Here's the trailer for Science Fiction
            <audio src="harrypotter.avi"/>
         <elseif cond="movie='Drama'"/>
            Here's the trailer for Drama
            <audio src="jurassic.avi"/>
         </if>
   </filled>
   </field>
   </form>
</vxml>
```

Video Recording

The following snippet of code provides an example of video recording:

```
<?xml version="1.0"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
<meta name="application" content="Video Recording
Example"/>
<property name="caching" value="safe"/>
<property name="bargein" value="false"/>
<property name="confidencelevel" value="0.45"/>
<property name="loglevel" value="4"/>
<form>
   <record name="video_message" beep="true" maxtime="30s"
finalsilence="5s" dtmfterm="true"
dest="RecordedFile/"
```

```
type="video/avi;codec=pcm16;videocodec=h263">
   <prompt> please re cord your message </prompt>
   <filled>
   Here is your video message <value expr="video_message"/>
   </filled>
   </record>
</form>
</vxml>
```

Video Text Overlay

GVP allows the VoiceXML application to specify the text that will be written on top of the video output. Each video file can be specified with pieces of text and other attributes that are used to control how the text is displayed. Each piece of text and its corresponding attributes are represented as an element in the `videotxt` array. The GVP `videotextexpr` extension attribute can be used to specify a number of different attributes for the text.

The following snippet of code provides an example of video text overlay:

```
<block name="setupTxt">
 <var name="videotxt" expr="new Array(1)"/>
 <assign name="videotxt[0]" expr="new Object()"/>
 <assign name="videotxt[0].xoffset" expr="0"/>
 <assign name="videotxt[0].yoffset" expr="0"/>
 <assign name="videotxt[0].text" expr="'My text on Video.'"/>
 <assign name="videotxt[0].fontsize" expr="60"/>
 <assign name="videotxt[0].fontwidth" expr="0"/>
 <assign name="videotxt[0].fontname" expr="'Courier New'"/>
 <assign name="videotxt[0].fontstyle" expr="'Regular'"/>
 <!-- Yellow color with black background -->
 <assign name="videotxt[0].fontcolor" expr="'ffff00'"/>
 <assign name="videotxt[0].bgcolor" expr="'000000'"/>
 <audio src="audiofile/superman-12s.avi" gvp:videotextexpr="videotxt"/>
</block>
```

For more information about the video text overlay feature, see the description of the `videotextexpr` attribute of the VoiceXML `<audio>` element in the *Genesys Voice Platform 8.1 VoiceXML Help.*

Advanced Features

GVP has many advanced features that work well with video. For more information, see the “Tutorials” in the *Genesys Voice Platform 8.1 Genesys VoiceXML Reference Help* file.

VCR Controls

GVP allows the caller to navigate within an audio or video stream using DTMF keys. Functions include pause, resume, skip forward, skip backward, as well as other features.

Note: When VCR control is used with video, the video will not be updated (and appear out-of-sync) until an I-frame is played from the Media Control Platform.

Advanced Barge-in

GVP allows intelligent prompt playback, and confirmation of what prompts the caller has heard with the reporting of barge-ins offsets based on time and marks set in the prompt stream.

Conferencing

Video conferencing can be managed by the CCXML and the Media platforms in the following ways:

- Full, or half-duplex conference connections (including listen only, send only, or full bidirectional video and audio).
- Video switching.
- Video pre-select.
- Video based on active (loudest) speaker
- Video mixing (tiled conferencing)

Note: Video mixing is enabled by setting the `[conference] video_output_type` configuration option value to `mixed`. The layout can be controlled by using the `[conference] video_mixer_layouts` configuration option.

The `</join>` attributes are added for specifying video conferencing behavior:

`videoalgorithm="loudest"|"fixed"|"none"(optional)`

Where:

- `loudest` is the video from the active participant.
- `fixed` is a pre-selected video channel.
- `none` disables video conferencing.

Video mixing is enabled by setting the `[conference] video_output_type` configuration option value to `mixed`. The layout that is used can be controlled by using the `[conference] video_mixer_layouts` configuration option.

Full Call Recording

GVP supports recording video interactions on the call.

Media Redirect Transfer

Transfer and media redirect works with video allowing GVP to remain in the call control path while redirecting media to the appropriate endpoints.

SIP and NETANN Access

GVP supports video access through the SIP and NETANN protocols.

Appendix

H Custom Log Sinks

This appendix describes how to develop a custom log sink that can integrate with the Genesys Voice Platform (GVP) logging infrastructure. It contains the following sections:

Overview

The GVP logging infrastructure enables you to develop custom log sinks to filter and process GVP logs according to your specific requirements. A custom log sink can receive all log types and call metrics.

On Windows, a custom sink must be a DLL (Dynamically Linked Library); on Linux, it must be a shared object. The instructions provided in this appendix focus on writing a log sink for C++ using Microsoft Visual Studio 2005 (C++ 2005) for Windows.

Log Sink Interface

A custom log sink must support the proper sink interface so that the logging infrastructure can load it, and execute the methods correctly. The main DLL interface (or export) consists of the `GetSink()` function only.

The following code snippet shows one way to define the function for Visual Studio 2005:

```
extern "C"
{
__declspec(dllexport) gvp::IEMSLogSink* GetSink();
}
```

The `GetSink()` function returns a pointer to a `gvp::IEMSLogSink` object. The following code snippet shows the implementation of the `GetSink()` function:

```
gvp::IEMSLogSink* GetSink()
{
return new myCustomLogSink();
}
```

Where `myCustomLogSink()` is a constructor for a call that is inherited from the `gvp::IEMSLogSink` class.

The following code snippet shows an overview of the abstract `gvp::IEMSLogSink` class:

```
class IEMSLogSink
{
public:
  IEMSLogSink() {m_bInitialized = false;}
  virtual ~IEMSLogSink() {};

  /**
  * @return The version of the Log Sink.
  */
  virtual const char * GetVersion() const = 0;

  /**
  * Initializes the Log Sink.
  * @param[in] pszName The name assigned to this sink.
  * @param[in] llNetworkID The network ID of the current process.
  * @param[in] pConfigService - For internal GVP use.
  * @param[in] pIEMSLog:: - For internal GVP use.
  * @return TRUE (success), FALSE (error).
  */
  virtual bool Initialize(const char* pszName, long long
llNetworkID,
      void* pConfigService, void* pEMSLogInterface) = 0;

        /**
        * Uninitialize the Log Sink.
        */
        virtual bool Uninitialize() = 0;

  /**
  * This function can be used to process log messages (or metrics).
* @param[in] uLogLevel The log level (LOG_0 - LOG_5 or METRICS).
  * @param[in] uLogID The log ID, including module bits and
specifier bits, or the METRICS ID.
      * @param[in] strCallID The caller ID string.
      * @param[in] timeValue The current time value.
  * @param[in] llOriginalSenderID The network ID of the calling
component.
      * @param[in] strData The log message.
 * @param[in] uThreadID The thread ID.
```

```
  * @return TRUE (success), FALSE (error).
  */
  virtual bool LogToSink(unsigned int uLogLevel, unsigned int
uLogID,
        const char* strCallID, const timeval & timeValue,
        long long llOriginalSenderID, const char* strData,
        unsigned long uThreadID = 0) = 0;

  /**
  * For Future use, NOT CURRENTLY USED
  */
  virtual bool ExecuteSinkCommand(const char* pszCommand,
list<string> & listCommandParameters, string & strResult) = 0;

  /**
  * For Future use, NOT CURRENTLY USED
  */
  virtual bool GetSupportedCommands(list<string> & commandList,
list<string> & descriptionList) = 0;

  /**
  * For Future use, NOT CURRENTLY USED
  */
  virtual bool GetSinkHealthData(std::map<std::string, void*> &
healthData, const char* pszAttribute = NULL) = 0;

protected:
  bool m_bInitialized;
};
```

When coding the custom sink, you need to write a new class to inherit from `gvp::IEMSLogSink`. The `GetSink()` function must be written so that it returns an instance of this new class. The implementation specifics of the `gvp::IEMSLogSink` functions in the new class depend on your specific requirements.

Initialization and Shutdown

When the GVP process starts, it creates the GVP Logging object which will load all sinks that are specified in the `[ems] log_sinks` parameter of that process (MCP, CCP or RM). For example, the MCP would load `MYSINK DLL` from the `$InstallationRoot$/bin/mySink.dll` file (along with the other default logs sinks) at startup, if the following parameters are configured:

```
[ems]
log_sinks=MFSINK|DATAC|TRAPSINK|MYSINK
log_dll.MYSINK=$InstallationRoot$/bin/mySink.dll
logconfig.MYSINK = *|*|*
metricsconfig.MYSINK = *
```

During logging initialization, the `mySink.dll`'s `GetSink()` method is called to create the `IEMSLogSink` object pointer within the `mySink.dll`. If successful, the `Initialize()` method is then called for the `IEMSLogSink` object.

In the following initialization example, the `pszName` parameter is `MYSINK`, and the `llNetworkID` is the unique Genesys Management Framework DBID associated with the MCP Application.

```
virtual bool Initialize(const char* pszName, long long llNetworkID,
     void* pConfigService, void* pEMSLogInterface);
```

When the MCP is shut down, the logging system is shut down with the `Uninitialize()` method.

The following uninitialized example is called for the `IEMSLogSink` object:

```
/**
* Uninitialize the Log Sink.
*/
virtual bool Uninitialize();
```

Note: You must use the `Uninitialize()` method to properly shut down your custom logging sink.

For more information on the [ems] log sink parameters, see Configuring Reporting, page 63.

LogToSink() Method

The `LogToSink()` method is called every time a log is sent to the custom sink through GVP Logging.

Table 105 lists and describes the `LogToSink()` method parameters.

Table 105: LogToSink() Method Parameters

Parameter	Description	Valid values and Examples
`unsigned int uLogLevel`	Specifies the type of log.	• `0`—CRITICAL LOGS • `1`—ERROR LOGS • `2`—WARNING LOGS • `3`—NOTICE LOGS • `4`—INFO LOGS • `5`—DEBUG LOGS • `6`—CALL METRICS
`unsigned int uLogID`	Specifies the unique identifier of the log. This includes the module identifier (the 12 most significant bits), and the log specifier (the 20 least significant bits). For metrics logs, the module ID is always `0`.	The module represents the code module that generated the log message (or `0` if a Call Metric). The specifier bits represent an ID associated with the log message within the given module.
`constr char* strCallID`	Specifies the session ID for the MCP and CCP session.	Example string for MCP: `00020023-100003E9`
`constr timeval & timeValue`	Specifies the time, in UTC, the log was created.	`timeval structure`
`long long llOriginalSenderID`	Specifies the unique Genesys Management Framework DBID that is associated with the GVP application the generates the log messages.	`Long long`
`const char* strData`	The string containing the entire log message.	• Metrics example string: `incall_initiated 13:1` • Regular log example string: `Starting Resource Manager`
`unsigned lonlog.g uThreadID`	Specifies the thread ID of the GVP logging thread that the `LogSink()` method uses.	`Unsigned long`

Each time a log is sent to the sink, the above values are passed through the `LogToSink()` method. At this point, the custom sink can perform additional filtering, and process the logs as desired.

Threading and Performance Issues

GVP logging is performed in a separate thread within each component's process. All logging sinks, including custom user defined sinks, carry out their processing within this single logging thread. This can help minimize the impact of logging on the performance of each GVP component, especially when logging sinks block when writing to disk. Nevertheless, be aware that the length of time a sink takes to process a log directly delays other logging sinks.

Building and Linking the Library

The following section describes how to build and link the library for Linux and Windows.

Linux

Building and linking the custom shared object on Linux can be done using g++ (GCC). The following example shows what commands and options to use:

```
g++ -c -g -Wall -D_REENTRANT -D_NO_LARGEFILE64_SOURCE -fPIC
myCustomLogSink.C -o myCustomLogSink.o

g++ -g -Wall -D_REENTRANT -D_NO_LARGEFILE64_SOURCE -fPIC -shared
-Wl, -soname, libMySink.so -o libMySink.so myCustomLogSink.o
-lpthread
```

You will need to use the –`fPIC compiler` option to make the code position independent, and the –`shared linker flag` to tell the linker that this is a shared object. After the shared object file is created, it can be used by MCP, CCP and RM.

Windows

Building and linking on Windows depends on the compiler available. For Microsoft Visual C++, create a Win32 Application - DLL project. Make sure that in the property pages of the project, the `Configuration Properties > General >Configuration Type` is configured as `Dynamic Library (.dll)`, and that the `Configuration Properties > C/C++ >Code Generation > Runtime Library` is configured as `Multi-threaded DLL`.

Appendix

I SSG HTTP Interface

Trigger Applications interact with the Supplementary Services Gateway through HTTP. This Appendix describes the XML schema for HTTP requests and responses:

- POST is used for outbound creation, query, and cancellation requests.
- GET is used to query the status of a previously submitted request.
- DELETE is used to cancel a previously submitted request.

This Appendix contains the following sections:

Creating Outbound Requests

HTTP POST is the only way in which Trigger Applications create outbound requests for the Supplementary Services Gateway. The POST body must conform to the XML request schema that is described in the "Fatal Errors" section.

HTTP Request

The HTTP POST request URI must contain `TenantName` as a query string parameter. The Content-Type of the POST request must be either `text/xml` or `application/xml`. The content can be either a single request, or multiple (bulk) requests. The Supplementary Services Gateway validates the content with the defined schema, and inserts create request(s) into the Database for persistence.

The following examples show how to create requests.

Single Create Request Without CPD Parameters

```
POST /SSG?TenantName=<tenant> HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: en-us
Content-Length: 990
Content-Type: application/xml
Host: 172.24.129.81:9820
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0;
InfoPath.1; .NET CLR 2.0.50727)
Connection: Keep-Alive

<SSGRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

     <CreateRequest Token="T7034" MaxAttempts="2" TimeToLive="123s"
IVRProfileName="Application" Telnum="9884719189"
NotificationURL="http://182.123.12.12/DIR/OutURL.xml">
      </CreateRequest>
</SSGRequest>
```

Single CreateRequest with CPD Parameters

```
POST /SSG?TenantName=<tenant> HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: en-us
Content-Length: 1123
Content-Type: application/xml
Host: 172.24.129.81:9820
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0;
InfoPath.1; .NET CLR 2.0.50727)
Connection: Keep-Alive

<SSGRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <CreateRequest
      IVRProfileName="SSGProfile"
      NotificationURL="http://172.24.129.86/Vamsi/Web/Outcome.xml"
      Telnum="9884719189"
      Token="T7034"
      MaxAttempts="3"
      TimeToLive="12000ms">
      <cpd record="false"
         postconnecttimeout="6000ms"
         rnatimeout="6000ms"
```

```
        preconnect="true"
        detect="all"/>
   </CreateRequest>
</SSGRequest>
```

Bulk CreateRequest with CPD Parameters

```
POST /SSG?TenantName=<tenant> HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: en-us
Content-Length: 1788
Content-Type: application/xml
Host: 172.24.129.81:9820
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0;
InfoPath.1; .NET CLR 2.0.50727)
Connection: Keep-Alive

<SSGRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <CreateRequest Token="Token7034" MaxAttempts="2
      TimeToLive="123s" IVRProfileName="Application"
      Telnum="9884719189"
      NotificationURL="http://182.123.12.12/DIR/OutURL.xml"
      Ani="12345">
      <cpd record="false"
         postconnecttimeout="6000ms"
         rnatimeout="6000ms"
         preconnect="true"
         detect="all"/>
   </CreateRequest>
   <CreateRequest Token="Token7035" MaxAttempts="2"TimeToLive="123s"
      IVRProfileName="Application" Telnum="9884719189"
      NotificationURL="http://182.123.12.12/DIR/OutURL.xml"
      Ani="12345"/>
      <cpd record="false"
            postconnecttimeout="6000ms"
            rnatimeout="6000ms"
            preconnect="true"
            detect="all"/>
      </CreateRequest>
</SSGRequest>
```

Table 106 describes the CreateRequest attributes for HTTP POST.

Table 106: CreateRequest Attributes

Attribute	Description
`Token`	A unique token generated by the TA for each single or bulk create request. When the Supplementary Services Gateway responds to the original `CREATE` request, the response contains the token associated with that request. The Supplementary Services Gateway does not enforce uniqueness of the token. This attribute is mandatory.
`IVRProfileName`	The name of the IVR profile that will be used for this outbound call. IVR profiles used in outbound calls are provisioned in Genesys Administrator and are sent to the Resource Manager through SIP Server. The Supplementary Services Gateway does not perform validation on IVR profiles. This attribute is mandatory.
`Telnum`	The telephone number used to make an outbound call (for SIP requests, it must be the SIP URI). This attribute is mandatory.
`NotificationURL`	An encoded URL sent to the Supplementary Services Gateway from the TA. The Supplementary Services Gateway uses this URL to send asynchronously notifications to the TA indicating the success or failure of the outbound call. This attribute is mandatory.
`TimeToLive`	The length of time (in seconds or milliseconds) that the request stays alive in persistent storage. If the outbound call is not initiated within this time period, no further attempts are made and the Supplementary Services Gateway sends a Notification URL to the TA indicating that call initiation failed. This attribute is mandatory.
`MaxAttempts`	The number of times the SSG attempts to place the outbound call, should it fail. When the maximum number of attempts is reached, no further attempts are made and the Supplementary Services Gateway sends a Notification URL to the TA indicating that call initiation failed. This attribute is mandatory.
`ANI`	The ANI in the outbound call that is presented to the external party. This attribute is optional.

HTTP Response

In most cases, the Supplementary Services Gateway responds to Trigger Applications with a `200 OK` message, and the body contains the result (success or failure) formatted in the XML response schema (see “Result Notification on Failure” on page 541). The Contents-Type of the response is `txt/xml`.

Once the Supplementary Services Gateway validates the HTTP POST request, it generates an internal unique Request ID for each request, and inserts the requests into the Database. The generated Request IDs and the Tokens are passed back to the Trigger Application in the response. For any further communication with the Supplementary Services Gateway (querying status or cancellation), the Trigger Application must use the Request IDs.

If the Supplementary Services Gateway encounters any failure during the HTTP POST validation or insertion into DB, it generates a failure response in the `200 OK`. For a bulk POST request, if parsing or validation fails, then the entire request fails. However, if the parsing and validation succeed, but later there are operational errors (for example, inserting a specific record failed), then the response would contain both success and failure indications.

If the Supplementary Services Gateway receives an HTTP POST request with multiple operations like CREATE, QUERY and CANCEL, and if the maximum number of database records are reached in the middle of processing, the Supplementary Services Gateway will execute all the QUERY and CANCEL operations only. It will insert the records in the database until it reaches the threshold, and for any remaining records, the Supplementary Services Gateway sends a failure response with the reason and the reason code.

The following examples show the possible responses to the CreateRequest.

Single CreateRequest Successful Response

```
HTTP/1.1 200 OK
Date: Tue, 03 Jul 2007 13:42:22 GMT
Content-Length: 347
Content-Type: text/xml
Cache-control: no-cache

<? xml version="1.0" ?>
<SSGResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <ResponseElement ResponseType="SUCCESS" Token="T1001"
      RequestID="123435" />
</SSGResponse>
```

Bulk CreateRequest Successful Response

```
HTTP/1.1 200 OK
Date: Tue, 03 Jul 2007 13:42:22 GMT
Content-Length: 683
Content-Type: text/xml
Cache-control: no-cache

<? xml version="1.0" ?>
<SSGResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <ResponseElement ResponseType="SUCCESS" Token="T1001"
```

```
    RequestID="123435"/>
  <ResponseElement ResponseType="SUCCESS" Token="T1002"
    RequestID="123436"/>
  <ResponseElement ResponseType="SUCCESS" Token="T1003
    RequestID="123437"/>
</SSGResponse>
```

Single/Bulk CreateRequest Failure Response (Entire Request Failed)

```
HTTP/1.1 200 OK
Date: Tue, 03 Jul 2007 13:42:22 GMT
Content-Length: 347
Content-Type: text/xml
Cache-control: no-cache
<?xml version="1.0" ?>

<?xml version="1.0" ?>
<SSGResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <FailureDescription ReasonCode="110" Reason="XML Data Parsing
    Failed: <Failure Reason> at LineNumber: <XXX> and
    ColumnNumber: <YYYY>"/>
</SSGResponse>
```

Bulk CreateRequest Mixed Response

```
HTTP/1.1 200 OK
Date: Tue, 03 Jul 2007 13:42:22 GMT
Content-Length: 623
Content-Type: text/xml
Cache-control: no-cache

<?xml version="1.0" ?>
<SSGResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ResponseElement ResponseType="SUCCESS" Token="T1001"
    RequestID="123435"/>
  <ResponseElement ResponseType="SUCCESS" Token="T1002"
    RequestID="123436"/>
  <ResponseElement ResponseType="FAILURE" Token="T1003"
    ReasonCode="120" Reason="DB Insertion failed"/>
    <ResponseElement ResponseType="FAILURE" Token="T1004"
    ReasonCode="121" Reason="Maximum DBRecords threshold
    reached"/>
</SSGResponse>
```

Table 107 describes the various HTTP status codes that the Supplementary Services Gateway generates:

Table 107: SSG Response Attributes

Attribute	Description

`ResponseType`	• `SUCCESS`—Sent if the `Create` request was added to the database successfully. The token and `RequestID` are passed back to the TA in the response.
	• `FAILURE`—Sent if the `Create` request parsing fails, mandatory attributes are missing, or it is not added to the database. The token, `Reason code,` and `Reason` are passed back to the TA in the response.
	  ◆ If parsing or validation fails, the Failure Description with the `Reason Code` and `Reason` are passed back. The entire `POST` request fails.
	  ◆ If parsing or validation succeeds, but other failures occur (for example, a specific request is not added to the database), the tokens are passed back to the TA.

`Token`	Received from the TA that is sending the request and is associated with the `Create` request.
`RequestID`	A unique internal ID that is generated by the Supplementary Services Gateway for each `Create` request. This attribute is passed back to the TA when requests are successfully added to the database.

`Reason Code,`
`Reason`
	• `TenantName` missing in `RequestURI`
	  ◆ Reason Code: `101`
	  ◆ Reason: `TenantName parameter is missing`

	• `TenantName` empty in `RequestURI`
	  ◆ Reason Code: `102`
	  ◆ Reason: `TenantName value is empty`

	• Value supplied in `TenantName` does not match the `Tenant1.TGDN` parameter
	  ◆ Reason Code: `103`
	  ◆ Reason: `Invalid TenantName`

Table 107: SSG Response Attributes (Continued)

Attribute	Description
	• `HTTP POST` request parsing failed due to invalid XML data ◦ Reason Code: `110` ◦ Reason: `XML Data Parsing Failed:: <Failure reason> at LineNumber: XXX and ColumnNumber: YYY`
	• Database insertion failed ◦ Reason Code: `120` ◦ Reason: `Failed to insert record into DB`
	• Maximum number of records in DB reached ◦ Reason Code: `121` ◦ Reason: `Maximum DBRecords threshohld reached`
	• Shutdown in progress ◦ Reason Code: `130` ◦ Reason: `Unable to process the request. Shutdown in progress`
	• SIP Server connection down ◦ Reason Code: `150` ◦ Reason: `SIP Server Connection is not available`
	• Tenant Resource DN is not in registered state ◦ Reason Code: `151` ◦ Reason: `Tenant Resource DN for the tenant is not registered`
	• HTTP Method is not supported ◦ Reason Code: `400` ◦ Reason: `HTTP Method is not supported`
	• Internal error ◦ Reason Code: `500` ◦ Reason: `Unable to process the request due to Internal Error`

Querying Outbound Request Status

The Trigger Application can make a request to the Supplementary Services Gateway for the status of a previously submitted outbound request. Both HTTP GET (single query) and HTTP POST (single/bulk query) are used. For HTTP POST, the content must conform to the XML request schema that is described in the "Fatal Errors" section.

HTTP Request

The `Tenant Name` and `RequestID` are the only required parameters in the HTTP GET or HTTP POST query request. The `Tenant Name` and `RequestID` must be passed in the query string of HTTP GET method, or in the XML body of the HTTP POST. For the HTTP POST, the Content-Type must be either `text/xml` or `application/xml`.

The following examples show the query requests:

HTTP GET Query Request

```
GET /SSG?RequestID=1234&TenantName=<name>
 Accept: */*
 Accept-Encoding: gzip, deflate
 Host: 172.24.129.81:9800
 User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0;
InfoPath.1; .NET CLR 2.0.50727)
 Cache-Control: max-stale=0
 Connection: Keep-Alive
```

Single HTTP POST Query Request

```
POST /SSG?TenantName=<name> HTTP/1.1
Accept: */*
 Accept-Encoding: gzip, deflate
Accept-Language: en-us
Content-Length: 376
Content-Type: application/xml
Host: 172.24.129.81:9800
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0;
InfoPath.1; .NET CLR 2.0.50727)
Connection: Keep-Alive

<SSGRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <QueryRequest RequestID="1231245"/>
</SSGRequest >
```

Bulk HTTP POST Query Request

```
POST /SSG?TenantName=<name> HTTP/1.1
Accept: */*
 Accept-Encoding: gzip, deflate
Accept-Language: en-us
Content-Length: 489
Content-Type: application/xml
Host: 172.24.129.81:9820
Pragma: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0;
InfoPath.1; .NET CLR 2.0.50727)
Connection: Keep-Alive

<SSGRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <QueryRequest RequestID="1231245"/>
   <QueryRequest RequestID="1000000"/>
</SSGRequest >
```

HTTP Response

In most scenarios, the Supplementary Services Gateway returns HTTP `200 OK` with the content (conforming to the XML response schema as described in the "Fatal Errors" section) to indicate success or failure. The Content-Type of the `200 OK` response is `text/xml`.

The following examples show the possible responses for querying.

Single Query Successful Response

```
HTTP/1.1 200 OK
Date: Tue, 03 Jul 2007 13:42:22 GMT
Content-Length: 392
Content-Type: text/xml
Cache-control: no-cache

<?xml version="1.0" ?>
<SSGResponse>
   <ResponseElement ResponseType="SUCCESS" Token="T1001"
      RequestID="123435"
      TenantName="Environment" IVRProfileName="Application"
      Telnum="11011"
      NotificationURL="http://182.24.129.82/Dir/Response.asp"
      AttemptsMade=4
      MaxAttempts=7 TimeToLive="12000s" TTLRemaining="3477s"
      Status="Waiting to be processed"/>
</SSGResponse>
```

Bulk Query Successful Response

```
HTTP/1.1 200 OK
```

```
Date: Tue, 03 Jul 2007 13:42:22 GMT
Content-Length: 272
Content-Type: text/xml
Cache-control: no-cache

<?xml version="1.0" ?>
<SSGResponse>
   <ResponseElement ResponseType="SUCCESS" Token="T1001"
      RequestID="123435"
      TenantName="Environment" IVRProfileName="Application"
      Telnum="11011"
      NotificationURL="http://182.24.129.82/Dir/Response.asp"
      AttemptsMade=4
      MaxAttempts=7 TimeToLive="12000s" TTLRemaining="3477s"
      Status="Waiting to be processed"/>
   <ResponseElement ResponseType="SUCCESS" Token="T1002"
      RequestID="376489"
      TenantName="Environment" IVRProfileName="Application"
      Telnum="11012"
      NotificationURL="http://182.24.129.82/Dir/Response.asp"
      AttemptsMade=0
      MaxAttempts=7 TimeToLive="12000s" TTLRemaining="11997s"
      Status="Waiting to be processed"/>
</SSGResponse>
```

Single/Bulk QueryRequest Failure Response (Validation Failure)

```
HTTP/1.1 200 OK
Date: Tue, 03 Jul 2007 13:42:22 GMT
Content-Length: 347
Content-Type: text/xml
Cache-control: no-cache

<?xml version="1.0" ?>
<SSGResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <FailureDescription ReasonCode="110" Reason="XML Data Parsing
      Failed:: <Failure Reason> at LineNumber: <XXX> and
      ColumnNumber: <YYYY>"/>
</SSGResponse>
```

Single Query Failure Response :

```
HTTP/1.1 200 OK
Date: Tue, 03 Jul 2007 13:42:22 GMT
Content-Length: 176
Content-Type: text/xml
Cache-control: no-cache

<?xml version="1.0" ?>
<SSGResponse>
   <ResponseElement ResponseType="FAILURE" RequestID="1234"
```

```
        ReasonCode="404"
        Reason="RequestID not found in the Database"/>
</SSGResponse>
```

Bulk Query Mixed Response

```
HTTP/1.1 200 OK
Date: Tue, 03 Jul 2007 13:42:22 GMT
Content-Length: 657
Content-Type: text/xml
Cache-control: no-cache

<?xml version="1.0" ?>
<SSGResponse>
   <ResponseElement ResponseType="SUCCESS" Token="T1001"
      RequestID="123435"
      TenantName="Environment" IVRProfileName="Application"
      Telnum="11011"
      NotificationURL="http://182.24.129.82/Dir/Response.asp"
      AttemptsMade=4
      MaxAttempts=7 TimeToLive="12000s" TTLRemaining="3477s"
      Status="Waiting to be processed"/>
   <ResponseElement ResponseType="FAILURE" RequestID="1234"
      ReasonCode="404"
      Reason="RequestID not found in the Database"/>
</SSGResponse>
```

Table 108 describes the Query attributes.

Table 108: Query Attributes

Attribute	Description
`Response Type`	• `SUCCESS`—Sent if the `Create` request was added to the database successfully. The token and `RequestID` are passed back to the TA in the response. • `FAILURE`—Sent if the `Create` request parsing fails, mandatory attributes are missing, or it is not added to the database. The token, `Reason code`, and `Reason` are passed back to the TA in the response. ◆ If parsing or validation fails, the Failure Description with the `Reason Code` and `Reason` are passed back. The entire `POST` request fails. ◆ If parsing or validation succeeds, but other failures occur (for example, a specific request is not added to the database), the tokens are passed back to the TA.
`Token`	A unique token that is generated by the TA for each single or bulk create request. When the Supplementary Services Gateway responds to the original `CREATE` request, the response contains the token that is associated with that request. The Supplementary Services Gateway does not enforce uniqueness of the token. This attribute is mandatory.
`Request ID`	A unique internal ID that is generated by the Supplementary Services Gateway for each `Create` request. This attribute is passed back to the TA when requests are successful added to the database.
`IVRProfileName`	The name of the IVR profile that will be used for this outbound call. IVR profiles that are used in outbound calls are provisioned in Genesys Administrator and sent to the Resource Manager through SIP Server. The Supplementary Services Gateway does not perform validation on IVR profiles. This attribute is mandatory.
`Telnum`	The telephone number that is used to make an outbound call (for SIP requests, it must be the SIP URI). This attribute is mandatory.
`NotificationURL`	An encoded URL that is sent to the Supplementary Services Gateway from the TA. The Supplementary Services Gateway uses this URL to send asynchronously notifications to the TA indicating the success or failure of the outbound call. This attribute is mandatory.

Table 108: Query Attributes (Continued)

Attribute	Description
`MaxAttempts`	The number of times that the Supplementary Services Gateway attempts to place the outbound call, should it fail. When the maximum number of attempts is reached, no further attempts are made and the Supplementary Services Gateway sends a Notification URL to the TA that indicate that call initiation failed. This attribute is optional.
`Attempts Made`	The total number of times that the Supplementary Services Gateway attempted to place this outbound call.
`TimeToLive`	The length of time (in seconds or milliseconds) that the request stays alive in persistent storage. If the outbound call is not initiated within this time period, no further attempts are made and the Supplementary Services Gateway sends a Notification URL to the TA indicating that call initiation failed. This attribute is optional.
`TTL Remaining`	The length of time (in seconds or milliseconds) that remain for the request to stay alive.
`Status`	The state returned of the current request. The valid states are: • `Waiting to be processed` • `Outbound call failed because TTL expired/TTL expired` • `Outbound call in progress` • `Outbound call has been completed`

Table 108: Query Attributes (Continued)

Attribute	Description
`Reason Code`, `Reason`	• RequestID missing in Request URI ◆ Reason Code = `104` ◆ `Reason = RequestID parameter is missing`
	• RequestID empty in Request URI ◆ Reason Code: `105` ◆ Reason: `RequestID value is empty`
	• RequestID value is not valid ◆ Reason Code: `106` ◆ Reason: Invalid `RequestID value`
	• `HTTP POST` request parsing failed—invalid schema ◆ Reason Code: `110` ◆ Reason: `XML Data Parsing Failed:: <Failure reason> at LineNumber: XXX and ColumnNumber: YYY`
	• Shutdown in progress ◆ Reason Code: `130` ◆ Reason: `Unable to process the request. Shutdown in progress`
	• Supplied Request ID does not match any database records ◆ Reason Code: `404` ◆ Reason: `Request not found in the database`
	• Internal error ◆ Reason Code: `500` ◆ Reason:`Unable to process the request due to Internal Error`

Canceling Outbound Requests

The Trigger Application can request the Supplementary Services Gateway to cancel a pending outbound request. The cancel operation succeeds if the outbound request is not already in progress (for example, a `MakeCall` request has been made to SIP Server). For an in progress outbound request, the cancel operation fails. You can either use the HTTP DELETE (single cancel), or HTTP POST (single/bulk cancel) to cancel the request. For HTTP POST, the content must conform to the XML request schema in the "Fatal Errors" section.

HTTP Request

The `TenantName` and `RequestID` are the only parameters required in the HTTP DELETE or the HTTP POST cancel request. The `TenantName` and `RequestID` must be passed in the query string of the HTTP DELETE or in the XML body of the HTTP POST. For HTTP POST, the Content-Type must be either `text/xml` or `application/xml`.

The following examples show how to cancel outbound requests.

HTTP DELETE for Cancel

```
DELETE /SSG?RequestID=1234&TenantName=<name>
 Accept: */*
 Accept-Encoding: gzip, deflate
 Host: 172.24.129.81:9820
 User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0;
InfoPath.1; .NET CLR 2.0.50727)
 Cache-Control: max-stale=0
 Connection: Keep-Alive
```

HTTP POST for a Single Cancel

```
POST /SSG&TenantName=<name>
Accept: */*
Accept-Encoding: gzip, deflate
Host: 172.24.129.81:9820
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0;
InfoPath.1; .NET CLR 2.0.50727)
Cache-Control: no-cache
Content-Type: application/xml
Content-Length: 273

<SSGRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <CancelRequest RequestID="1231245"/>
</SSGRequest >
```

HTTP POST for a Bulk Cancel

```
POST /SSG&TenantName=<name>
Accept: */*
Accept-Encoding: gzip, deflate
Host: 172.24.129.81:9820
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0;
InfoPath.1; .NET CLR 2.0.50727)
Cache-Control: no-cache
Content-Type: application/xml
Content-Length: 381
```

```
<SSGRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <CancelRequest RequestID="1231245"/>
   <CancelRequest RequestID="1000000"/>
</SSGRequest >
```

HTTP Response

For most scenarios, the Supplementary Services Gateway returns the HTTP `200 OK` with the content indicating success or failure. When using a bulk cancel request, some of the cancel requests might succeed while others might fail. The Content-Type of the `200 OK` response is `text/xml`.

The following examples show how the possible responses to the CancelRequest.

Single CancelRequest Successful Response

```
HTTP/1.1 200 OK
Date: Tue, 03 Jul 2007 13:42:22 GMT
Content-Length: 120
Content-Type: text/xml
Cache-control: no-cache

<?xml version="1.0" ?>
<SSGResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <ResponseElement ResponseType="SUCCESS" RequestID="1231245"/>
</SSGResponse>
```

Bulk CancelRequest Mixed Response

```
HTTP/1.1 200 OK
Date: Tue, 03 Jul 2007 13:42:22 GMT
Content-Length: 488
Content-Type: text/xml
Cache-control: no-cache

<?xml version="1.0" ?>
   <SSGResponse
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <ResponseElement ResponseType="SUCCESS" RequestID="1121245"/>
      <ResponseElement ResponseType="FAILURE" RequestID="1000000"
       ReasonCode="106" Reason="RequestID not found"/>
</SSGResponse>
```

Single and Bulk CancelRequest Failure Response

```
HTTP/1.1 200 OK
Date: Tue, 03 Jul 2007 13:42:22 GMT
Content-Length: 488
Content-Type: text/xml
```

```
Cache-control: no-cache

<?xml version="1.0" ?>
   <SSGResponse
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <FailureDescription ReasonCode="110" Reason=" "XML Data
       Parsing Failed:: <Failure Reason> at LineNumber: <XXX> and
       ColumnNumber: <YYYY>"/>
</SSGResponse>
```

Table 109 describes the various attributes in the SSG cancel response.

Table 109: SSG Cancel Response Attributes

Attribute	Description
ResponseType	• SUCCESS—Sent if the Create request was added to the database successfully. The token and RequestID are passed back to the TA in the response. • FAILURE—Sent if the Create request parsing fails, mandatory attributes are missing, or it is not added to the database. The token, Reason code, and Reason are passed back to the TA in the response. ◦ If parsing or validation fails, the Failure Description with the Reason Code and Reason are passed back. The entire POST request fails. ◦ If parsing or validation succeeds, but other failures occur (for example, a specific request is not added to the database), the tokens are passed back to the TA.
RequestID	A unique internal ID that is generated by the Supplementary Services Gateway for each Create request. This attribute is passed back to the TA when requests are successfully added to the database.
Reason Code, Reason	• RequestID missing in Request URI ◦ Reason Code = 104 ◦ Reason = RequestID parameter is missing
	• RequestID empty in Request URI ◦ Reason Code: 105 ◦ Reason: RequestID value is empty
	• RequestID value is not valid ◦ Reason Code: 106 ◦ Reason: Invalid RequestID value

Table 109: SSG Cancel Response Attributes (Continued)

Attribute	Description
`Reason Code`, `Reason` (continued)	• `HTTP POST` request parsing failed due to invalid XML data ◆ Reason Code: `110` ◆ Reason: `XML Data Parsing Failed:: <Failure reason> at LineNumber: XXX and ColumnNumber: YYY`
	• Shutdown in progress ◆ Reason Code: `130` ◆ Reason: `Unable to process the request. Shutdown in progress`
	• Outbound request identified by RequestID is already in progress or RequestID not found in the database ◆ Reason Code: `140` ◆ Reason: `Outbound request already in progress/RequestId not found in database`
	• Failed to delete record in DB ◆ Reason Code: `401` ◆ Reason: `Database Access Error. Unable to process the request`
	• Internal error ◆ Reason Code: `500` ◆ Reason: `Unable to process the request due to Internal Error`

SSG Database Queue Clearing During a Restart

The InitiatedCallRetryFlag parameter specifies what the Supplementary Services Gateway (SSG) does with requests that are present in the database during a restart.

Table 110: InitiatedCallRetryFlag Settings

Value	SSG Action on requests in the database
0	Considers all INITIATED failed and invokes a notification URL for those requests when they expire.
1	Retries the INITIATED requests. Deletes nothing.
3	Deletes only requests with status NEW (1) and INITITATED (2).
4	Deletes all requests in the database (status can be NEW, INITIATED or PROCESSED).

Single HTTP POST (Create/Query/Cancel)

The Supplementary Services Gateway supports any combination of create, query, and cancel in the same HTTP POST request.

The following example shows a single request:

```
<SSGRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <CreateRequest
        Token="T1001"
        MaxAttempts="2"
        TimeToLive="123"
        IVRProfileName="Application"
        Telnum="9884719189"
     </CreateRequest>
  <QueryRequest RequestID="1000000"/>
  <CancelRequest RequestID="1231245"/>
</SSGRequest >
```

Note: Although TenantName is not required for a query, or cancel request, if the POST content has a create request, the request URI query string must contain the TenantName parameter.

The responses that the Supplementary Services Gateway generates contain a combination of create, query, and cancel responses.

Asynchronous Result Notification

The Supplementary Services Gateway notifies the Trigger Application of the final outcome of the outbound request with the Notification URL (which is a mandatory parameter in the CreateRequest). The Supplementary Services Gateway uses the HTTP GET on the Notification URL, and it appends certain query string parameters regarding the status of the outbound request.

Result Notification on Success

The Supplementary Services Gateway supports positive notification to the Trigger Application when a call between GVP and an external party is successful. The outbound call is marked as successful when the Supplementary Services Gateway receives the `TreatmentApplied` event from SIP Server, and at that time, the request is marked for deletion.

SSG supports positive notification back to the TA in the case of successful call establishment between GVP and the external party. For any outbound call, once SSG receives the TreatmentApplied event from SIP Server, the call is deemed as successful and SSG invokes the NotificationURL. Note that the NotificationURL is invoked when the successful request is removed from the DB during batched deletions.

For example:

```
http://test.genesyslab.com/trigger/result.asp?Token=T3001&RequestID
=123345&TenantName=Environment&IVRProfileName=Application&Telnum=40
86552367&AttemptsMade=1&MaxAttempts=3&TimeToLive=12000s&TTLRemainin
g=4367s&CallUUID=
8GQE8C05B56SV2HRTTJ8M9RFR8000001&Result=SUCCESS&Status=DestinationB
usy:RingNoAnswer
```

Table 111 describes the various parameters that Supplementary Services Gateway adds in the query string of the Notifications during a successful notification.

Table 111: Successful Notification URL Parameters

Attribute	Description
`Token`	A unique token that is generated by the TA for each single or bulk create request. When the Supplementary Services Gateway responds to the original `CREATE` request, the response contains the token that is associated with that request. The Supplementary Services Gateway does not enforce uniqueness of the token. This attribute is mandatory.
`Request ID`	A unique internal ID that is generated by the Supplementary Services Gateway for each `Create` request. This attribute is passed back to the TA when requests are successfully added to the database.
`TenantName`	The name of the tenant in the original create request.
`IVRProfileName`	The name of the IVR profile that will be used for this outbound call. IVR profiles that are used in outbound calls are provisioned in Genesys Administrator and are sent to the Resource Manager through SIP Server. The Supplementary Services Gateway does not perform validation on IVR profiles. This attribute is mandatory.
`Telnum`	The telephone number that is used to make an outbound call (for SIP requests, it must be the SIP URI). This attribute is mandatory.
`MaxAttempts`	The number of times that the Supplementary Services Gateway attempts to place the outbound call, should it fail. When the maximum number of attempts is reached, no further attempts are made and the Supplementary Services Gateway sends a Notification URL to the TA indicating that call initiation failed. This attribute is optional.
`Attempts Made`	The total number of times that the Supplementary Services Gateway attempted to place this outbound call.
`TimeToLive`	The length of time (in seconds or milliseconds) that the request stays alive in persistent storage. If the outbound call is not initiated within this time period, no further attempts are made and the Supplementary Services Gateway sends a Notification URL to the TA indicating that call initiation failed. This attribute is optional.
`TTL Remaining`	The length of time remaining (in seconds or milliseconds) for the request to stay alive.

Table 111: Successful Notification URL Parameters (Continued)

Attribute	Description
CallUUID	The unique identifier that is generated by SIP Server for this call. **Note:** If multiple attempts have been made for this outbound call, the last call's CallUUID is included in the Notification URL.
Result	SUCCESS
Status	The failure reasons (separated by a colon) of prior attempts if more than one attempt was made for this call. For example, • CallStateNoAnswer:voice • AnsweringMachineDetected:voice • FaxDetected:AnsweringMachineDetected

Result Notification on Failure

When an outbound request fails (exceeds the maximum number of attempts, or TTL expires), the Supplementary Services Gateway sends the Notification URL to the Trigger Application with the results.

The parameters that are passed in the query string of the Notification URL are the same as the positive notification scenario. The following table describes the differences in values, where applicable.

Table 112: Failure Notification URL Parameters

Attribute	Description
Token	A unique token that is generated by the TA for each single or bulk create request. When the Supplementary Services Gateway responds to the original `CREATE` request, the response contains the token that is associated with that request. The Supplementary Services Gateway does not enforce uniqueness of the token. This attribute is mandatory.
Request ID	A unique internal ID that is generated by the Supplementary Services Gateway for each `Create` request. This attribute is passed back to the TA when requests are successful added to the database.
TenantName	The name of the tenant in the original create request.

Table 112: Failure Notification URL Parameters (Continued)

Attribute	Description
`IVRProfileName`	The name of the IVR profile that will be used for this outbound call. IVR profiles that are used in outbound calls are provisioned in Genesys Administrator and sent to the Resource Manager through SIP Server. The Supplementary Services Gateway does not perform validation on IVR profiles. This attribute is mandatory.
`Telnum`	The telephone number that is used to make an outbound call (for SIP requests, it must be the SIP URI). This attribute is mandatory.
`MaxAttempts`	The number of times that the Supplementary Services Gateway attempts to place the outbound call, should it fail. When the maximum number of attempts is reached, no further attempts are made and the Supplementary Services Gateway sends a Notification URL to the TA indicating that call initiation failed. This attribute is optional.
`Attempts Made`	The total number of times that the Supplementary Services Gateway attempted to place this outbound call.
`TimeToLive`	The length of time (in seconds or milliseconds) that the request stays alive in persistent storage. If the outbound call is not initiated within this time period, no further attempts are made and the Supplementary Services Gateway sends a Notification URL to the TA indicating that call initiation failed. This attribute is optional.
`TTL Remaining`	The length of time remaining (in seconds or milliseconds) for the request to stay alive.
`CallUUID`	The unique identifier generated by SIP Server for this call. **Note:** If multiple attempts have been made for this outbound call, the last call's CallUUID is included in the Notification URL.
`Result`	FAILURE
`Status`	The failure reasons (separated by a colon) of prior attempts for all attempts that are made for this call. For example, • `voice:MaxAttempts Exceeded` • `Unknown External Error:Unknown External Error:TTL Expired` • `FATAL ERROR - InvalidNum` • `DestinationBusy:FATAL ERROR - FaxDetected`

Fatal Errors

When the Supplementary Services Gateway receives any failure reason, it is considered a `FATAL ERROR`, and the request is no longer sent for outbound call processing. For example, the reason `FaxDetected` would be considered a `FATAL ERROR` when the CPD tag includes `detect=voice,am,` or `voice,am,` and the call lands on a Fax machine.

`Status = FATAL ERROR—<Reason from any one from the below list>`

- `InvalidNum`
- `CallStateSitVacant`
- `CallStateSitNocircuit`
- `FaxDetected`

`Status = Fatal Error—InvalidNum`

Table 113 lists and describes the possible call failure reasons.

Table 113: Call Failure Reasons

Reason	Description
Voice	Positive human voice is detected.
DestinationBusy	The dialed number is busy. The call is retried again for processing.
CallStateSit Detected	A SIT error occurs. If the call has `Attempts` and `TTL` remaining, it is retried for processing.
AnsweringMachine Detected	Answering Machine is detected.
InvalidNum	An invalid number is dialed. The call is considered a `FATAL ERROR`, and it is not held for processing.
CallStateSit Vacant	A SIT error occurs. The call is considered a `FATAL ERROR`, and it is not held for processing.
CallStateSit Intercept	A SIT error occurs. If the call has `Attempts` and `TTL` remaining, it is retried for processing.
CallStateSit Unknown	A SIT error occurs. If the call has `Attempts` and `TTL` remaining, it is retried for processing.
CallStateSit Nocircuit	A SIT error occurs. The call is considered a `FATAL ERROR`, and it is not held for processing.
CallStateSit Reorder	A SIT error occurs. If the call has `Attempts` and `TTL` remaining, it is retried for processing.

Table 113: Call Failure Reasons (Continued)

Reason	Description
FaxDetected	Fax Machine is detected.
CallStateUnknown	An unknown error occurred. The call is considered as human voice.
CallStateTransferred CallStateConferenced CallStateGeneralError CallStateRemote Release CallStateNoAnswer CallStateAll TrunksBusy CallStateQueueFull CallStateCleared CallStateOverflowed CallStateAbandoned CallStateRedirected CallStateForwarded CallStateConsult CallStatePickedup CallStateDropped CallState Droppednoanswer CallStateCovered CallStateConverseOn CallStateBridged CallStateDeafened CallStateHeld	If these calls have `Attempts` and `TTL` remaining, they are retried for processing.
UnknownExternal Error	The call failure reason is not known. If this call has `Attempts` and `TTL` remaining, it is retried for processing.

Table 113: Call Failure Reasons (Continued)

Reason	Description
FATALERROR	The call failed due to the following reasons: • `InvalidNum` • `CallStateSitVacant` • `CallStateSitNocircuit` • `FaxDetected` The call is not held for further processing.
MaxAttempts Exceeded	The maximum number of attempts for the request is reached.
TTL Expired	The TimeToLive for the request has expired.
Request is Cancelled by Trigger Application	The request is cancelled by the Trigger Application.

Root Page Access

The Supplementary Services Gateway provides Service Description details when it's root URL is accessed through a web browser. The HTTP interface returns a listing of all available services when the root page is accessed which enables the SSG's HTTP interface self-documented.

The service listing is returned in Web Application Description Language (WADL) format that details services such as Method, Parameters, and help information.

The root page can be accessed with any browser in the following format:

`http://<Name/IP Address of SSG Machine>:<Http Port>/` (for example, `http://172.24.129.86:9800/`).

The following shows and example of the HTTP GET for Service Listing:

```
GET /
Accept: */*
 Accept-Encoding: gzip, deflate
 Host: 172.24.129.81:9800
 User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0;
InfoPath.1; .NET CLR 2.0.50727)
 Cache-Control: max-stale=0
 Connection: Keep-Alive
```

HTTP XML Schema

The Supplementary Services Gateway uses an embedded HTTP server to initiate outbound calls with third-party Trigger Applications. For more information on HTTP requests and responses, see the "How the Supplementary Services Gateway Works" chapter of the *Genesys Voice Platform 8.1 Deployment Guide*.

Genesys uses the standard HTTP POST, GET, and DELETE methods for client/server communication which conforms a defined XML schema. The POST, GET, and DELETE methods are used to send requests and responses between the server and the client.

Request Schema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="SSGRequest">
   <xs:complexType>
   <xs:annotation>
    <xs:documentation xml:lang="en">
    A single POST body can contain single create/query/cancel or multiple,
    and any combination of the three.
    It must conform to the XML request schema present in schema directory
    under root path.
    </xs:documentation>
   </xs:annotation>
     <xs:sequence>
       <xs:element name="CreateRequest" minOccurs="0" maxOccurs="unbounded"
  type="CreateRequestDef"/>
       <xs:element name="QueryRequest" minOccurs="0" maxOccurs="unbounded"
  type="QueryRequestDef"/>
       <xs:element name="CancelRequest" minOccurs="0" maxOccurs="unbounded"
  type="CancelRequestDef"/>
     </xs:sequence>
   </xs:complexType>
  </xs:element>

  <xs:complexType name="CreateRequestDef">
   <xs:annotation>
    <xs:documentation xml:lang="en">
    "CreateRequest" tag is used to specify the attributes used for
    creating new outbound call requests.
    </xs:documentation>
   </xs:annotation>
     <xs:sequence>
     <xs:element name="cpd" minOccurs="0" maxOccurs="1">
      <xs:annotation>
      <xs:documentation xml:lang="en">
```

```
     "cpd" tag defined in CreateRequest is used for supporting Call Progress
Detection in SSG.
     All CPD attributes are optional.
    </xs:documentation>
    </xs:annotation>
      <xs:complexType>
         <xs:attribute name="record" use="optional">
          <xs:annotation>
          <xs:documentation xml:lang="en">
           Specifies if the CPD part of the call should be recorded.
           true or 1: CPD part to be recorded
           false or 0: Do not record CPD part
          </xs:documentation>
          </xs:annotation>
         <xs:simpleType>
            <xs:restriction base="xs:boolean"/>
         </xs:simpleType>
         </xs:attribute>

         <xs:attribute name="preconnect" use="optional">
          <xs:annotation>
          <xs:documentation xml:lang="en">
           This attribute is used to decide when to start the CPD.
           true or 1: CPD is started as soon as the first RTP packet is received.
           false or 0: CPD is started when call is connected.
          </xs:documentation>
          </xs:annotation>
         <xs:simpleType>
            <xs:restriction base="xs:boolean"/>
         </xs:simpleType>
         </xs:attribute>

         <xs:attribute name="rnatimeout" use="optional">
          <xs:annotation>
          <xs:documentation xml:lang="en">
           Timeout to be applied for Ring No Answer scenario.
           Unit is in sec (e.g. 30s). If no unit is specified, seconds assumed.
           The range, enforced by SSG through XML Schema is 1 to 60 seconds.
          </xs:documentation>
          </xs:annotation>
         <xs:simpleType>
            <xs:restriction base="xs:string">
               <xs:pattern value="60|[1-9]s|[1-5][0-9]s|60s|[1-9]|[1-5][0-9]"/>
            </xs:restriction>
         </xs:simpleType>
         </xs:attribute>

         <xs:attribute name="postconnecttimeout" use="optional">
          <xs:annotation>
          <xs:documentation xml:lang="en">
           Timeout to be applied for postconnect CPD.
```

```
         Unit is in sec or msec (e.g. 20s or 3000ms). If no unit is specified,
seconds assumed.
         The range, enforced by SSG through XML Schema is 1 to 60 seconds.
        </xs:documentation>
        </xs:annotation>
       <xs:simpleType>
          <xs:restriction base="xs:string">
             <xs:pattern
value="60|[1-9]s|[1-5][0-9]s|60s|[1-9]|[1-5][0-9]|[1-9][0-9]{3}ms|[1-5][0-9]{4}ms|60
000ms"/>
          </xs:restriction>
       </xs:simpleType>
       </xs:attribute>

       <xs:attribute name="detect" use="optional">
        <xs:annotation>
        <xs:documentation xml:lang="en">
         This attribute provides control to the Trigger Application about
         what to do with the outbound call when various types of CPD are detected.
          none (default): CPD in not requested at all by the customer. As soon as
call is connected, start IVR.
          all: Turn on full CPD. As soon as call is connected, start IVR.
          voice: Only if voice is detected, connect to IVR. Any other detection,
retry.
          am: Only if answering m/c is detected, connect to IVR. Any other
detection, retry.
          fax: Only if fax is detected, connect to IVR. Any other detection, retry.
          voice/am/fax can be combined with comma separation (e.g. voice,am or
am,fax or voice,am,fax etc.).
          Refer the XML Schema for the combinations
        </xs:documentation>
        </xs:annotation>
       <xs:simpleType>
         <xs:restriction base="xs:string">
             <xs:enumeration value="none" />
             <xs:enumeration value="all" />
             <xs:enumeration value="voice" />
             <xs:enumeration value="am" />
             <xs:enumeration value="fax" />
             <xs:enumeration value="voice,am" />
             <xs:enumeration value="voice,fax" />
             <xs:enumeration value="am,fax" />
             <xs:enumeration value="voice,am,fax" />
         </xs:restriction>
       </xs:simpleType>
       </xs:attribute>

    </xs:complexType>
  </xs:element>

  <xs:element name="CustomData" minOccurs="0" maxOccurs="1">
```

```
      <xs:complexType>
       <xs:annotation>
        <xs:documentation xml:lang="en">
        "CustomData" tag defined in CreateRequest is to allow the user to pass
         additional key/value pairs to the IVR application.
         To add each Key/Value pair, a sub-element "KeyValue" should be added with
attributes
         "Key" carrying "KeyName" and
         "Value" carrying "Value" for the above KeyName.
        </xs:documentation>
        </xs:annotation>
      <xs:sequence>
         <xs:element name="KeyValue" minOccurs="1" maxOccurs="unbounded">
         <xs:complexType>
            <xs:sequence>
            </xs:sequence>
            <xs:attribute name="Key" use="required">
             <xs:annotation>
              <xs:documentation xml:lang="en">
              "Key" carrying "KeyName"
              </xs:documentation>
               </xs:annotation>
            <xs:simpleType>
               <xs:restriction base="xs:NMTOKEN">
                  <xs:maxLength value="255"/>
               </xs:restriction>
            </xs:simpleType>
            </xs:attribute>

            <xs:attribute name="Value" use="required">
             <xs:annotation>
              <xs:documentation xml:lang="en">
              Value to be provided for the KeyName
              </xs:documentation>
               </xs:annotation>

            <xs:simpleType>
               <xs:restriction base="xs:string">
                  <xs:maxLength value="255"/>
               </xs:restriction>
            </xs:simpleType>
            </xs:attribute>
         </xs:complexType>
         </xs:element>
      </xs:sequence>
      </xs:complexType>
   </xs:element>
   </xs:sequence>

   <xs:attribute name="IVRProfileName" use="required">
    <xs:annotation>
```

```
 <xs:documentation xml:lang="en">
  Name of the Application Profile to be used for an outbound call.
 </xs:documentation>
 </xs:annotation>
<xs:simpleType>
   <xs:restriction base="xs:NMTOKEN">
      <xs:minLength value="1"/>
      <xs:maxLength value="255"/>
   </xs:restriction>
</xs:simpleType>
</xs:attribute>

<xs:attribute name="NotificationURL" use="required">
 <xs:annotation>
 <xs:documentation xml:lang="en">
  This URL will be used by SSG to asynchronously notify the
  Trigger Application with the result of an outbound call
  (success or failure).
 </xs:documentation>
 </xs:annotation>
<xs:simpleType>
   <xs:restriction base="xs:token">
      <xs:minLength value="1"/>
   </xs:restriction>
</xs:simpleType>
</xs:attribute>

<xs:attribute name="Telnum" use="required">
 <xs:annotation>
 <xs:documentation xml:lang="en">
  Telephone Number to make an outbound call.
 </xs:documentation>
 </xs:annotation>
<xs:simpleType>
   <xs:restriction base="xs:token">
      <xs:minLength value="1"/>
      <xs:pattern value="([a-z0-9A-Z.])*(@)?([a-z0-9A-Z.])*"/>
   </xs:restriction>
</xs:simpleType>
</xs:attribute>

<xs:attribute name="Token" use="required">
 <xs:annotation>
 <xs:documentation xml:lang="en">
  The Trigger Application is expected to pass a unique Token with each
  create request to SSG.
 </xs:documentation>
 </xs:annotation>
<xs:simpleType>
   <xs:restriction base="xs:token">
      <xs:minLength value="1"/>
```

```
        <xs:maxLength value="255"/>
      </xs:restriction>
    </xs:simpleType>
    </xs:attribute>

    <xs:attribute name="MaxAttempts" use="required">
     <xs:annotation>
     <xs:documentation xml:lang="en">
      Number of times SSG should attempt to place the outbound call.
     </xs:documentation>
     </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:nonNegativeInteger">
        <xs:minInclusive value="1"/>
      </xs:restriction>
    </xs:simpleType>
    </xs:attribute>

    <xs:attribute name="TimeToLive" use="required">
     <xs:annotation>
     <xs:documentation xml:lang="en">
      Duration the outbound call request can live in the persistent storage.
     </xs:documentation>
     </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
         <xs:pattern
  value="[6-9][0-9]s|[1-9][0-9]{2}s|[1-9][0-9]{3}s|[1-9][0-9]{4}s|[6-9][0-9]|[1-9][0-9
  ]{2}|[1-9][0-9]{3}|[1-9][0-9]{4}"/>
      </xs:restriction>
    </xs:simpleType>
    </xs:attribute>

    <xs:attribute name="ANI" use="optional">
     <xs:annotation>
     <xs:documentation xml:lang="en">
      ANI that is passed on in the outbound call to the external party.
     </xs:documentation>
     </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:pattern value="([a-z0-9A-Z.])*(@)?([a-z0-9A-Z.])*"/>
      </xs:restriction>
    </xs:simpleType>
    </xs:attribute>

  </xs:complexType>

  <xs:complexType name="QueryRequestDef">
   <xs:annotation>
    <xs:documentation xml:lang="en">
```

```
     "QueryRequest" tag is used to specify the attributes used for
     fetching the details of an existing outbound call requests from
     SSG's persistence storage.
     </xs:documentation>
    </xs:annotation>

      <xs:attribute name="Token" use="optional">
       <xs:annotation>
       <xs:documentation xml:lang="en">
        The Trigger Application is expected to pass the Token with each
        query request to SSG that was received in create request.
       </xs:documentation>
       </xs:annotation>
      <xs:simpleType>
         <xs:restriction base="xs:token">
            <xs:maxLength value="255"/>
         </xs:restriction>
      </xs:simpleType>
      </xs:attribute>

      <xs:attribute name="RequestID" use="required">
       <xs:annotation>
       <xs:documentation xml:lang="en">
        The identifier of the outbound call request whose details needs to be
        fetched from SSG's persistent storage when passed in QueryRequest.
       </xs:documentation>
       </xs:annotation>
      <xs:simpleType>
         <xs:restriction base="xs:nonNegativeInteger">
            <xs:minInclusive value="1"/>
         </xs:restriction>
      </xs:simpleType>
      </xs:attribute>

   </xs:complexType>

   <xs:complexType name="CancelRequestDef">
    <xs:annotation>
     <xs:documentation xml:lang="en">
     "CancelRequest" tag is used to specify the attributes used for
     cancelling an existing outbound call requests from
     SSG's persistence storage.
     </xs:documentation>
    </xs:annotation>


      <xs:attribute name="Token" use="optional">
       <xs:annotation>
       <xs:documentation xml:lang="en">
        The Trigger Application is expected to pass the Token with each
        cancel request to SSG that was received in create request.
```

```
      </xs:documentation>
      </xs:annotation>
     <xs:simpleType>
        <xs:restriction base="xs:token">
           <xs:maxLength value="255"/>
        </xs:restriction>
     </xs:simpleType>
     </xs:attribute>

     <xs:attribute name="RequestID" use="required">
      <xs:annotation>
      <xs:documentation xml:lang="en">
       The identifier of the outbound call request whose details needs to be
       deleted from persistent storage when passed in CancelRequest.
      </xs:documentation>
      </xs:annotation>
     <xs:simpleType>
        <xs:restriction base="xs:nonNegativeInteger">
           <xs:minInclusive value="1"/>
        </xs:restriction>
     </xs:simpleType>
     </xs:attribute>

  </xs:complexType>

</xs:schema>
```

Response Schema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:element name="SSGResponse">
    <xs:complexType>
    <xs:annotation>
     <xs:documentation xml:lang="en">
     SSG responds to the Trigger Application with 200 OK and the body contains the
   result
     (either success or failure) formatted in XML response schema.
     The response body in 200 OK is single/bulk depending on whether the POST request
   was single/bulk.
     It must conform to the XML response schema present in schema directory
     under root path.
     </xs:documentation>
    </xs:annotation>
    <xs:sequence>
            <xs:element name="ResponseElement" minOccurs="0" maxOccurs="unbounded"
   type="ResponseElementDef"/>
            <xs:element name="FailureDescription" minOccurs="0" maxOccurs="1"
   type="FailureDescriptionDef"/>
    </xs:sequence>
    </xs:complexType>
```

```
</xs:element>

<xs:complexType name="ResponseElementDef">
<xs:annotation>
<xs:documentation xml:lang="en">
 "ResponseElement" tag is used to specify the attributes sent as
 response to the received create/query/cancel request.
</xs:documentation>
</xs:annotation>

       <xs:attribute name="ResponseType" use="required">
    <xs:annotation>
     <xs:documentation xml:lang="en">
      "CreateRequest" tag is used to specify the attributes used for
     creating new outbound call requests.
     SUCCESS : If the create/quey/cancel request operation was successful.

     FAILURE : If the create/query/cancel request parsing failed or some
         mandatory attributes missing or DB insertion failed.
         If parsing or other validation fails, then FailureDescription
         is generated with ReasonCode and Reason.
         If parsing and other validations succeed, but there are other
         failures (e.g. specific record insertion into db failed),
         then the response would contain Token(s) in the ResponseElement(s).

     </xs:documentation>
    </xs:annotation>
             <xs:simpleType>
             <xs:restriction base="xs:string">
                  <xs:pattern value="SUCCESS|FAILURE"/>
             </xs:restriction>
             </xs:simpleType>
       </xs:attribute>

       <xs:attribute name="Token" use="required">
    <xs:annotation>
     <xs:documentation xml:lang="en">
      Token provided in the original create request is passed back.
     </xs:documentation>
    </xs:annotation>
             <xs:simpleType>
             <xs:restriction base="xs:NMTOKEN">
                  <xs:maxLength value="255"/>
             </xs:restriction>
             </xs:simpleType>
       </xs:attribute>

       <xs:attribute name="RequestID" use="optional">
    <xs:annotation>
     <xs:documentation xml:lang="en">
      SSG generates an internal unique request ID for each create request.
```

```
         For query or cancel request operation, RequestID provided is passed back.
        </xs:documentation>
       </xs:annotation>
              <xs:simpleType>
              <xs:restriction base="xs:nonNegativeInteger">
                   <xs:minInclusive value="0"/>
              </xs:restriction>
              </xs:simpleType>
         </xs:attribute>

         <xs:attribute name="TenantName" use="optional">
       <xs:annotation>
        <xs:documentation xml:lang="en">
         Name of the Tenant provided in the original create request is passed back.
        </xs:documentation>
       </xs:annotation>
              <xs:simpleType>
              <xs:restriction base="xs:string"/>
              </xs:simpleType>
         </xs:attribute>

         <xs:attribute name="IVRProfileName" use="optional">
       <xs:annotation>
        <xs:documentation xml:lang="en">
         Name of the IVRProfile provided in the original create request is passed
back.
        </xs:documentation>
       </xs:annotation>
              <xs:simpleType>
              <xs:restriction base="xs:string"/>
              </xs:simpleType>
         </xs:attribute>

         <xs:attribute name="NotificationURL" use="optional">
       <xs:annotation>
        <xs:documentation xml:lang="en">
         NotificationURL provided in the original create request is passed back.
        </xs:documentation>
       </xs:annotation>
              <xs:simpleType>
              <xs:restriction base="xs:string"/>
              </xs:simpleType>
         </xs:attribute>

         <xs:attribute name="MaxAttempts" use="optional">
       <xs:annotation>
        <xs:documentation xml:lang="en">
         Maximum Attempts provided in the original create request is passed back..
        </xs:documentation>
       </xs:annotation>
              <xs:simpleType>
```

```
                <xs:restriction base="xs:nonNegativeInteger">
                     <xs:minInclusive value="0"/>
                </xs:restriction>
               </xs:simpleType>
         </xs:attribute>

         <xs:attribute name="AttemptsMade" use="optional">
      <xs:annotation>
       <xs:documentation xml:lang="en">
        The Number of times the requested outbound call has been attempted so far.
       </xs:documentation>
      </xs:annotation>
               <xs:simpleType>
                <xs:restriction base="xs:nonNegativeInteger">
                     <xs:minInclusive value="0"/>
                </xs:restriction>
               </xs:simpleType>
         </xs:attribute>

         <xs:attribute name="TimeToLive" use="required">
      <xs:annotation>
       <xs:documentation xml:lang="en">
        TimeToLive provided in the original create request is passed back.
       </xs:documentation>
      </xs:annotation>
               <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:pattern
value="[6-9][0-9]s|[1-9][0-9]{2}s|[1-9][0-9]{3}s|[1-9][0-9]{4}s"/>
                </xs:restriction>
          </xs:simpleType>
         </xs:attribute>

         <xs:attribute name="TTLRemaining" use="required">
      <xs:annotation>
       <xs:documentation xml:lang="en">
        Time remaining for a request out of received TimeToLive.
       </xs:documentation>
      </xs:annotation>
               <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:pattern
value="[6-9][0-9]s|[1-9][0-9]{2}s|[1-9][0-9]{3}s|[1-9][0-9]{4}s"/>
                </xs:restriction>
               </xs:simpleType>
         </xs:attribute>

         <xs:attribute name="ReasonCode" use="optional">
      <xs:annotation>
       <xs:documentation xml:lang="en">
        Error code generated by SSG for any failures.
```

```
        </xs:documentation>
       </xs:annotation>
                <xs:simpleType>
                <xs:restriction base="xs:nonNegativeInteger"/>
                </xs:simpleType>
          </xs:attribute>

          <xs:attribute name="Reason" use="optional">
       <xs:annotation>
        <xs:documentation xml:lang="en">
         Description of the error for any failures.
        </xs:documentation>
       </xs:annotation>
                <xs:simpleType>
                <xs:restriction base="xs:token"/>
                </xs:simpleType>
          </xs:attribute>

     <xs:attribute name="InternalAttempts" use="optional">
                <xs:simpleType>
                <xs:restriction base="xs:nonNegativeInteger"/>
                </xs:simpleType>
          </xs:attribute>

          <xs:attribute name="Status" use="optional">
       <xs:annotation>
        <xs:documentation xml:lang="en">
         Current state of the request.
        </xs:documentation>
       </xs:annotation>
                <xs:simpleType>
                <xs:restriction base="xs:string"/>
                </xs:simpleType>
          </xs:attribute>
   </xs:complexType>
   <xs:complexType name="FailureDescriptionDef">
  <xs:annotation>
   <xs:documentation xml:lang="en">
    SSG returns a FailureDescription with appropriate ReasonCode
    and the line number of the parsing error when request parsing
    fails due to schema validation or missing mandatory parameters.
   </xs:documentation>
  </xs:annotation>
    If the ,
  .

          <xs:attribute name="ReasonCode" use="required">
       <xs:annotation>
        <xs:documentation xml:lang="en">
         Error code generated by SSG for any schema validation or missing parameter
failures.
```

```
          </xs:documentation>
         </xs:annotation>
                  <xs:simpleType>
                  <xs:restriction base="xs:nonNegativeInteger"/>
                  </xs:simpleType>
            </xs:attribute>

            <xs:attribute name="Reason" use="required">
         <xs:annotation>
          <xs:documentation xml:lang="en">
           Description of Error code for any schema validation or missing parameter
   failures.
          </xs:documentation>
         </xs:annotation>
                  <xs:simpleType>
                  <xs:restriction base="xs:token"/>
                  </xs:simpleType>
            </xs:attribute>

      </xs:complexType>

</xs:schema>
```

Appendix

J Network Partitioning Configuration Options

This appendix describes the Genesys Voice Platform (GVP) configuration options for each component that are used to support network traffic partitioning.

It contains the following section:

Configuration Options and Protocols

This section provides a list of configuration options that are used by the GVP components to specify the network interfaces that are used for certain types of network traffic.

Table 114 lists the configuration options and protocols that are used by the Media Control Platform.

Table 114: Media Control Platform

Purpose	Protocol	Configuration Option
SIP Calls	SIP	`[sip] transport.*`
	RTP	`[mpc] rtp.localaddr`
HTTP Fetch Requests	HTTP	`[fm] interface`
Connection to RTSP Server for RTSP Prompt Playback	RTSP	`[vrm] rtp.localadddr`
	RTP	`[mpc] rtsp.rtp.localaddr`

Table 114: Media Control Platform (Continued)

Purpose	Protocol	Configuration Option
Connection to MRCPv1 Resources	RTSP	[vrm] rtp.localadddr
	RTP	[vrm] rtp.localadddr
Connection to MRCPv2 Resources	SIP	[mrcpv2client] sip.transport.*
	MRCPv2	[vrm] client.mrcpv2.localaddr
	RTP	[vrm] rtp.localaddr
Composer	Proprietary via TCP	[vxmli] debug.server.ip

Table 115 lists the configuration options and protocols that are used by the Call Control Platform.

Table 115: Call Control Platform

Purpose	Protocol	Configuration Option
SIP Calls	SIP	[sip] transport.*
HTTP Fetch Requests	HTTP	[fm] interface
Basic HTTP Event I/O Processor	HTTP	[ccxmli] basichttp.recv.host
Session Creation Event I/O Processor	HTTP	[ccxmli] createsession.recv.host

Table 116 lists the configuration options and protocols that are used by the Resource Manager.

Table 116: Resource Manager

Purpose	Protocol	Configuration Option
SIP Calls	SIP	[proxy] sip.transport.*, sip.localport, sip.localsecureport
SIP Monitoring	SIP	[monitor] sip.transport.*, sip.localport, sip.localsecureport
SIP Subscriptions	SIP	[subscription] sip.transport.*, sip.localport, sip.localsecureport
SIP Registrations	SIP	[registrar] sip.transport.*, sip.localport, sip.localsecureport
Cluster Messaging	TCP	[cluster] virtual-ip, member.1, member.2

Table 117 lists the configuration options and protocols that are used by the MRCP Proxy.

Table 117: MRCP Proxy

Purpose	Protocol	Configuration Option
RTSP Contact URI	RTSP	`[vrmproxy] uri`
RTSP Port Range	RTSP	`[stack] connection.portrange`

Table 118 lists the configuration options and protocols that are used by the CTI Connector.

Table 118: CTI Connector

Purpose	Protocol	Configuration Option
SIP Calls	SIP	`[sip] transport.*`, `[sip] transport.localaddress`, `[sip] transport.localaddress_ipv6`, `[sip] transport.localaddress.srv`
Connection to Cisco ICM for CTI Interaction using GED-125 interface	TCP	`[Tenant1] Ports`
Connection to IVR Server for CTI interaction using XML interface	TCP	`[IServer_sample] iserveraddr`, `iserversocket`

Table 119 lists the configuration options and protocols that are used by the Supplementary Services Gateway.

Table 119: Supplementary Services Gateway

Purpose	Protocol	Configuration Option
HTTP Requests Processing	HTTP	`[HTTP] HTTPPort`
HTTPS Requests Processing	HTTP	`[HTTP] HTTPSPort`, `[HTTP] CertFile`, `[HTTP] CertKeyFile`,
The configured parameters (above) are the which is the IP address of the Supplementary Services Gateway host and are used in the HTTP requests as the port and host name.		

Table 120 lists the configuration options and protocols that are used by the PSTN Connector.

Table 120: PSTN Connector

Purpose	Protocol	Configuration Option
SIP Calls	SIP	[GatewayManager] LocalIPAddress, [GatewayManager] UserAgentAddr
	RTP	[MediaManager] LocalHostAddress

Table 121 lists the configuration options and protocols that are used by the Reporting Server.

Table 121: Reporting Server

Purpose	Protocol	Configuration Option
SIP Calls	SIP	[GatewayManager] LocalIPAddress, [GatewayManager] UserAgentAddr
	RTP	[MediaManager] LocalHostAddress

Appendix

K SIP Customizable Headers and Parameters

The GVP 8.1.6 Media Control Platform (MCP) supports propagation of SIP headers, parameters and request URI parameters to the VXML applications for incoming SIP messages, and customization of SIP headers, parameters and request URI parameters for outgoing SIP messages.

The MCP can be configured to pass incoming SIP INVITE requests for the URI's parameters, headers, and parameters of any headers to the VXML application as session variables.

The contents of this chapter includes:

Abstract Information from Incoming SIP Messages

The configuration parameters `sip.in.invite.headers` and `sip.in.invite.params` can be defined to abstract information from incoming SIP messages. They will generate variables to be sent from the Media Control Platform to the VXML Interpreter in `Sip.Invite.<headername>` and `Sip.Inivte.<headername>.<paramname>` formats, respectively. `Sip.Invite.<headername>` contains the header value, as well as all its parameters. `Sip.Invite.<headername>.<paramname>` contains the value of a specific header parameter.

sip.in.invite.headers

Defines the list of headers to expose to the application. This specifies a list of header names from the incoming INVITE requests, whose values will be exposed to the application.

For example, `sip.in.invite.headers = From To Via`. The exposed values' names will be in the format `sip.invite.<headername>=<value>`. If this value is `*`, then all headers will be exposed. If this value is `none`, then no headers will be exposed. `none` will be ignored alongside other values.

Default: `*`

sip.in.invite.params

Defines list of parameters to expose to the application. This specifies a list of header names from the incoming INVITE requests, whose parameter values will be exposed to the application.

For example, `sip.in.invite.params = From To Via`. The exposed values' names will be in for format `sip.invite.<headername>.<paramname>=<value>`. If this value is `none`, then no parameters will be exposed. `none` will be ignored alongside other values.

Default: `RequestURI`

Session Variables for VXML

The configuration parameter `session_vars` and the Next Generation Interpreter configuration parameter `vxmli.session_vars` can be defined to provide the session variables to the VXML applications.

session_vars

Each session variable entry is composed of three components. The first component is the session variable name as exposed within VoiceXML. The second component is the variable name sent back from the Call Manager. The third component indicates whether the session variable will be included in the request for the initial page URL.

Default:

```
session.connection.answeredby|ANSWEREDBY|0|session.connection.uuipr
otocol|UUIPROTOCOL|0|session.connection.redirect|REDIRECT|0|session
.connection.aai|UUIDATA|0|session.connection.local.uri|LOCALURI|1|s
ession.connection.remote.uri|REMOTEURI|1|session.connection.origina
tor|ORIGIN|0|session.connection.channelidref|PSTNCHANNELID|1|sessio
n.connection.protocol.name|PROTOCOLNAME|0|session.connection.protoc
ol.version|PROTOCOLVERSION|0|session.com.voicegenie.consultdata|con
sultdata|1|session.com.voicegenie.instance.parent|PARENT|1|session.
connection.protocol.isup.natureofconnection.si|NatureOfConnection.S
I|0|session.connection.protocol.isup.natureofconnection.cc|NatureOf
Connection.CCI|0|session.connection.protocol.isup.natureofconnectio
n.ec|NatureOfConnection.EC|0|session.connection.protocol.isup.origi
nalcallednumber.num|OriginalCalledNumber.num|0|session.connection.p
rotocol.isup.originalcallednumber.nai|OriginalCalledNumber.NAI|0|se
ssion.connection.protocol.isup.originalcallednumbe...
```

vxmli.session_vars

Each session variable entry is composed of three components. The first component is the session variable name as exposed within VoiceXML. The second component is the variable name sent back from the Call Manager. The third component indicates either whether the session variable will be included in the request for the initial page URL (0 = do not include, 1 = include in GET, 2 = include in POST, 3 = include in GET and POST), or the type of array of the session variable (6 = associative array, 7 = ???).

Default:

```
session.connection.local.uri|LOCALURI|1|session.connection.remote.u
ri|REMOTEURI|1|session.connection.originator|ORIGIN|1|session.conne
ction.protocol.name|PROTOCOLNAME|0|session.connection.protocol.vers
ion|PROTOCOLVERSION|0|session.connection.protocol.sip.headers|Sip.I
nvite|6|session.connection.redirect|REDIRECTHEADER|7|session.connec
tion.callidref|CALLIDREF|1|session.com.voicegenie.instance.parent|P
ARENT|1|session.connection.ocn|OCN|1|session.connection.rdnis|RDNIS
|1|session.connection.rreason|RREASON|1
```

Here is an example configuration for exposing request URI's `paramA,` request URI's `paramB,` `From` header, and `To` header's `paramC:`

Media Control Platform

```
sip.in.invite.headers=From
sip.in.invite.params=RequestURI To
```

Next Generation Interpreter

```
Vxmli.session_vars=...|session.connection.protocol.sip.invite.from
|Sip.Invite.From|0|session.connection.protocol.sip.invite.requestur
i.paramA|Sip.Invite.RequestURI.paramA|0|session.connection.protocol
.sip.invite.requesturi.paramB|Sip.Invite.RequestURI.paramB|0|sessio
```

```
n.connection.protocol.sip.invite.to.paramC|Sip.Invite.To.paramC|0
```

With the configuration above and the following SIP INVITE message:

```
INVITE sip:test1@10.0.0.25;paramA=valueA;paramB=valueB
SIP/2.0
Via: SIP/2.0/UDP
205.150.90.207:5060;branch=z9hG4bK0809fb404f9bcd
From: <sip:VoiceGenie@205.150.90.207:5060>;tag=9FB30200-
B96C-01D0-5052-C114EBCA0416
To: <sip:test1@10.0.0.25>;paramC=valueC
Max-Forwards: 70
CSeq: 1 INVITE
Call-ID: 9FB30200-B96C-C781-2A00-
F3B654BEA9AD@205.150.90.207:5060
Contact: sip:VoiceGenie@205.150.90.207:5060
Content-Length: 190
Content-Type: application/sdp
v=0
o=Cisco-SIPUA 2455 9673 IN IP4 205.150.90.208
s=SIP Call
c=IN IP4 205.150.90.208
t=0 0
m=audio 30400 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

The following session variables will be defined:

Session Variable	Value
session.connection.protocol.sip.invite.from	<sip:VoiceGenie@205.150.90.207:5060>;tag=9FB30200-B96C-01D0-5052-C114EBCA0416
session.connection.protocol.sip.invite.requesturi.paramA	valueA
session.connection.protocol.sip.invite.requesturi.paramB	valueB
session.connection.protocol.sip.invite.to.paramC	valueC

The Media Control Platform can also be configured to set outgoing SIP INVITE or REFER requests' URI parameters, headers, and parameters of any headers (limitations) using signaling variables from the VXML application. This feature is supported for transfers and calls initiated using `RemDial`, and can be enabled by configuring `sip.out.invite.headers`, `sip.out.invite.params`, `sip.out.refer.headers` and `sip.out.refer.params`. Please see the *Media Control Platform Deployment Guide* regarding the details for these parameters.

Below is an example VoiceXML page that will perform configuration that is described in the above paragraph, using `signalvar`:

```
<?xml version="1.0"?>
```

```
<vxml version="2.1" xmlns="http://www.w3.org/2001/vxml"
xmlns:gvp="http://www.anexample.com/1111/vxml21-extension">
··<property name="com.genesyslab.externalevents.enable" value="false"/>
··<property name="com.genesyslab.externalevents.queue" value="true"/>

··<form id="form1">
····<var name="callvars" expr="new Object()"/>
····<block>
······<assign name="callvars['sip.invite.requesturi.parama']" expr="'valueA'"/>
······<assign name="callvars['sip.invite.requesturi.paramb']" expr="'valueB'"/>
······<assign name="callvars['sip.invite.headerc']" expr="'valueC'"/>
····</block>
····<transfer destexpr="'sip:9090@127.0.0.1:5060'" bridge="true" gvp:signalvar="callvars">
······<filled>
········<exit/>
······</filled>
····</transfer>

··</form>
</vxml>
```

Below is an example Media Control Platform configuration for customizing request URI's paramA, request URI's paramB and HeaderC in outgoing INVITE messages (for <transfer> involving two call legs and remdial calls):

```
sip.out.invite.headers=HeaderC
sip.out.invite.params=RequestURI
```

If the following signaling variables are defined (or the equivalent name/value list is defined and appended to the remdial call request):

```
Sip.Invite.RequestURI.paramA=valueA
Sip.Invite.RequestURI.paramB=valueB
Sip.Invite.HeaderC=valueC
```

Then, the following SIP INVITE message will be generated for the outgoing call:

```
INVITE sip:test1@10.0.0.25;paramA=valueA;paramB=valueB
SIP/2.0
Via: SIP/2.0/UDP
205.150.90.207:5060;branch=z9hG4bK0809fb404f9bcd
From: <sip:VoiceGenie@205.150.90.207:5060>;tag=9FB30200-
B96C-01D0-5052-C114EBCA0416
Session Variable Value
session.connection.protocol.sip.invite.from
<sip:VoiceGenie@205.150.90.207:5060>;tag
=9FB30200-B96C-01D0-5052-
C114EBCA0416
session.connection.protocol.sip.invite.requestur
i.paramA
valueA
session.connection.protocol.sip.invite.requestur
i.paramB
valueB
session.connection.protocol.sip.invite.to.paramC valueC
```

```
User's Guide 545
Appendix K: SIP Customizable Headers and Parameters
To: <sip:test1@10.0.0.25>
Max-Forwards: 70
CSeq: 1 INVITE
Call-ID: 9FB30200-B96C-C781-2A00-
F3B654BEA9AD@205.150.90.207:5060
Contact: sip:VoiceGenie@205.150.90.207:5060
HeaderC: valueC
Content-Length: 190
Content-Type: application/sdp
v=0
o=Cisco-SIPUA 2455 9673 IN IP4 205.150.90.208
s=SIP Call
c=IN IP4 205.150.90.208
t=0 0
m=audio 30400 RTP/AVP 0 101
Chapter 4: Network Interfaces 4.1 SIP
66 VoiceGenie 7.2
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

Genesys

Supplements

Related Documentation Resources

The following resources provide additional information that is relevant to this software. Consult these additional resources as necessary.

Management Framework

- *Framework 8.1 Deployment Guide,* which provides information about configuring, installing, starting, and stopping Framework components.
- *Framework 8.1 Genesys Administrator Deployment Guide*, which provides information on installing, and configuring Genesys Administrator.
- *Framework 8.1 Genesys Administrator Help,* which provides information about configuring and provisioning contact center objects by using the Genesys Administrator.
- *Framework 8.1 Configuration Options Reference Manual,* which provides descriptions of the configuration options for Framework components.

SIP Server

- *Framework 8.1 SIP Server Deployment Guide,* which provides information about configuring and installing SIP Server.

Genesys Voice Platform

- *Genesys Voice Platform 8.1 Deployment Guide,* which provides information about installing and configuring Genesys Voice Platform (GVP).
- *Genesys Voice Platform 8.1 Troubleshooting Guide,* which provides troubleshooting methodology, basic troubleshooting information, and troubleshooting tools.

- *Genesys Voice Platform 8.1 SNMP and MIB Reference*, which provides information about all of the Simple Network Management Protocol (SNMP) Management Information Bases (MIBs) and traps for GVP, including descriptions and user actions.
- *Genesys Voice Platform 8.1 Genesys VoiceXML 2.1 Reference Help,* which provides information about developing Voice Extensible Markup Language (VoiceXML) applications. It presents VoiceXML concepts, and provides examples that focus on the GVP Next Generation Interpreter (NGI) implementation of VoiceXML.
- *Genesys Voice Platform 8.1 Legacy Genesys VoiceXML 2.1 Reference Manual*, which describes the VoiceXML 2.1 language as implemented by the Legacy GVP Interpreter (GVPi) in GVP 7.6 and earlier, and which is now supported in the GVP 8.1 release.
- *Genesys Voice Platform 8.1 Application Migration Guide*, which provides detailed information about the application modifications that are required to use legacy GVP 7.6 voice and call-control applications in GVP 8.1.
- *Genesys Voice Platform 8.1 CCXML Reference Manual,* which provides information about developing Call Control Extensible Markup Language (CCXML) applications for GVP.
- *Genesys Voice Platform 8.1 Configuration Options Reference,* which replicates the metadata available in the Genesys provisioning GUI, to provide information about all the GVP configuration options, including descriptions, syntax, valid values, and default values.
- *Genesys Voice Platform 8.1 Metrics Reference,* which provides information about all the GVP metrics (VoiceXML and CCXML application event logs), including descriptions, format, logging level, source component, and metric ID.

Genesys Voice Platform 8.1 Web Services API wiki, which describes the Web Services API that the Reporting Server supports.

Voice Platform Solution

- *Voice Platform Solution 8.1 Integration Guide,* which provides information about integrating GVP 8.1, SIP Server 8.1, and, if applicable, IVR Server.

Composer Voice

- *Composer 8.1 Deployment Guide,* which provides information about installing and configuring Composer Voice.
- *Composer 8.1 Help,* which provides information about using Composer Voice, a GUI for developing applications based on VoiceXML and CCXML.

Open Standards

- *W3C Voice Extensible Markup Language (VoiceXML) 2.1, W3C Recommendation 19 June 2007,* which is the World Wide Web Consortium (W3C) VoiceXML specification that GVP NGI supports.
- *W3C Voice Extensible Markup Language (VoiceXML) 2.0, W3C Recommendation 16 March 2004,* which is the W3C VoiceXML specification that GVP supports.
- *W3C Speech Synthesis Markup Language (SSML) Version 1.0, Recommendation 7 September 2004,* which is the W3C SSML specification that GVP supports.
- *W3C Voice Browser Call Control: CCXML Version 1.0, W3C Working Draft 29 June 2005,* which is the W3C CCXML specification that GVP supports.
- *W3C Semantic Interpretation for Speech Recognition (SISR) Version 1.0, W3C Recommendation 5 April 2007,* which is the W3C SISR specification that GVP supports.
- *W3C Speech Recognition Grammar Specification (SRGS) Version 1.0, W3C Recommendation 16 March 2004,* which is the W3C SRGS specification that GVP supports.

Genesys

- *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and CTI terminology and acronyms used in this document.
- *Genesys Migration Guide*, which ships on the Genesys Documentation Library DVD, and which provides documented migration strategies for Genesys product releases. Contact Genesys Technical Support for more information.
- Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at `http://genesyslab.com/support`.

Information about supported operating systems and third-party software is available on the Genesys Customer Care website in the following documents:

Genesys Supported Operating Environment Reference Guide

Genesys Supported Media Interfaces Reference Manual

Genesys product documentation is available on the:

- Genesys Technical Support website at `http://genesyslab.com/support`.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at `orderman@genesyslab.com`.

Consult these additional resources as necessary:

- *Genesys Hardware Sizing Guide,* which provides information about Genesys hardware sizing guidelines for the Genesys 8.x releases.
- *Genesys Interoperability Guide,* which provides information on the compatibility of Genesys products with various Configuration Layer Environments; Interoperability of Reporting Templates and Solutions; and Gplus Adapters Interoperability.
- *Genesys Licensing Guide,* which introduces you to the concepts, terminology, and procedures relevant to the Genesys licensing system.
- *Genesys Database Sizing Estimator 7.6 Worksheets,* which provides a range of expected database sizes for various Genesys products.

For additional system-wide planning tools and information, see the release-specific listings of System Level Documents on the Genesys Technical Support website, accessible from the `system level documents by release` tab in the Knowledge Base `Browse Documents` Section.

Genesys product documentation is available on the:

- Genesys Technical Support website at `http://genesyslab.com/support`.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at `orderman@genesyslab.com`.
- Genesys Online Documentation at `docs.genesyslab.com`.

Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

80fr_ref_06-2008_v8.0.001.00

You will need this number when you are talking with Genesys Technical Support about this product.

Screen Captures Used in This Document

Screen captures from the product graphical user interface (GUI), as used in this document, may sometimes contain minor spelling, capitalization, or grammatical errors. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Type Styles

Table 122 describes and illustrates the type conventions that are used in this document.

Table 122: Type Styles

Type Style	Used For	Examples
Italic	• Document titles • Emphasis • Definitions of (or first references to) unfamiliar terms • Mathematical variables Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on page 574).	Please consult the *Genesys Migration Guide* for more information. Do *not* use this value for this option. A *customary and usual* practice is one that is widely accepted and used within a particular industry or profession. The formula, $x + 1 = 7$ where x stands for . . .

Table 122: Type Styles (Continued)

Type Style	Used For	Examples
Monospace font (Looks like `teletype` or `typewriter text`)	All programming identifiers and GUI elements. This convention includes: • The *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages. • The values of options. • Logical arguments and command syntax. • Code samples. Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.	Select the `Show variables on screen` check box. In the `Operand` text box, enter your formula. Click `OK` to exit the `Properties` dialog box. T-Server distributes the error messages in `EventError` events. If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls. Enter `exit` on the command line.
Square brackets ([ ])	A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.	`smcp_server -host [/flags]`
Angle brackets (< >)	A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise. **Note:** In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.	`smcp_server -host <confighost>`

Index

Symbols

Numerics

A

B

C

D

E

I

N

O

P

T

U

V

W

X